MTAT.03.227 Machine Learning

No Free Lunch Theorems and Statistical Learning Theory

Sven Laur
University of Tartu

Quick recap of the basic concepts

Let $L(a, b)$ be a loss function that characterises the cost of incorrect guesses. Then the aim of machine learning algorithms is to minimise *risk*

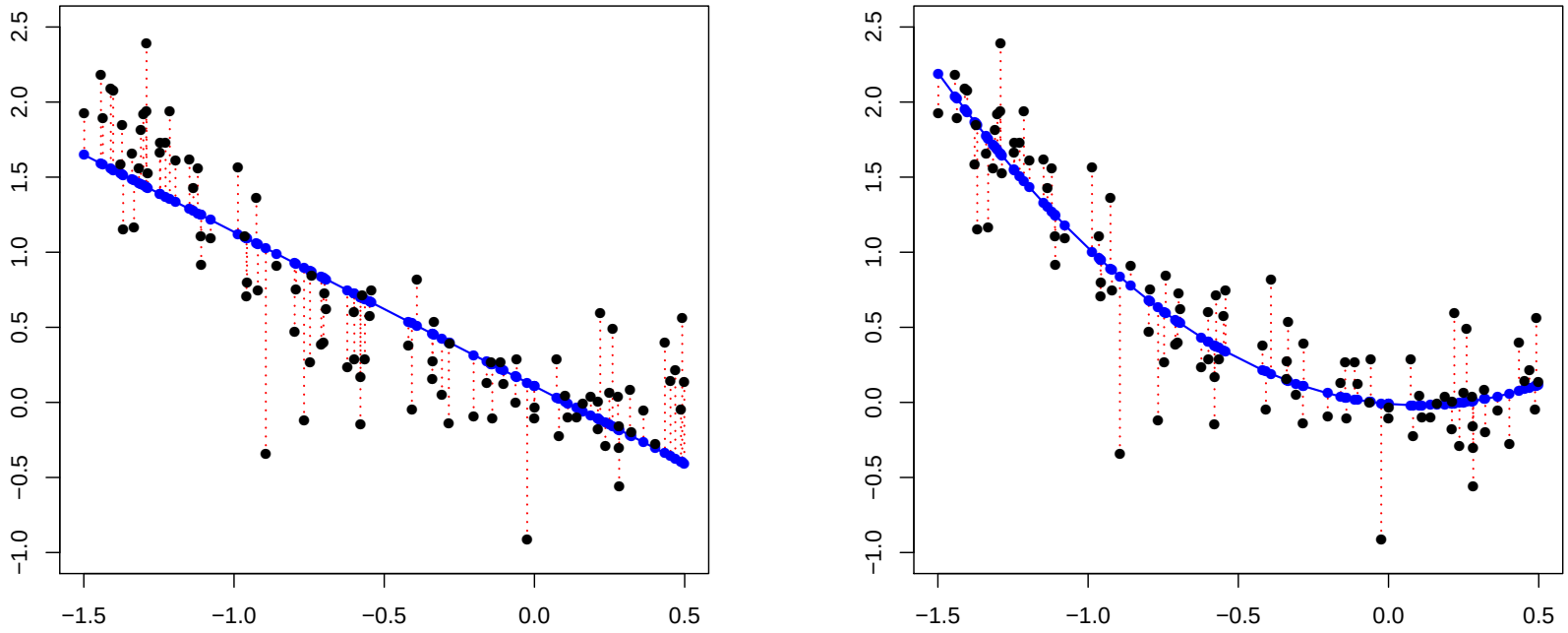
$$R(f) = \mathop{\mathbf{E}}_{\boldsymbol{x}, y \leftarrow \mathcal{D}} \big(L(f(\boldsymbol{x}), y) \big)$$

over an existing but unknown distribution $\mathcal{D}$. Usually, we seek a solution from a fixed function class $\mathcal{H}$ and try to minimise empirical risk

$$R_m(f) = \frac{1}{m} \cdot \sum_{i=1}^{m} L(f(\boldsymbol{x}_i), y_i)$$

where samples are assumed to be independent and identically distributed.

Mean square error as an example



Mean square error for a linear and quadratic model that minimise the empirical risk. Dotted lines visualise individual residues.

Bias-Variance Dilemma

Model bias and consistency

As the set of potential solutions is limited by $\mathcal{H}$, we might never discover the true function. A *model bias* for $\mathcal{H}$ and true distribution $\mathcal{D}$ is

$$\text{Bias}(\mathcal{H}|\mathcal{D}) = \min_{f \in \mathcal{H}} \mathop{\mathbf{E}}_{\boldsymbol{x}, y \leftarrow \mathcal{D}} \big(L(f(\boldsymbol{x}), y)\big)$$

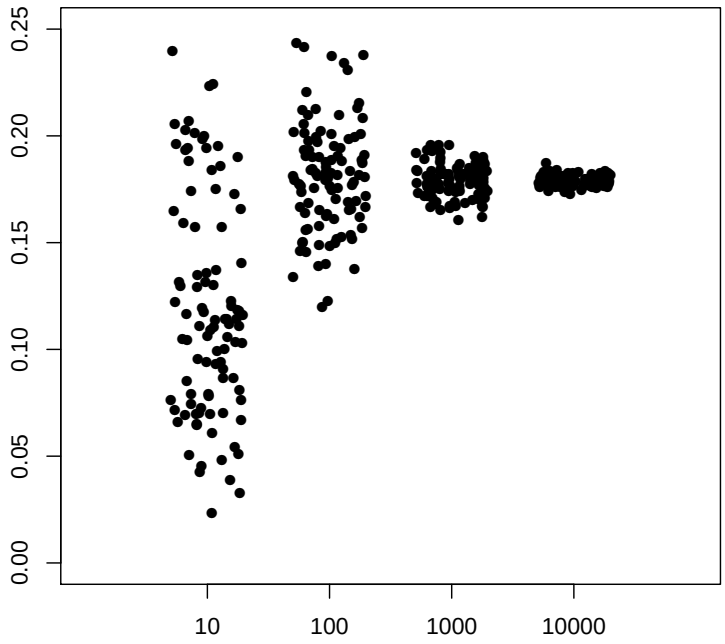
A machine learning algorithm $\mathcal{A}$ is *asymptotically consistent* for a class of data distributions $\mathfrak{D}$ if for any data distribution $\mathcal{D}$ form the class $\mathfrak{D}$

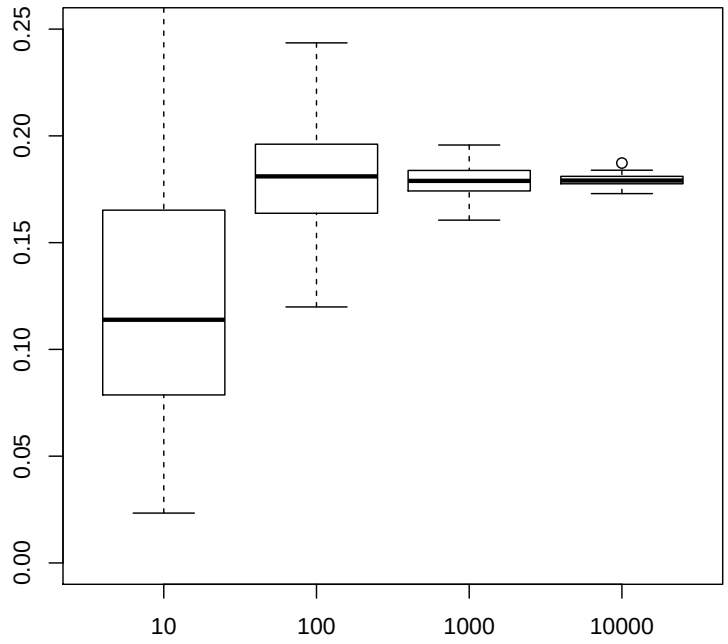
$$\mathop{\mathbf{E}}_{\mathcal{D}}(R(f_m)) \underset{m}{\longrightarrow} \text{Bias}(\mathcal{H}|\mathcal{D})$$

where f_m is the function returned by $\mathcal{A}$ given m samples and $\mathcal{H}$ is the function class from which $\mathcal{A}$ chooses its output.

Asymptotic consistency of a linear model

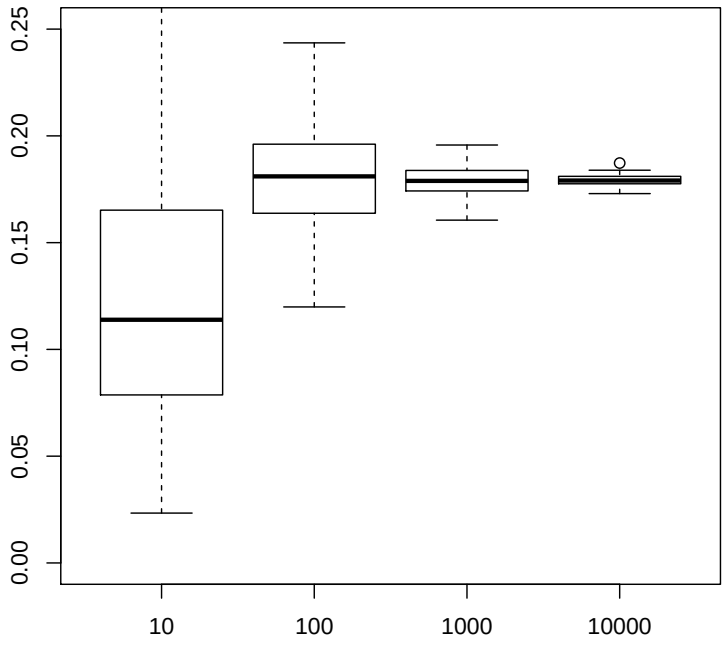
Linear models are known to be asymptotically consistent. Figures depict the empirical risk for the data distribution visualised before.

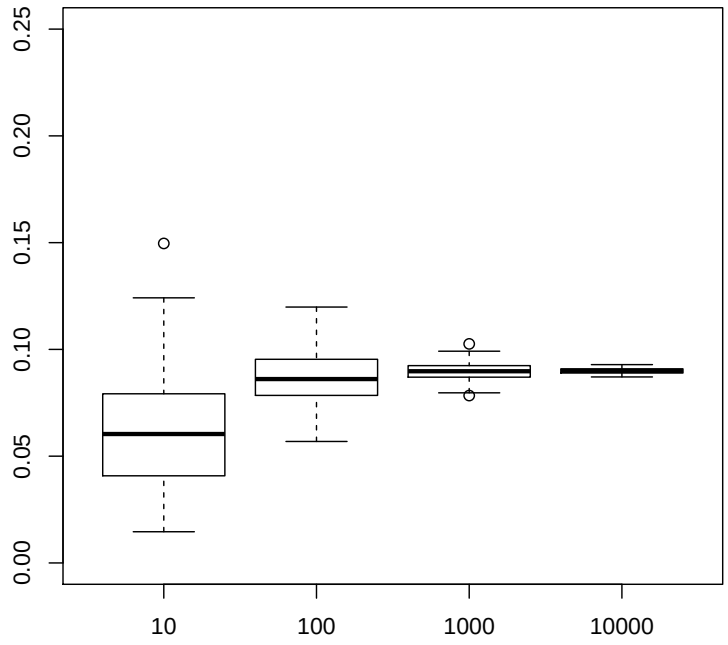




Different model classes have a different bias

Empirical risk for linear and quadratic models for the data distribution generated by a quadratic model with noise. The error does not go to zero.





Inherent noise in the data

Let $f : \mathbb{R}^n \to \mathbb{R}$ be the true dependence between the input $\boldsymbol{x}$ and the output y corrupted by the noise. Let $\mathcal{D}$ be the corresponding data distribution. Then we can consider the risk of the true model

$$R_{\text{true}} = \mathop{\mathbf{E}}_{\mathcal{D}}\big(L(f(\boldsymbol{x}), y)\big)$$

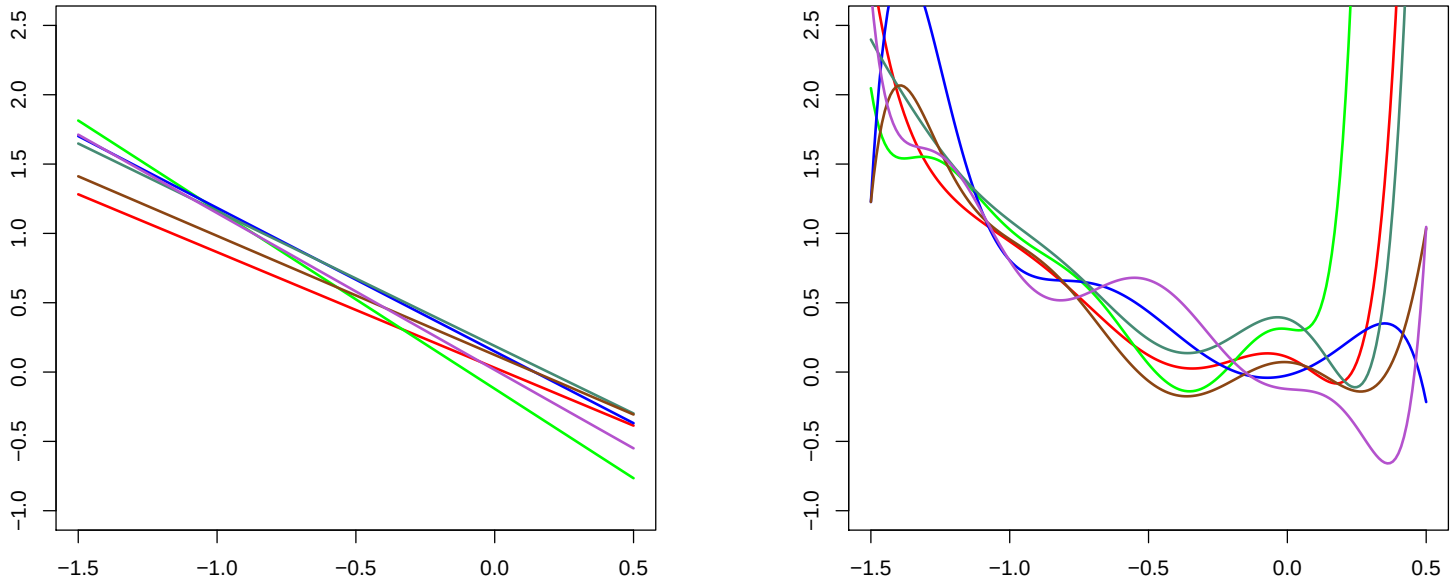
For a quadratic loss function and symmetric error distribution, the minimal achievable risk over all functions and the risk of the true model coincide.

We will refer to this inherent residual risk as *inherent noise*.

If additive error is distributed according to the normal distribution then the inherent noise is equal to the variance of the normal distribution.

Smallest bias is not always the best

Predictions of the model depend on the training set. A method is good if the inferred model is not radically different for different training sets.



Linear model on the left and 8th order polynomial on the right. Models are trained on 20 data points drawn form the quadratic model with the noise.

Bias-variance-noise decomposition

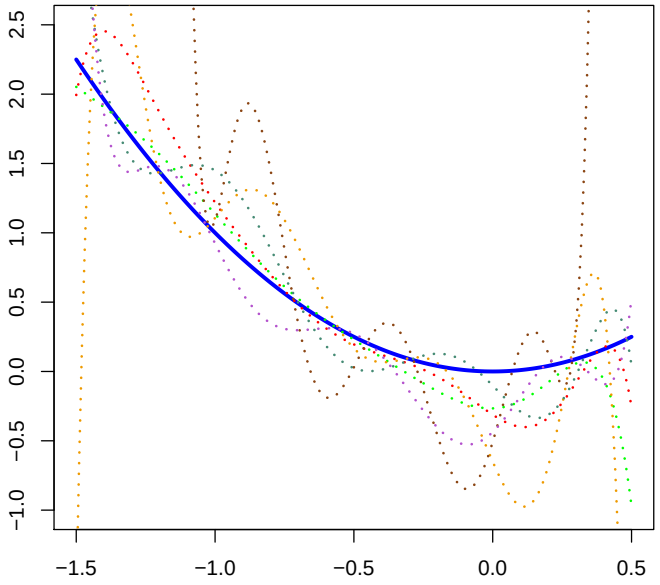
Let $\mathcal{S}$ denote training sample and let $\mathcal{A}_\mathcal{S} : \mathbb{R}^n \to \mathbb{R}$ be the predictor function inferred by the machine learning algorithm $\mathcal{A}$ based on the sample $\mathcal{S}$.

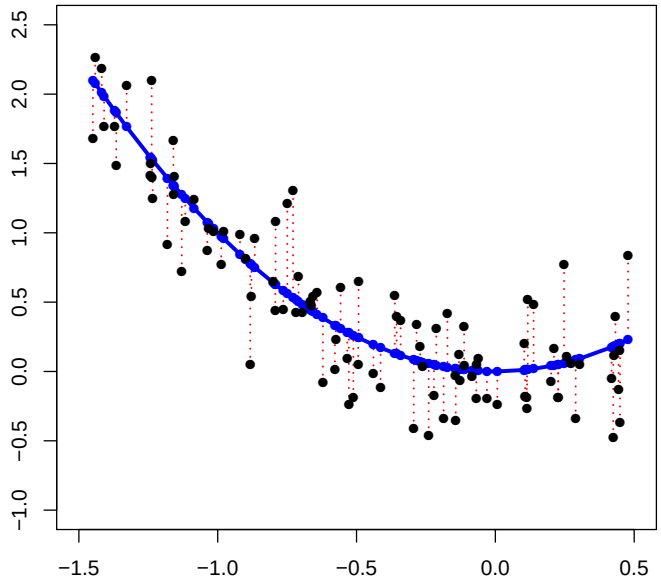
Assume that $f : \mathbb{R}^n \to \mathbb{R}$ be the true dependence between the input $\boldsymbol{x}$ and output y that is corrupted by additive noise that is independent form $f(\boldsymbol{x})$ and has zero mean. Now if we consider quadratic loss function, then the risk can be further decomposed:

$$\begin{aligned}\mathop{\mathbf{E}}_{\mathcal{D}}\big((\mathcal{A}_\mathcal{S}(\boldsymbol{x}) - y)^2\big) &= \mathop{\mathbf{E}}_{\mathcal{S}}\left[\mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\mathop{\mathbf{E}}_{y}\big((\mathcal{A}_\mathcal{S}(\boldsymbol{x}) - f(\boldsymbol{x}) + f(\boldsymbol{x}) - y)^2\big)\right]\right] \\ &= \underbrace{\mathop{\mathbf{E}}_{\mathcal{S},\boldsymbol{x}}\big((\mathcal{A}_\mathcal{S}(\boldsymbol{x}) - f(\boldsymbol{x}))^2\big)}_{\text{variability caused by } \mathcal{S}} + \underbrace{\mathop{\mathbf{E}}_{(\boldsymbol{x},y)}\big((f(\boldsymbol{x}) - y)^2\big)}_{\text{inherent noise}}\end{aligned}$$

Corresponding illustration

The left pane shows the fluctuations of inferred predictors. The right pane shows fluctuations of data points due to the noise.





Bias-variance-noise decomposition

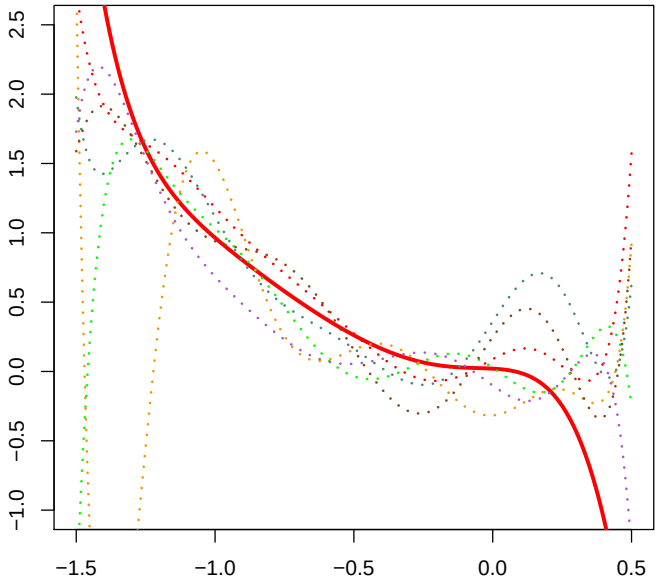
The component estimating the risk caused by the variability of training data can be further decomposed. Let $\mathcal{H}$ be the set of of potential prediction functions of the machine learning algorithm $\mathcal{A}$. Let $f \in \mathcal{H}$ be the optimal function among them. The average of the first term simplifies

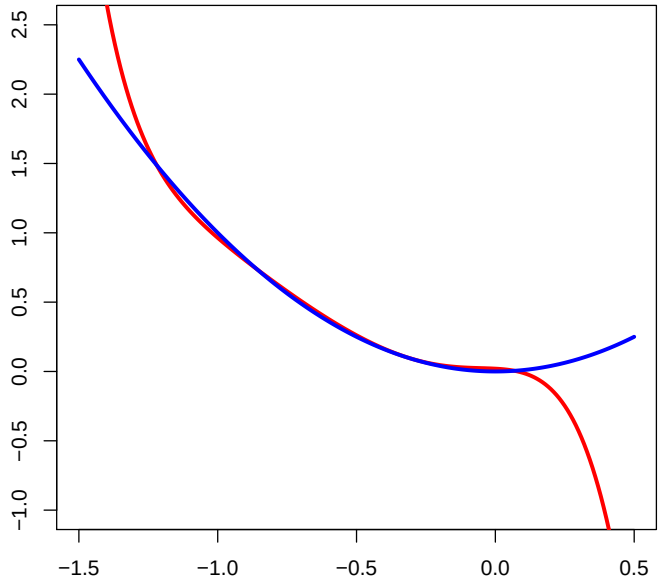
$$\begin{aligned}\mathop{\mathbf{E}}_{\mathcal{S},\boldsymbol{x}}\left((\mathcal{A}_{\mathcal{S}}(\boldsymbol{x}) - f(\boldsymbol{x}))^2\right) &= \mathop{\mathbf{E}}_{\mathcal{S},\boldsymbol{x}}\left((\mathcal{A}_{\mathcal{S}}(\boldsymbol{x}) - \overline{\mathcal{A}_{\mathcal{S}}}(\boldsymbol{x}) + \overline{\mathcal{A}_{\mathcal{S}}}(\boldsymbol{x}) - f(\boldsymbol{x}))^2\right) \\ &= \underbrace{\mathop{\mathbf{E}}_{\mathcal{S},\boldsymbol{x}}\left((\mathcal{A}_{\mathcal{S}}(\boldsymbol{x}) - \overline{\mathcal{A}_{\mathcal{S}}}(\boldsymbol{x}))^2\right)}_{variance} + \underbrace{\mathop{\mathbf{E}}_{\boldsymbol{x}}\left((\overline{\mathcal{A}_{\mathcal{S}}}(\boldsymbol{x}) - f(\boldsymbol{x}))^2\right)}_{bias}\end{aligned}$$

where $\overline{\mathcal{A}_{\mathcal{S}}}(\boldsymbol{x})$ is averaged prediction of $\mathcal{A}_{\mathcal{S}}(\boldsymbol{x})$ over all training sets $\mathcal{S}$.

Corresponding illustration

The left pane shows the fluctuations of inferred predictors compared to the averaged predictor. The right pane shows the difference between the actual model and the averaged prediction.





Bias-variance trade-off

- ▷ Simple models have small variance but high bias.
- ▷ Complex models have small bias but high variance.
- ▷ We need to balance both components with regularisation.

Minimal training error can be viewed as estimate on model bias

$$E_{tr} = \min_{f \in \mathcal{H}} R_m(f) \approx \min_{f \in \mathcal{H}} R(f) = \text{Bias}(\mathcal{H}|\mathcal{D}) \ .$$

Hence we somehow need to estimate the variance term in terms of model coefficients count or something similar.

Statistical Learning Theory

General treatment of generalisation error

Sampling bounds for training error

Let us consider finite set of functions $\mathcal{H} = \{f_1, \ldots, f_k\}$. The iid assumption allows us to find confidence intervals for $R(f_j)$. They are of type

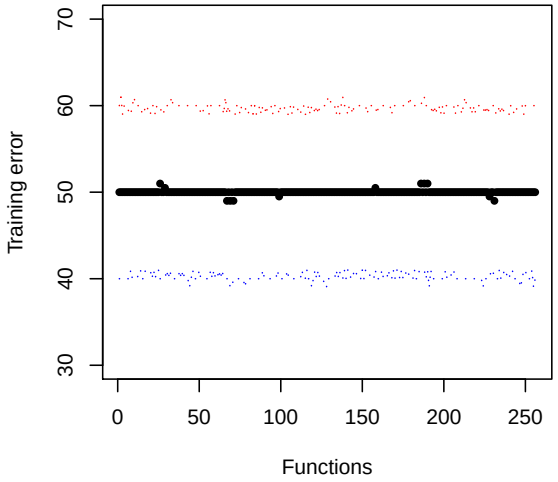
$$\Pr\left[|R_m(f_j) - R(f_j)| \geq \varepsilon\right] \leq c \cdot \exp\left(-\beta m^2 \varepsilon^2\right) =: \delta$$

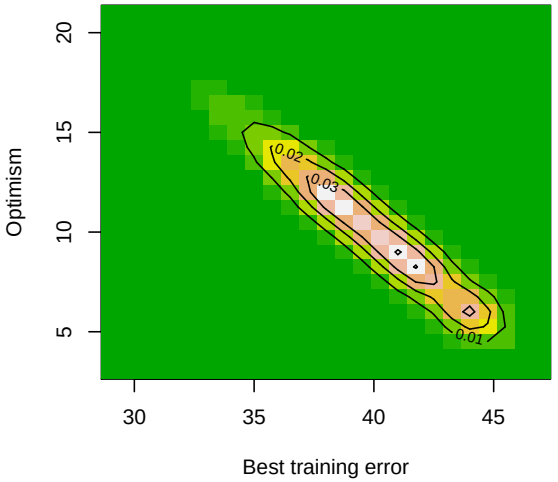
for some constants $c, \beta > 0$. They follow from Chernoff, Hoeffding or McDiarmid inequalities. Now applying the union bound we get

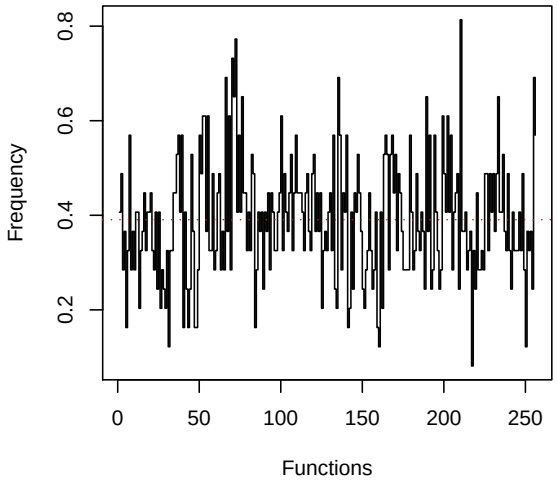
$$\Pr\left[\exists j : |R_m(f_j) - R(f_j)| \geq \varepsilon\right] \leq |\mathcal{H}| \cdot \delta \ .$$

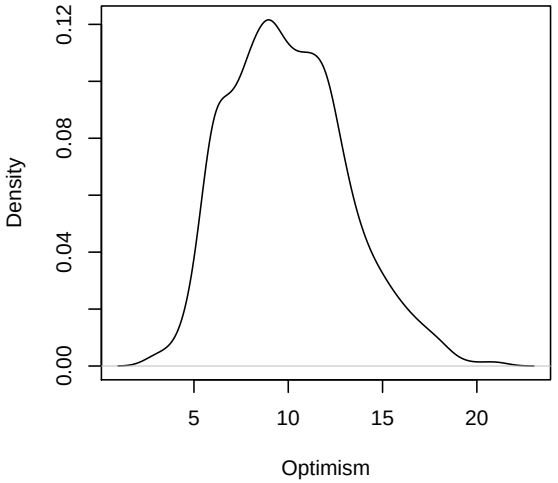
This inequality bounds optimism $\Delta = R_m(f_*) - R(f_*)$ for the proposed solution f_* that minimises training error. Since δ decreases as a function of m the method is asymptotically consistent.

Tightness of inequality. Bad case

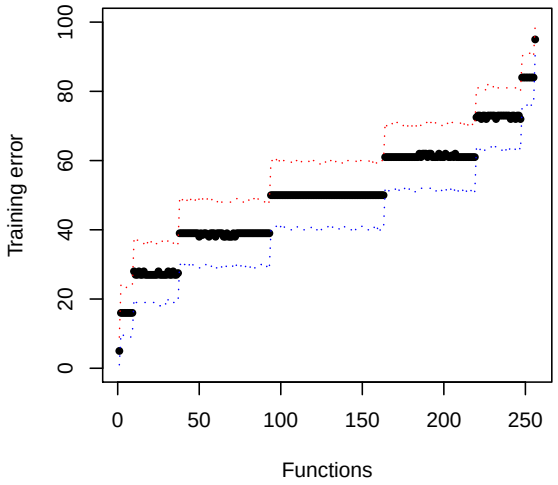


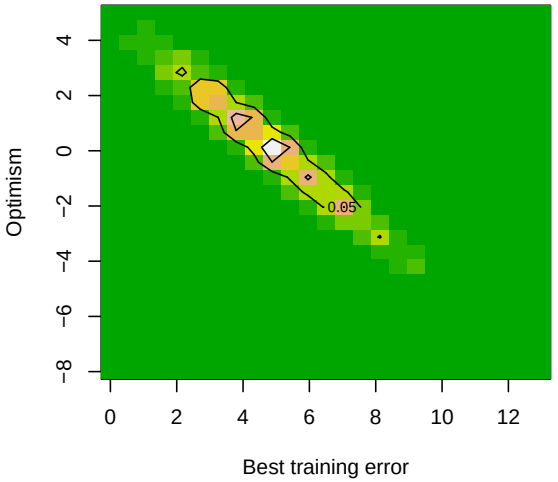


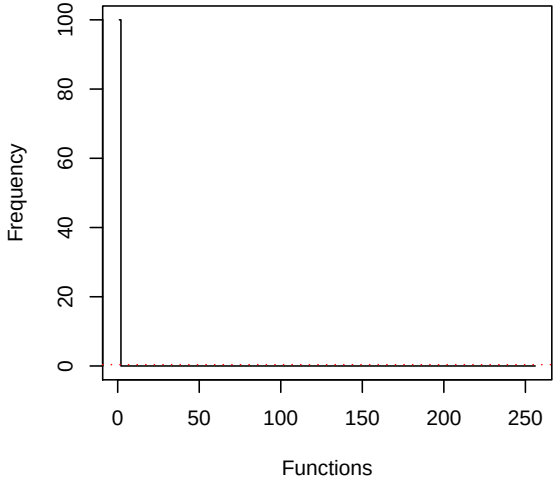


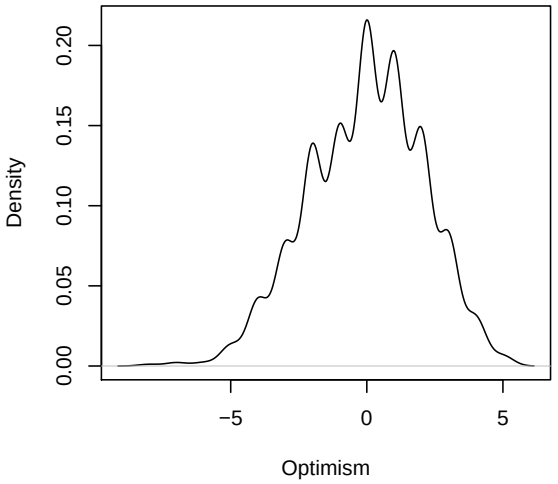


Tightness of inequality. Good case









Effective dimension of function families

- ▷ A good function family has few functions that are near optimal.
- ▷ Very similar functions *imply* many near-optimal solutions.

Vapnik-Chervonenkis dimension is one way to characterise flexibility of $\mathcal{H}$

The *VC dimension* of a function class $\mathcal{H}$ if the largest number of samples d for which one can find $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ such that any labelling $y_1, \ldots, y_d$ can be realised.

If the number of samples $m > d$ then only up to

$$G_{\mathcal{H}}(m) \leq \left(\frac{em}{d}\right)^d$$

labellings can be implemented. For $m > 2d$, $\mathcal{H}$ is really sparse net.

Uniform bound on the optimism

Suppose you have two independent m element sets $\mathcal{S}_1$ and $\mathcal{S}_2$ and you optimise classifier on one and test it on other how to estimate optimism.

- ▷ Although the number of functions in $\mathcal{H}$ is potentially infinite, the number of realisable labellings in the $2m$ element data set $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$ is

$$G_{\mathcal{H}}(2m) \leq \left(\frac{2em}{d}\right)^d$$

  where d is the VC dimension of $\mathcal{H}$.
- ▷ Each split $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$ and labelling ℓ defines empirical errors

$$R^1_m(\mathcal{S}_1, \mathcal{S}_2, \ell) \qquad \text{and} \qquad R^2_m(\mathcal{S}_1, \mathcal{S}_2, \ell)$$

  and in training we choose ℓ that minimises $R^1_m(\ell)$.

Bad splits for each labelling

- Lets consider worst case $R^1_m(\ell) = 0$ and $R^2_m(\ell) = 1$.
- For a fixed labelling ℓ, the number of splits where all errors are in the second term is at most 1.
- As all splits are equiprobable, the probability of such a bad split is

$$1 \Big/ \binom{2m}{m} = \mathrm{O}(2^{-m/2})$$

- Thus the probability of large optimism is decreasing exponentially

$$\Pr\left[\exists f \in \mathcal{H} : R^1_m(f) = 0 \wedge R^2_m(f) = 1\right] \leq \left(\frac{em}{d}\right)^d \mathrm{O}(2^{-\frac{m}{2}}) = \mathrm{O}(2^{-\frac{m}{2}})$$

More advanced SLT bound

More careful analysis gives you the bound

$$\Pr\left[\forall f \in \mathcal{H} : R(f) \leq R_m(f) + c(\delta)\sqrt{\frac{\mathsf{VCdim}(\mathcal{H})}{n}}\right] \geq 1 - \delta$$

Alternatives to VC-dimension

There are many alternatives to VC-dimension:

- ▷ Rademacher complexity

$$\mathcal{R}_m(\mathcal{H}|\mathcal{S}) = \mathbf{E}\left[\sup_{f\in\mathcal{H}} \frac{2}{m} \cdot \sum_{i=1}^{m} \sigma_i f(\boldsymbol{x}_i)\right] \qquad \text{for} \qquad \sigma_i \in \{-1, 1\}$$

$$\mathcal{R}_m(\mathcal{H}) = \mathop{\mathbf{E}}_{\mathcal{S}\leftarrow\mathcal{D}}(\mathcal{R}_m(\mathcal{H}|\mathcal{S}))$$

- ▷ Covering numbers

They are important since sometimes we cannot compute VC-dimension while the other complexity measures do exist and are easy to compute.

NFL Theorems

Why do we need NFL theorems?

No Free Lunch theorems show the following important facts

- ◇ **There are no universal solutions.** All machine learning methods are *equally good* if we do not place *strong* assumptions on the input data.

- ◇ **Benchmarks can lie:** For each machine learning algorithm there exists a *sample or sample class* where it outperforms some other method.

- ◇ **All clever bounds have fine print conditions:** Sampling bounds and SLT results are *correct* but do not tell *exactly* what we need and hope.

We state NFL theorems in maximally simple way to show their banal nature.

Setup for the NFL theorem

Consider a finite learning problem.

- ▷ Let $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$ be the training set.
- ▷ Let $\mathcal{T} = \{(\boldsymbol{x}_{m+1}, y_{m+1}), \ldots, (\boldsymbol{x}_n, y_n)\}$ be the test data.
- ▷ Assume that the inputs of the training and test data are *disjoint*.
- ▷ Let $\mathcal{A}_{\mathcal{S}}(\boldsymbol{x})$ prediction of machine learning method on $\boldsymbol{x}$.

How large is the test error (*out-of-training sample error*)?

$$R_{ot}(\mathcal{A}) = \frac{1}{n-m} \cdot \sum_{i=m+1}^{m} [\mathcal{A}_{\mathcal{S}}(\boldsymbol{x}_i) \neq y_i]$$

The simplest version of NFL theorem

Theorem I. If we assume that labels y_i are computed as $f(\boldsymbol{x}_i)$ for a function chosen uniformly form all possible functions then

$$\mathop{\mathbf{E}}_{f}\left(R_{ot}(\mathcal{A}|f)\right) = \frac{1}{2} \ .$$

Proof

▷ Let $f^c(\boldsymbol{x}_{m+i}) = \neg f(\boldsymbol{x}_{m+i})$ denote the complement of f over $\mathcal{T}$. Then

$$R_{ot}(\mathcal{A}|f) + R_{ot}(\mathcal{A}|f^c) = 1$$

▷ As f and f° have equal weight $\Pr[f|\mathcal{S}] = \Pr[f|\mathcal{S}]$, we get

$$\underset{f}{\mathbf{E}}\left(R_{ot}(\mathcal{A}|f)\right) = \frac{1}{2} \cdot \sum_{f:X\to Y} \Pr[f|\mathcal{S}]\left(R_{to}\left(\mathcal{A}|f\right) + R_{ot}(\mathcal{A}|f^{c})\right) = \frac{1}{2}$$

Obvious conclusions

The claim can be extended to different loss functions L and function sets $\mathcal{F}$ as long as the symmetry condition on the test set is preserved:

- ▷ We can define complement operator such that

$$\forall f \in \mathcal{F}: \ \exists f^{\circ} \in \mathcal{F} \wedge \forall f \in \mathcal{F} : (f^{\circ})^{\circ} = f$$

- ▷ Corresponding losses are constant for any prediction

$$L(y, f(\boldsymbol{x}_{m+i})) + L(y, f^{\circ}(\boldsymbol{x}_{m+i})) = g(\boldsymbol{x}_{m+i})$$

- ▷ Probabilities in complement pairs are equal

$$\Pr[f|\mathcal{S}] = \Pr[f^{\circ}|\mathcal{S}]$$

NFL theorem for problem classes

- ▷ In real life tasks, some problems are harder than others.
- ▷ Let $\mathcal{F}$ denote probability distribution over all functions
- ▷ Which problem instances $\mathcal{F}$ the algorithm $\mathcal{A}$ can handle?

Theorem II. Consider a uniform distribution over all possible problem classes. Then

$$\mathop{\mathbf{E}}_{\mathcal{F}} \left(\mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} (R_{ot}(\mathcal{A})) \right) = \frac{1}{2} \ .$$

As a result, no algorithm can be better than the other—there are some problem classes for which the other algorithm is better.

Concentration equation for NFL

By combining Theorem II and Law of large numbers we obtain the following surprising result about benchmarking.

Theorem III. Let $\mathcal{F}_1, \ldots, \mathcal{F}_k$ be the set of problem classes chosen uniformly over the design space. Then

$$\frac{1}{k} \cdot \sum_{j=1}^{k} \underset{f \leftarrow \mathcal{F}_j}{\mathbf{E}} \left(R_{ot}(\mathcal{A}) \right) \approx \frac{1}{2}$$

for any learning algorithm $\mathcal{A}$ and thus for any two learning algorithms $\mathcal{A}$ and $\mathcal{B}$ there exist exist a particular example problem

$$\underset{f \leftarrow \mathcal{F}_j}{\mathbf{E}} \left(R_{ot}(\mathcal{A}) \right) \lesssim \underset{f \leftarrow \mathcal{F}_j}{\mathbf{E}} \left(R_{ot}(\mathcal{B}) \right) \quad .$$

Wait SLT results contradict NFL theorems!

NFL theorem says that you cannot learn arbitrary function. On the same time, SLT bounds assure that $R_m(f) + \Delta$ is rather accurate estimate.

What happens then?

- ▷ We can assume that n large enough so that $R(f) \approx R_{ot}(f)$.
- ▷ For most functions $R_m(f) + \Delta \approx 0.5$ and we do not care.
- ▷ For few functions $R_m(f) + \Delta \approx 0.0$
  - ◇ For half of these functions $R_{ot}(f) \approx 0.0$ and we are happy.
  - ◇ For half of these functions $R_{ot}(f) \approx 1.0$ and we are happy.

Small print notice. For fixed error estimate SLT bounds say nothing.

- ▷ Rejection of uninteresting results creates bias!
- ▷ We must additionally assume that the learning task is easy.