GI01/M055: Supervised Learning

Statistical Learning Theory

Massimiliano Pontil

(based on notes by John Shawe-Taylor)

Today’s plan

- Model selection problem
- Generalisation error bounds
- Error bound analysis
- VC–dimension
- Bias/variance trade-off

Supervised learning model (I) – review

$P(\mathbf{x}, y)$: **fixed but unknown** probability density (defines the learning environment)

Expected error:

$$\mathcal{E}(f) := \mathbf{E}\left[V(y, f(\mathbf{x}))\right] = \int V(y, f(\mathbf{x}))dP(\mathbf{x}, y)$$

where $V : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a **loss function**

- Our goal is to minimize $\mathcal{E}$
- Optimal solution: $f^* := \operatorname{argmin}_f \mathcal{E}(f)$
- We cannot compute f^* because P is unknown

Supervised learning model (II)

We have encountered different loss functions:

- square loss: $V(y, z) = (y - z)^2$
- misclassification loss (0-1 loss): $V(y, z) = 1$ if $y \neq z$ and zero otherwise (here $y, z \in \{c_1, \ldots, c_K\}$)
- logistic regression: $V(y, z) = y \log(1 + e^{-z}) + (1 - y) \log(1 + e^{z})$

Supervised learning model (III)

$P(\mathbf{x}, y)$ is unknown $\Rightarrow$ cannot compute f^*

We are only given an i.i.d. sample from P:

$$S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$$

We approximate (replace) the expected error $\mathcal{E}(f)$ with the **empirical error**

$$\mathcal{E}_S(f) := \frac{1}{m} \sum_{i=1}^{m} V(y_i, f(\mathbf{x}_i))$$

Supervised learning model (IV)

If we minimize $\mathcal{E}_S$ over a sufficiently large set of functions, we can always find a function f with zero empirical error!
But $\mathcal{E}(f)$ may be far away from zero! (**overfitting**)

We introduce a restrictive class of functions $\mathcal{H}$ (**hypothesis space**) and minimize $\mathcal{E}_S$ within $\mathcal{H}$. That is, our learning algorithm is:

$$f_S = \mathsf{argmin}_{f\in\mathcal{H}}\mathcal{E}_S(f)$$

Linear regression: $\mathcal{H} = \{f(\mathbf{x}) = \mathbf{w}^\top\mathbf{x} : \mathbf{w} \in \mathbb{R}^d\}$

Regularization

We minimize the penalized (regularized) empirical error

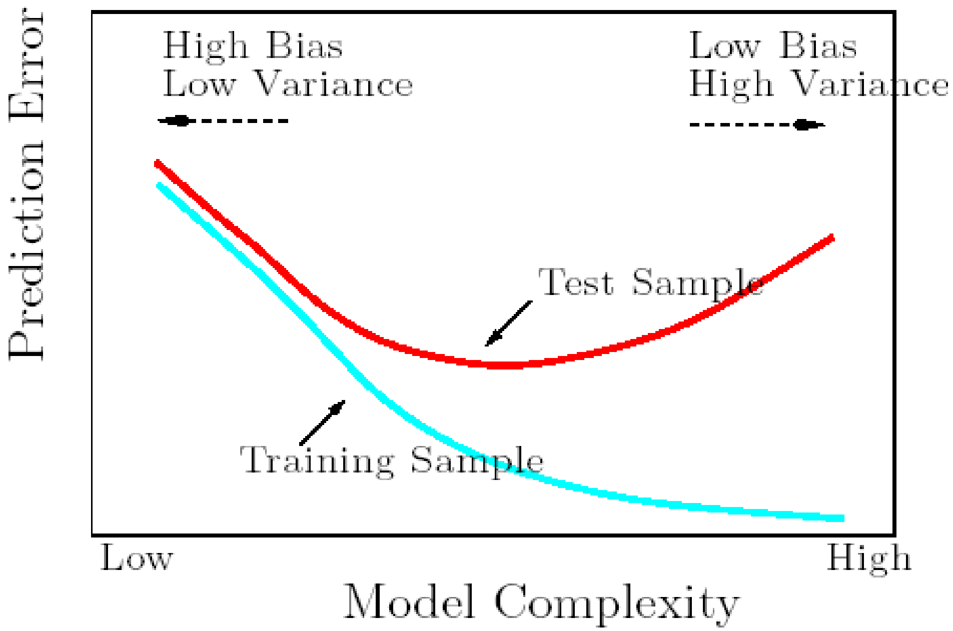
$$\mathcal{E}_{S,\lambda}(f) := \frac{1}{m}\sum_{i=1}^{m} V(y_i, f(\mathbf{x}_i)) + \lambda R(f), \qquad \lambda > 0$$

Ridge regression: $V(y,z) = (y-z)^2$, $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$, $R(f) = \|\mathbf{w}\|^2$

This is similar to empirical error minimization in the hypothesis space $\mathcal{H}_A = \{f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} : \|\mathbf{w}\| \leq A\}$ for some $A > 0$

(this connection can be made formal: given $A > 0$ there is $\lambda(A) > 0$ such that ridge regression gives the same solution as empirical error minimization in $\mathcal{H}_A$ and vice versa, given λ there is $A(\lambda)$ such that...)

Model complexity and overfitting



- If we pick the best model (learning algorithm) by minimizing the training error we overfit the data
- We wish to estimate the expected (test) error

Model selection and assessment

- **Model selection:** aims at estimating the performance of different models (learning algorithms) in order to choose the (approximately) best one, for example:

  - best hypothesis space among many possible ones
  - best λ in ridge regression
  - best k in k–NN etc.

- **Model assessment:** having chosen a final model, we wish to estimate its expected error (aka generalization) on new data

Model selection and assessment (cont.)

If we have a large set of examples, a natural approach is to split the data in three parts: training, validation and test set

1. Use the training set to fit the models (train different learning algorithms on it)

2. Use the validation set for model selection. So the best model is the one which minimizes the validation error

3. Use the test set for assessment of the expected error of the best model above

Typically, we keep most of the data for training (say 1/2 for training, 1/4 for validation and testing)

Generalization error bound

- Often, we have only few examples. Can we choose a model and assess its expected error directly (without splitting the training data)?

- **Learning theory** studies conditions which ensure the **predictivity** of a learning algorithm:
  - The expected error is close to the empirical error
  - The expected error decreases when the number of samples increases

- We discuss a central approach in the theory which allows us to **relate the training error to the generalization error**

General statistical considerations cont.

- Usually the distribution subsumes the processes of the natural world that we are studying

- We assumes that we are given a ‘training sample’ or ‘training set’

$$S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$$

  generated identically and independently (i.i.d.) according to the distribution P

Generalisation of a learner

- Assume that we have a learning algorithm $\mathcal{A}$ that chooses a function $\mathcal{A}_{\mathcal{F}}(S)$ from a function space $\mathcal{F}$ in response to the training set S

- From a statistical point of view the quantity of interest is the random variable:

$$\epsilon(S, \mathcal{A}, \mathcal{F}) = \mathbb{E}_{(\mathbf{x},y)} \left[V(\mathcal{A}_{\mathcal{F}}(S)(\mathbf{x}), y) \right],$$

  where V is a ‘loss’ function that measures the discrepancy between $\mathcal{A}_{\mathcal{F}}(S)(\mathbf{x})$ and y

Generalisation of a learner

- For example, in the case of classification V is 1 if the two disagree and 0 otherwise, while for regression it could be the square of the difference between $\mathcal{A}_{\mathcal{F}}(S)(\mathbf{x})$ and y

- We refer to the random variable $\epsilon(S, \mathcal{A}, \mathcal{F})$ as the generalisation of the learner

Example of Generalisation I

- We consider the Breast Cancer dataset from the UCI repository

- Use a simple kind of Parzen window classifier: weight vector is

$$\mathbf{w}^+ - \mathbf{w}^-$$

where $\mathbf{w}^+$ is the average of the positive training examples and $\mathbf{w}^-$ is average of negative training examples. Threshold is set so hyperplane bisects the line joining these two points

Example of Generalisation II

- Given a size m of the training set, by repeatedly drawing random training sets S we estimate the distribution of

$$\epsilon(S, \mathcal{A}, \mathcal{F}) = \mathbb{E}_{(\mathbf{x},y)} \left[V(\mathcal{A}_{\mathcal{F}}(S)(\mathbf{x}), y) \right],$$

by using the test set error as a proxy for the true generalisation

- We plot the histogram and the average of ϵ for various sizes of training set – initially the whole dataset gives a single value if we use training and test as the all the examples, but then we plot for training set sizes:

$$342, 273, 205, 137, 68, 34, 27, 20, 14, 7$$

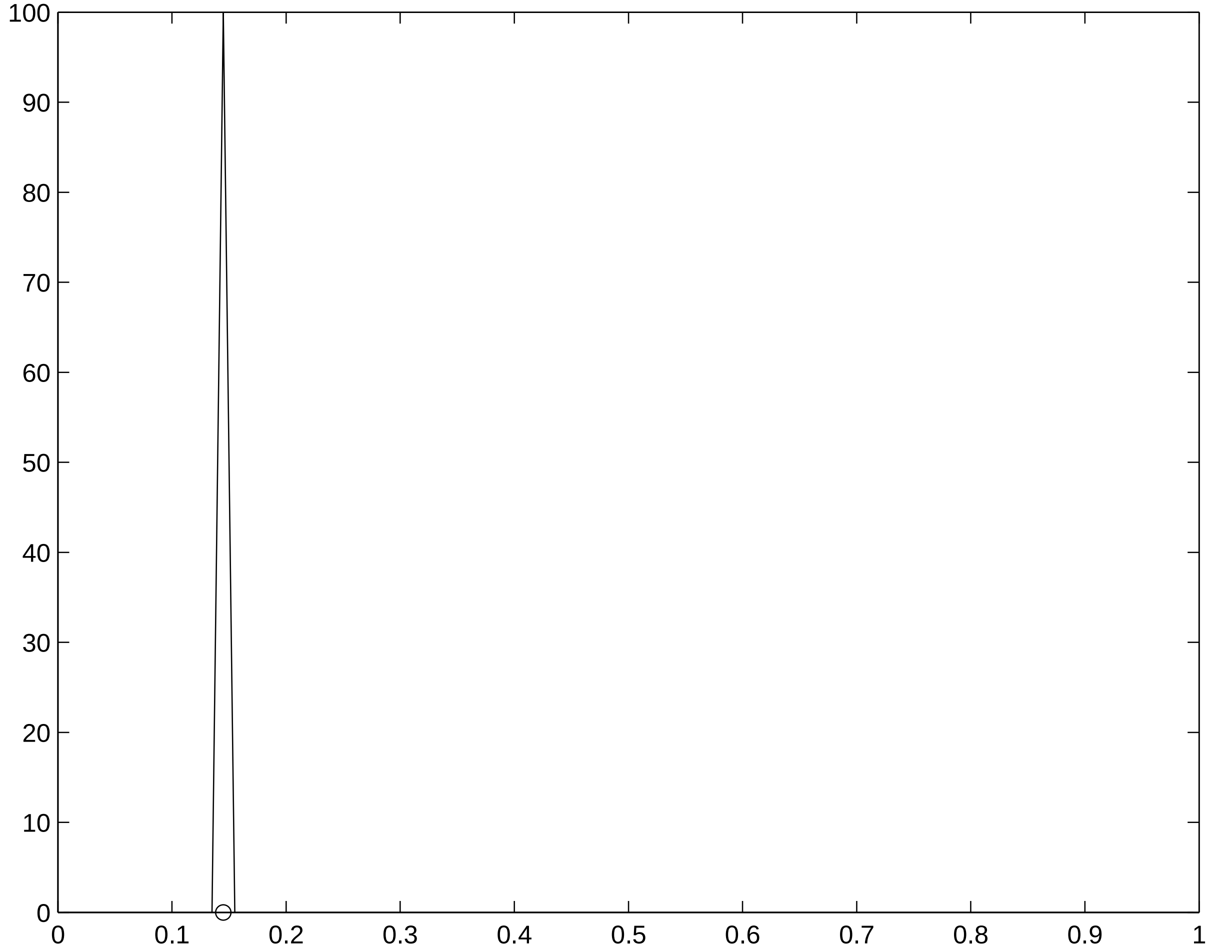
Example of Generalisation III

- Since the expected classifier is in all cases the same:

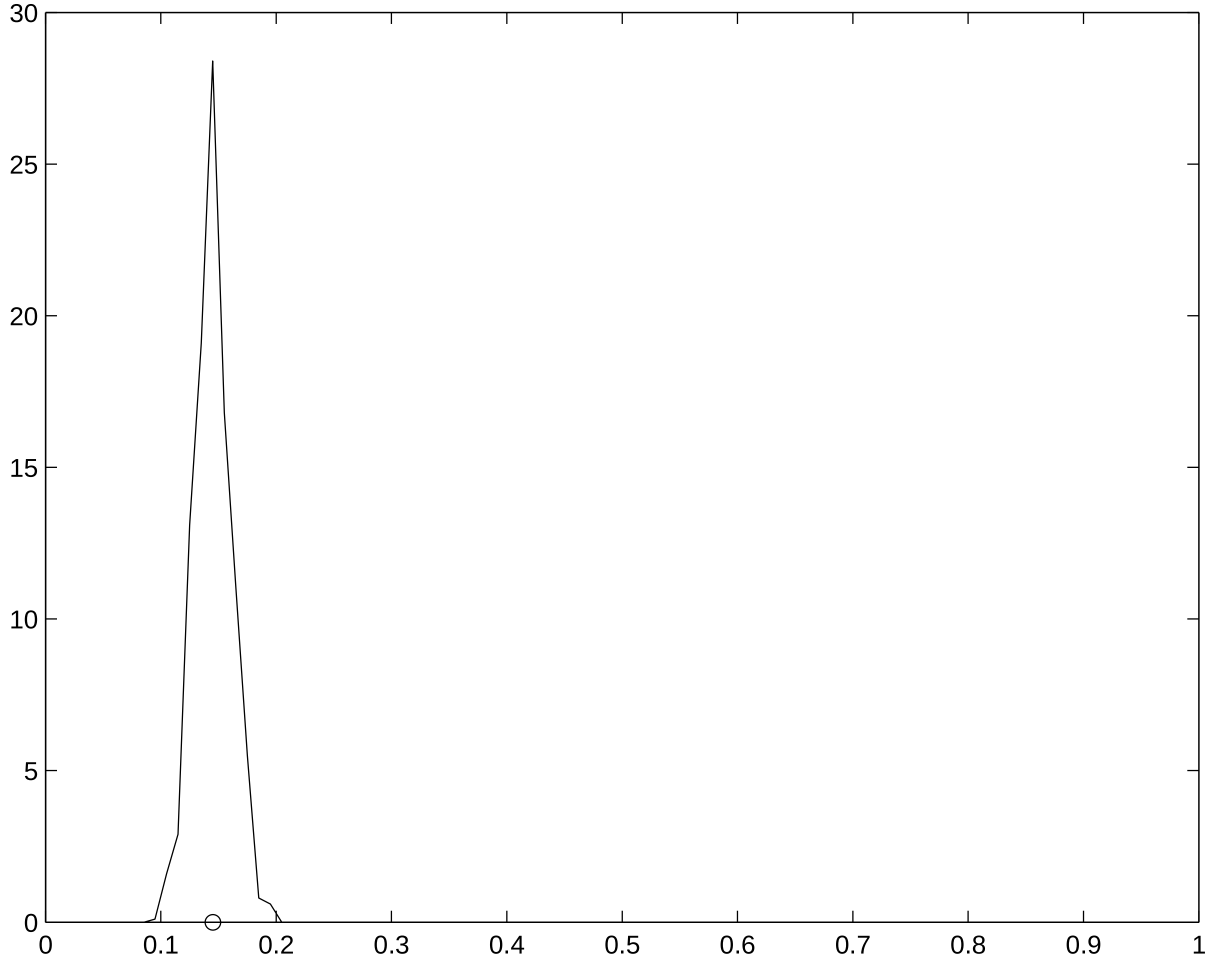
$$\begin{aligned}\mathbb{E}\left[\mathcal{A}_{\mathfrak{F}}(S)\right] &= \mathbb{E}_S\left[\mathbf{w}_S^+ - \mathbf{w}_S^-\right] \\ &= \mathbb{E}_S\left[\mathbf{w}_S^+\right] - \mathbb{E}_S\left[\mathbf{w}_S^-\right] \\ &= \mathbb{E}_{y=+1}\left[\mathbf{x}\right] - \mathbb{E}_{y=-1}\left[\mathbf{x}\right],\end{aligned}$$

  we do not expect large differences in the average of the distribution, though the non-linearity of the loss function means they won't be the same exactly

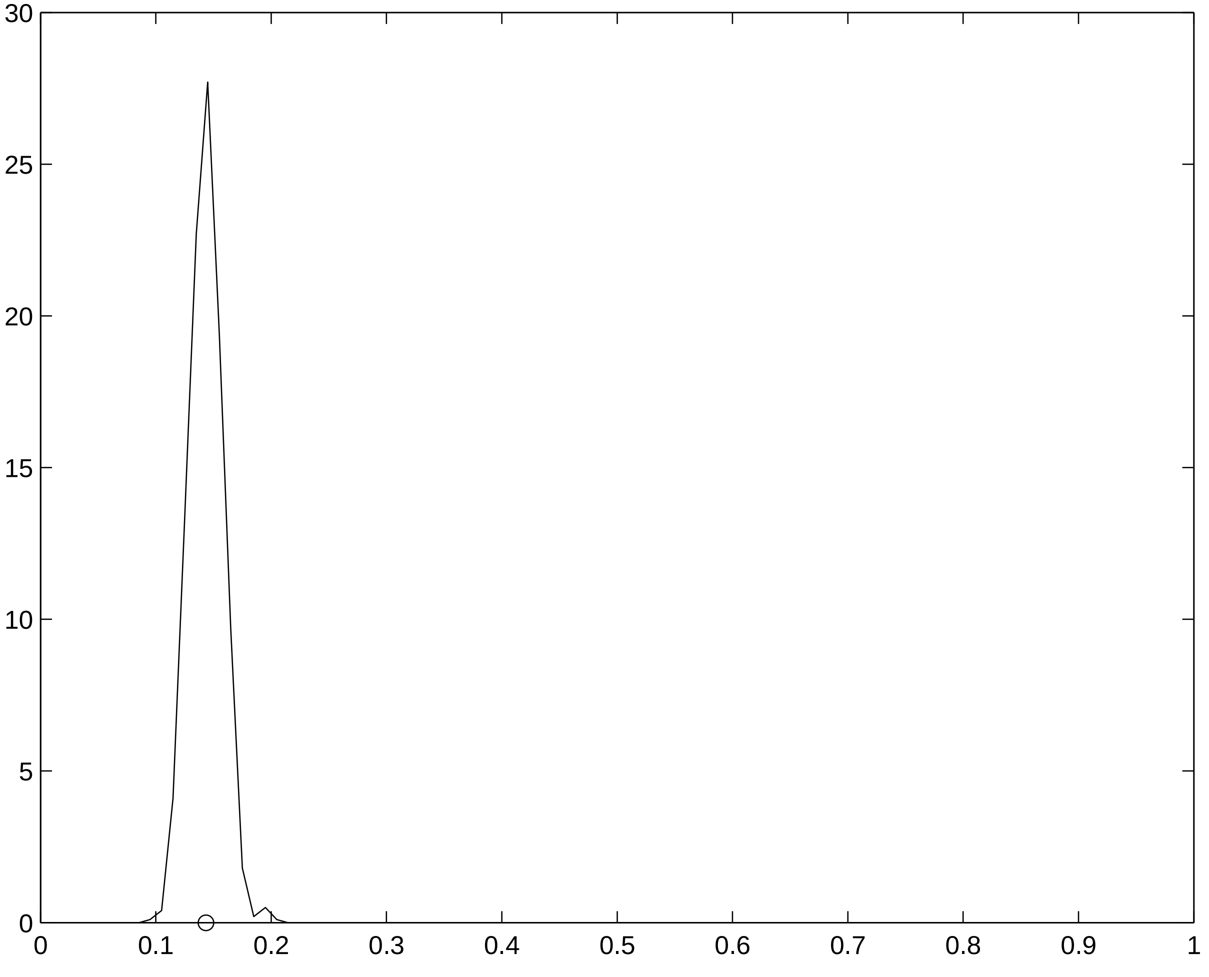
Error distribution: full dataset



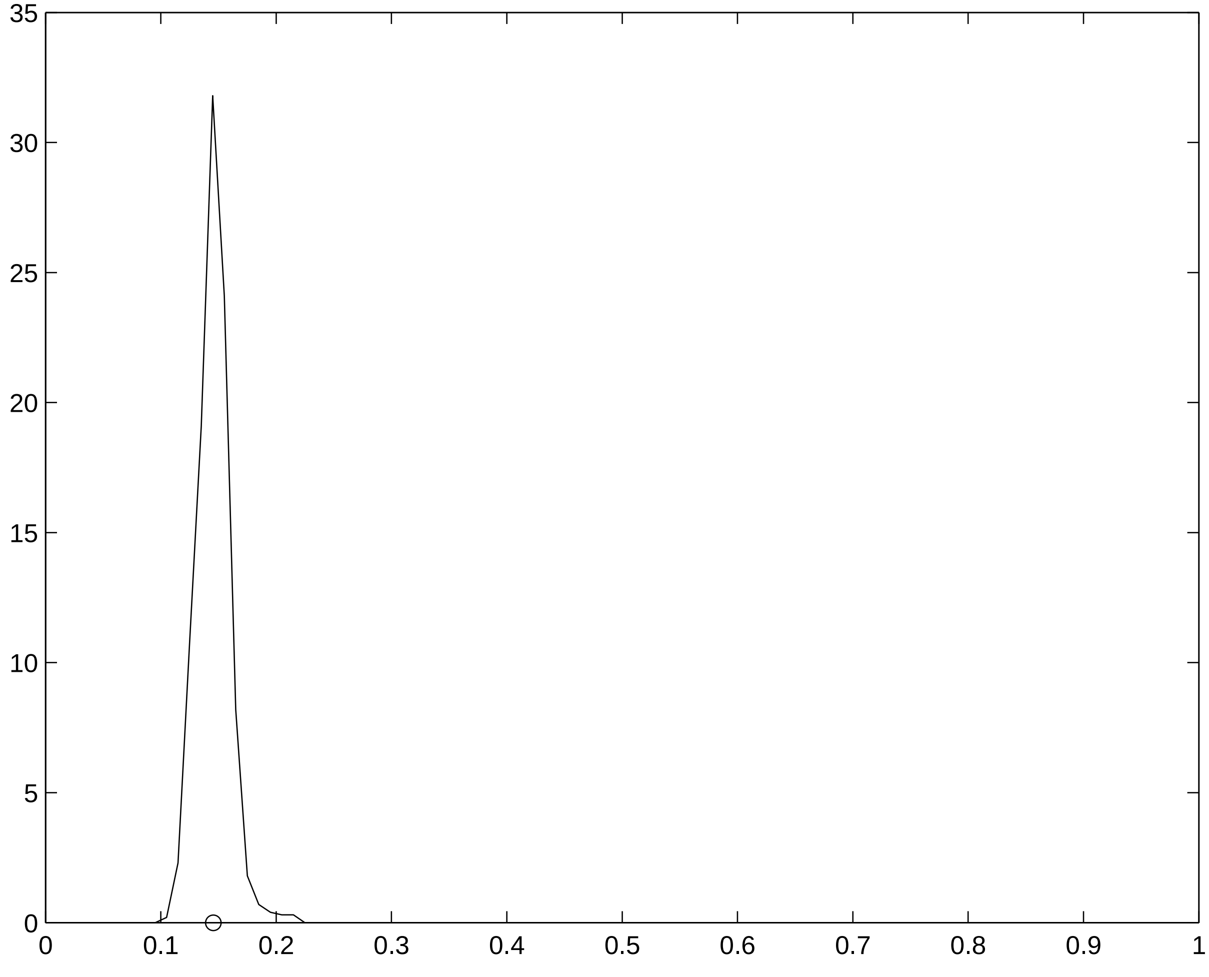
Error distribution: dataset size: 342



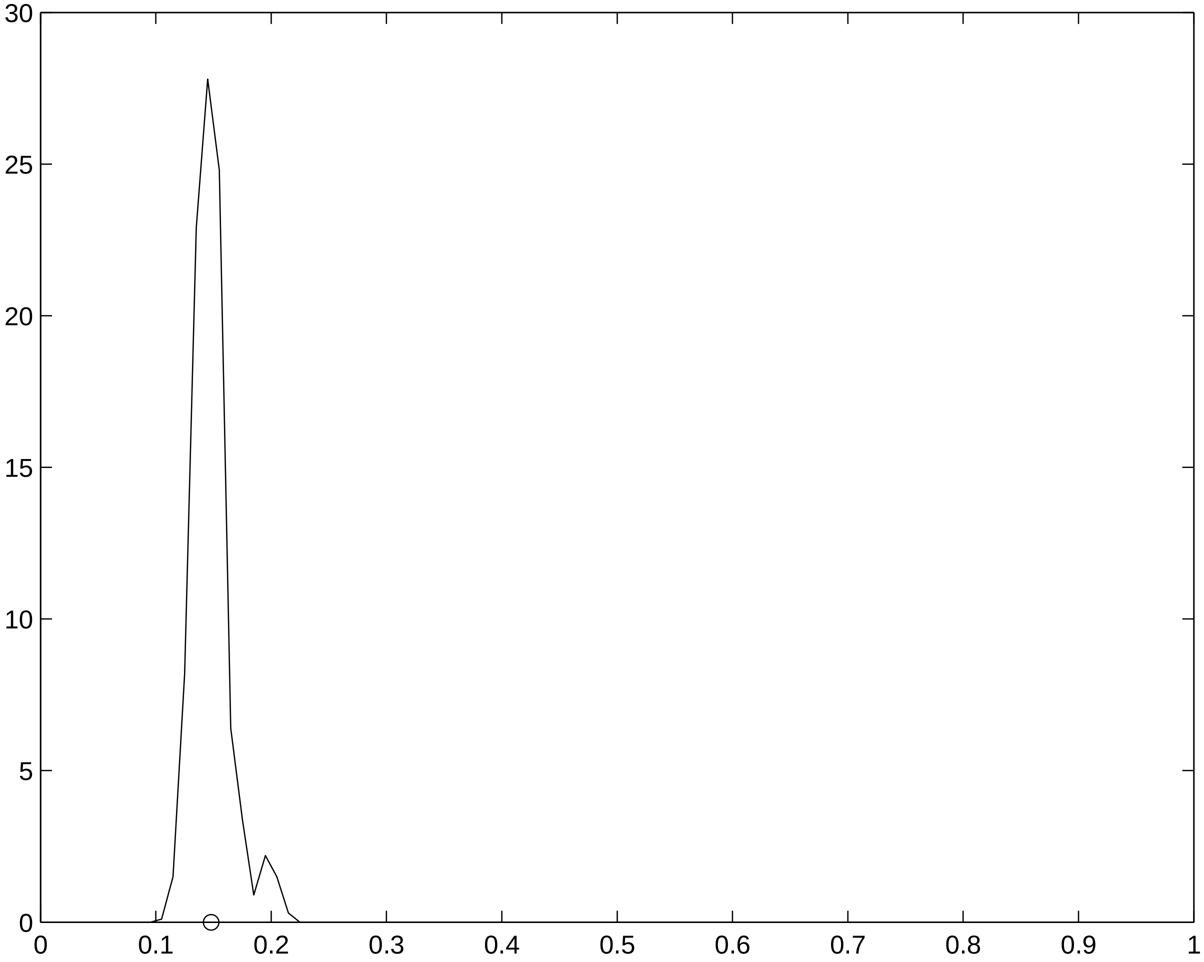
Error distribution: dataset size: 273



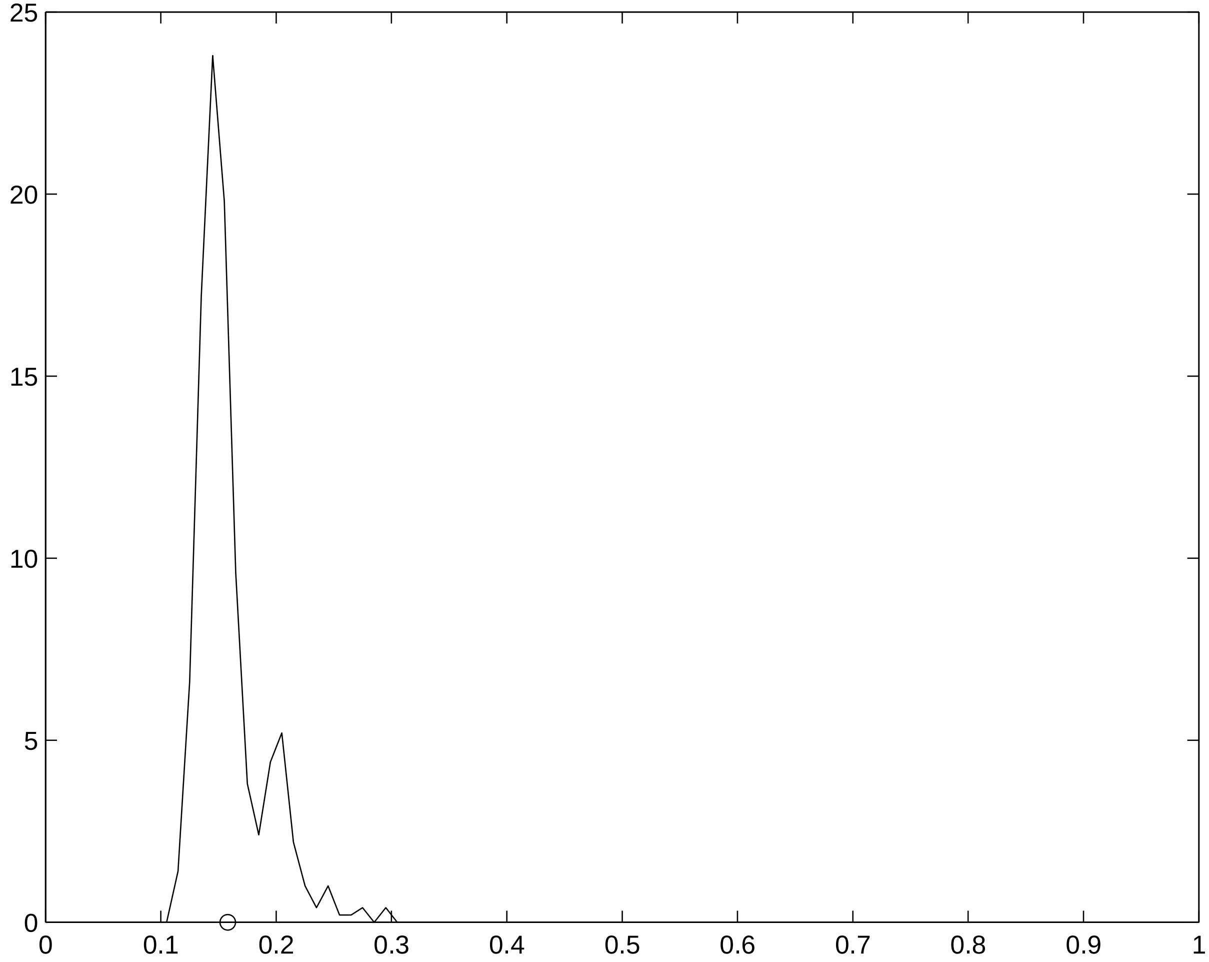
Error distribution: dataset size: 205



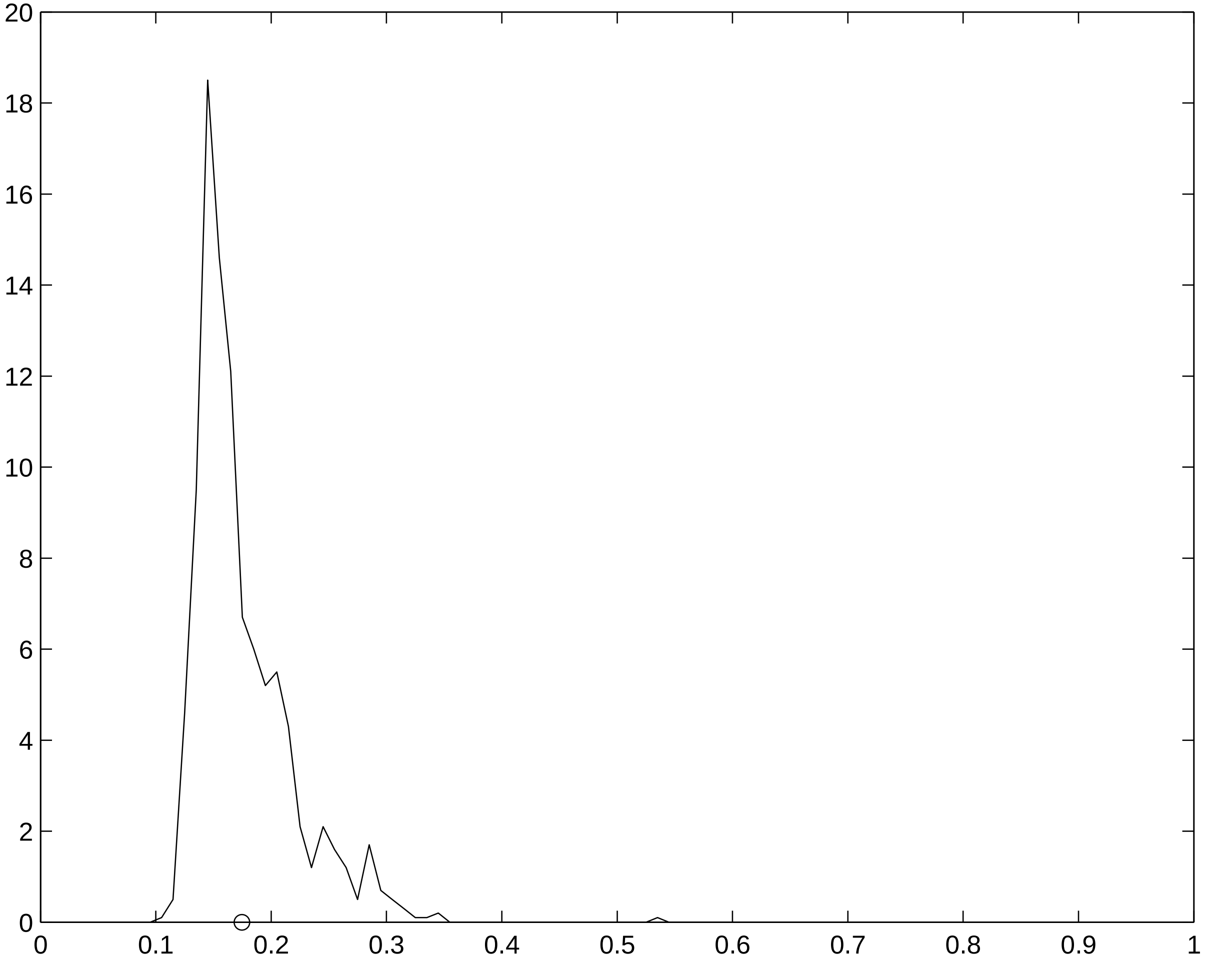
Error distribution: dataset size: 137



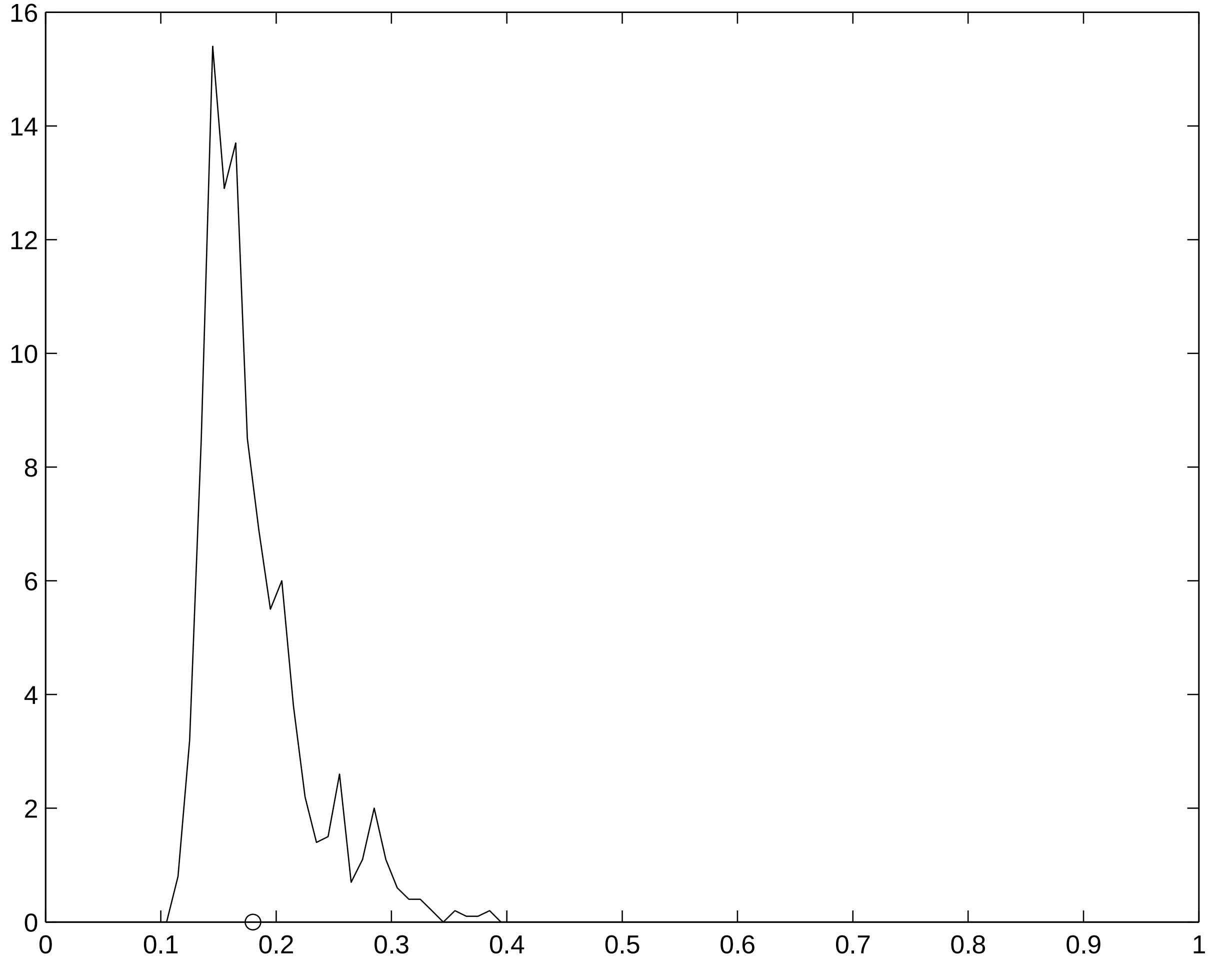
Error distribution: dataset size: 68



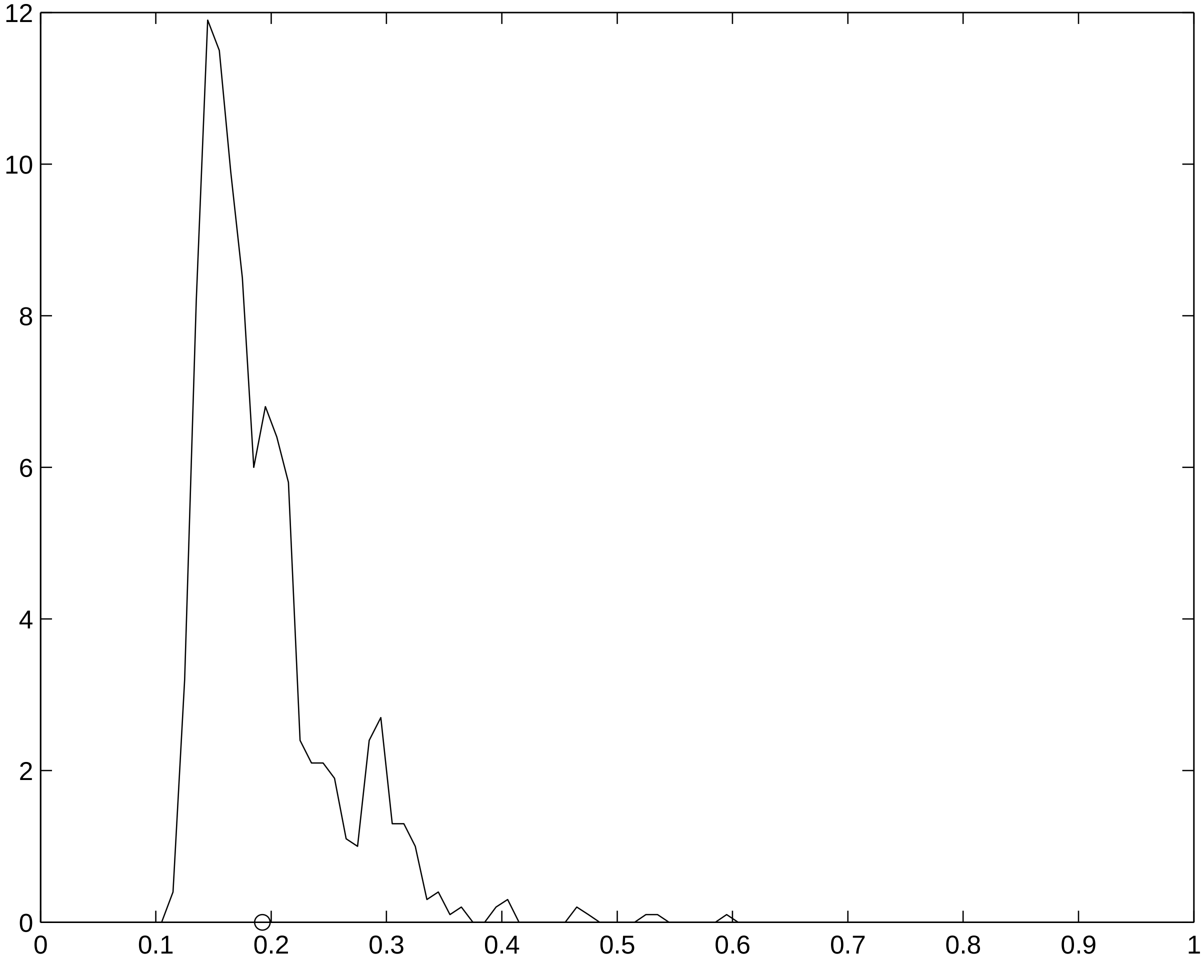
Error distribution: dataset size: 34



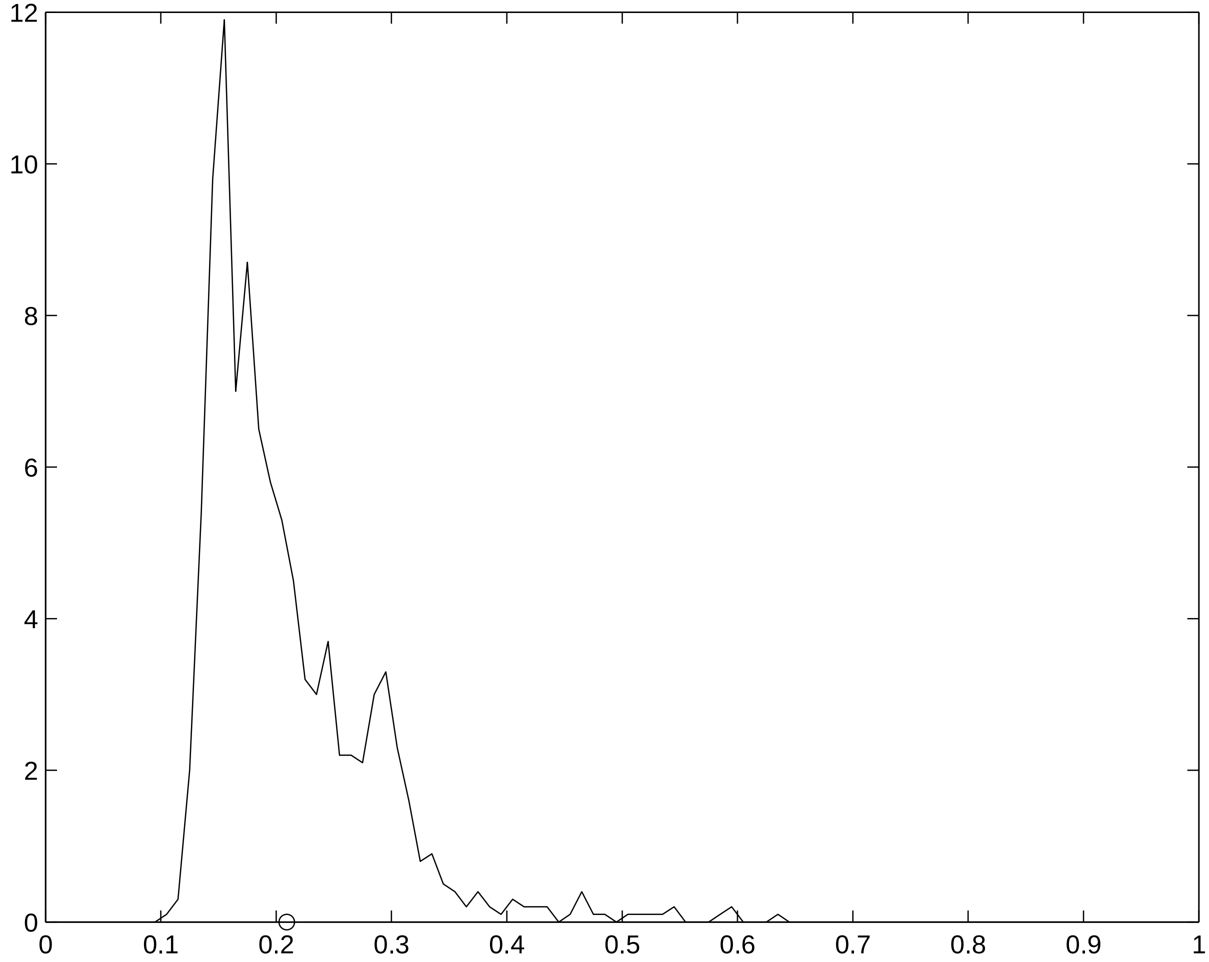
Error distribution: dataset size: 27



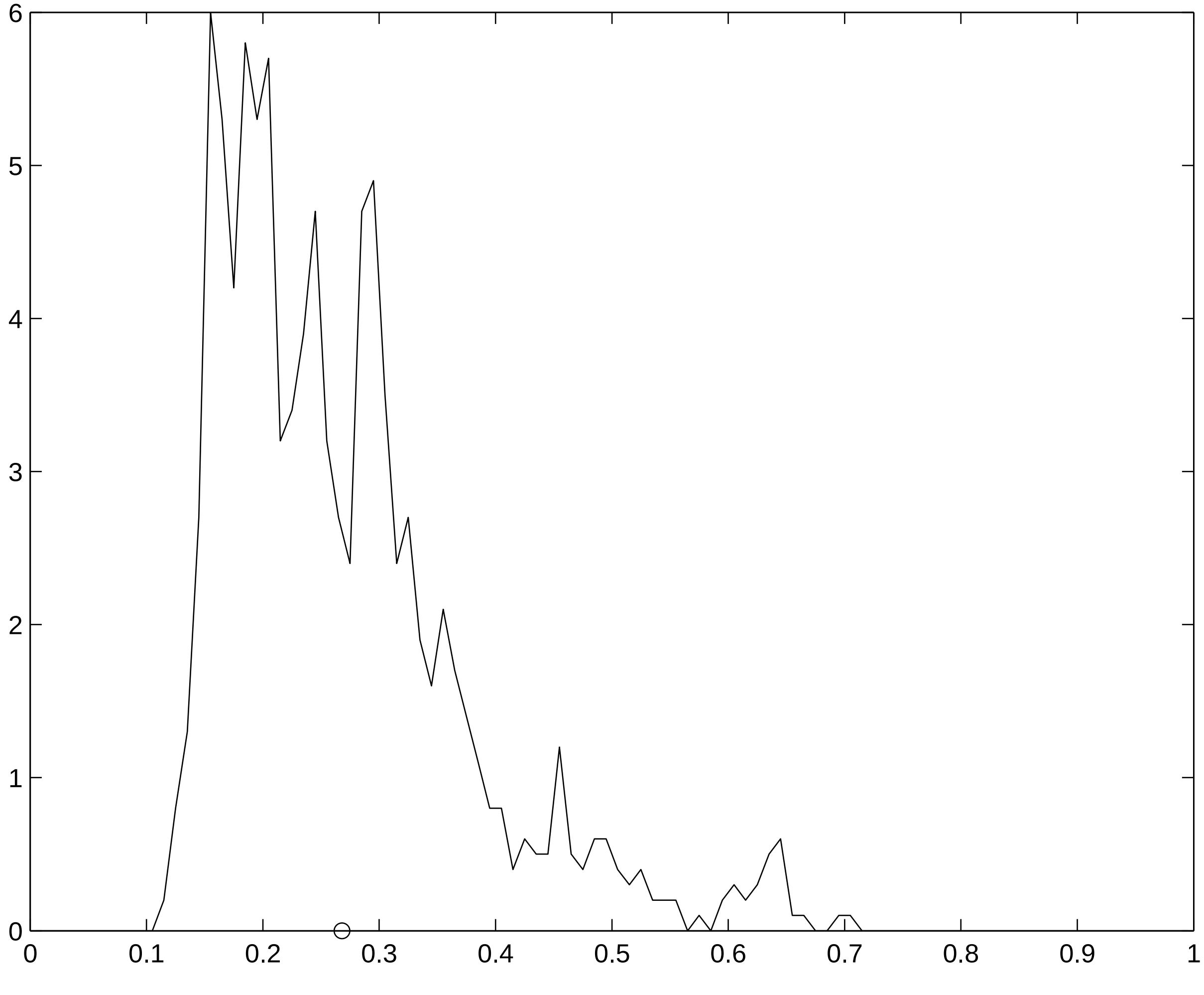
Error distribution: dataset size: 20



Error distribution: dataset size: 14



Error distribution: dataset size: 7



Bayes risk and consistency

- Traditional statistics has concentrated on analysing

$$\mathbb{E}_S\left[\epsilon(S, \mathcal{A}, \mathcal{F})\right]$$

- For example consistency of a classification algorithm $\mathcal{A}$ and function class $\mathcal{F}$ means

$$\lim_{m\to\infty} \mathbb{E}_S\left[\epsilon(S, \mathcal{A}, \mathcal{F})\right] = f_{\mathsf{Bayes}},$$

  where

$$f_{\mathsf{Bayes}}(\mathbf{x}) = \begin{cases} 1 & \text{if } P(\mathbf{x}, 1) > P(\mathbf{x}, 0), \\ 0 & \text{otherwise} \end{cases}$$

  is the function with the lowest possible risk, often referred to as the Bayes risk

Expected versus confident bounds

- For a finite sample the generalisation $\epsilon(S, \mathcal{A}, \mathcal{F})$ has a distribution depending on the algorithm, function class and sample size m

- Traditional statistics as indicated above has concentrated on the mean of this distribution – but this quantity can be misleading, eg for low fold cross-validation

Expected versus confident bounds cont.

- Statistical learning theory has preferred to analyse the tail of the distribution, finding a bound which holds with high probability

- This looks like a statistical test – significant at a 1% confidence means that the chances of the conclusion not being true are less than 1% over random samples of that size

- This is also the source of the acronym PAC: probably approximately correct, the ‘confidence’ parameter δ is the probability that we have been misled by the training set

Probability of being misled in classification

- Aim to cover a number of key techniques of SLT. Basic approach is usually to bound the probability of being misled and set this equal to δ

- What is the chance of being misled by a single bad function f, i.e. training error $\mathcal{E}_S(f) = 0$, while true error is bad $\mathcal{E}(f) > \epsilon$?

$$\begin{aligned} P_S\{\mathcal{E}_S(f) = 0\} &= (1 - \mathcal{E}(f))^m \\ &\leq (1 - \epsilon)^m \\ &\leq \exp(-\epsilon m) \end{aligned}$$

so that choosing $\epsilon \geq \ln(1/\delta)/m$ ensures probability less than δ

In other words, with probability at least $1 - \delta$

$$\mathcal{E}(f) \leq \frac{1}{m} \log(\frac{1}{\delta})$$

Finite function class

If we now consider a function class

$$\mathcal{F} = \{f_1, f_2, \ldots, f_N\}$$

then the probability of being misled by one of the functions while its true error is more than ϵ is bounded by

$$P_S\{\exists f_n \colon \mathcal{E}_S(f_n) = 0\} \leq \sum_{n=1}^{N} P_S\{\mathcal{E}_S(f_n) = 0\} \leq N \exp(-\epsilon m)$$

This uses the **union bound** – the probability of the union of a set of events is at most the sum of the individual probabilities

Finite function class (cont.)

If we make the probability of being misled by f_n less than $1/(N\delta)$, we have that

- The bound translates into a theorem: given $\mathcal{F}$, with probability at least $1 - \delta$ over random m samples the generalisation error of a function $f_n \in \mathcal{F}$ with zero training error is bounded by

$$\mathcal{E}(f_n) \leq \frac{1}{m}\left(\ln N + \ln\left(\frac{1}{\delta}\right)\right)$$

Countable function classes

If we now consider a function class

$$\mathcal{F} = \{f_n : n \in \mathbb{N}\}$$

and make the probability of being misled by f_n less than $q_n\delta$ with

$$\sum_{n=1}^{\infty} q_n = 1,$$

then with probability at least $1 - \delta$ over random m samples the generalisation error of a function $f_n \in \mathcal{F}$ with zero training error is bounded by

$$\mathcal{E}(f_n) \leq \frac{1}{m}\left(\ln\left(\frac{1}{q_n}\right) + \ln\left(\frac{1}{\delta}\right)\right)$$

Some comments on the result

- We can think of the term $\ln\left(\frac{1}{q_n}\right)$ as the complexity / description length of the function f_n

- Note that we must put a prior weight on the functions. If the functions are drawn at random according to a distribution p_n, the expected generalisation will be minimal if we choose our prior $q = p$

- This is the starting point of the PAC-Bayes analysis

Hoeffding inequality

Our next goal is to address the generalization problem with a general loss function

Let ξ be a random variable with mean $\mu = \mathbf{E}[\xi]$ and taking values in the interval $[a, b]$. Let $\xi_1, \ldots, \xi_m$ be an i.i.d. sample of ξ and define the empirical mean $\bar{\xi}(m) = \frac{1}{m} \sum_{i=1}^{m} \xi_i$

Then for every $\epsilon > 0$ we have that

$$\text{Prob}\left(\bar{\xi}(m) - \mu > \epsilon\right) \leq \exp\left(-\frac{2m\epsilon^2}{(b-a)^2}\right)$$

and

$$\text{Prob}\left(\mu - \bar{\xi}(m) > \epsilon\right) \leq \exp\left(-\frac{2m\epsilon^2}{(b-a)^2}\right)$$

Note: for a proof, see §8.2 Devroye, Györfi and Lugosi

Simplest case: $\mathcal{H} = \{f\}$

We shall apply Hoeffding's inequality to the random variable $\xi = V(y, f(\mathbf{x}))$
For simplicity, assume that $V(y, f(\mathbf{x})) \in [0, 1]$

Hoeffding's inequality gives

$$\text{Prob}\,(\mathcal{E}(f) - \mathcal{E}_S(f) > \epsilon) \leq \exp(-2m\epsilon^2)$$

This implies that with probability (confidence) at least $1 - \delta$
(think of δ as a small positive number)

$$\mathcal{E}(f) \leq \mathcal{E}_S(f) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$

We call $\mathcal{D}(f) := \mathcal{E}(f) - \mathcal{E}_S(f)$ the deviation error of f

Finite hypothesis space

Now suppose that $\mathcal{H} = \{f^{(1)}, \ldots, f^{(N)}\}$

For each fixed function in $\mathcal{H}$, Hoeffding's inequality holds true

However, now we are not interested in the deviation error of a fixed function but in the deviation error of the **minimizer** of $\mathcal{E}_S$

$$f_S \equiv f^{(n_*)} := \mathrm{argmin}_{n=1}^{N} \sum_{i=1}^{m} V(y_i, f^{(n)}(\mathbf{x}_i))$$

What can we say about the probability of the following event?

$$\{\mathcal{D}(f^{(n_*)}) > \epsilon\}$$

Since $f^{(n_*)}$ can be any function in $\mathcal{H}$ we need a **uniform bound** over $\mathcal{H}$!

Union bound

Recall the following fundamental property of probability: for every set $A_1, A_2, \ldots, A_n$ of events, we have that

$$P(A_1 \cup A_2 \cup \ldots \cup A_N) \leq \sum_{n=1}^{N} P(A_n)$$

So, if we let $A_n = \{|\mathcal{E}(f^{(n)}) - \mathcal{E}_S(f^{(n)})| \geq \epsilon\}$ we conclude that

$$A_1 \cup A_2 \cup \ldots \cup A_N = \left\{ \max_{n=1}^{N} |\mathcal{E}(f^{(n)}) - \mathcal{E}_S(f^{(n)})| \geq \epsilon \right\}$$

Thus, we have

$$\text{Prob}\left\{ \max_{n=1}^{N} \left\{ \mathcal{E}(f^{(n)}) - \mathcal{E}_S(f^{(n)}) \right\} \geq \epsilon \right\} \leq N \exp(-2m\epsilon^2)$$

Uniform bound

$$\text{Prob}\left\{\max_{n=1}^{N}\left\{\mathcal{E}(f^{(n)}) - \mathcal{E}_S(f^{(n)})\right\} \geq \epsilon\right\} \leq N\exp(-2m\epsilon^2)$$

This is also called a **uniform** bound over $\mathcal{H}$ (it holds for every $f \in \mathcal{H}$). The bound implies that

$$\text{Prob}\left\{\mathcal{E}(f_S) - \mathcal{E}_S(f_S) \geq \epsilon\right\} \leq N\exp(-2m\epsilon^2)$$

which is equivalent to say that with confidence at least $1 - \delta$

$$\mathcal{E}(f_S) \leq \mathcal{E}_S(f_S) + \sqrt{\frac{\log N + \log\frac{1}{\delta}}{2m}}$$

As N increases more examples are required in order to avoid overfitting!

Sample complexity bound

$$\text{Prob}\left\{\mathcal{E}(f_S) - \mathcal{E}_S(f_S) \geq \epsilon\right\} \leq |\mathcal{H}| \exp(-2m\epsilon^2)$$

How many examples are needed to avoid overfitting?

Sample complexity: minimum number m of examples that we need in order to ensure that the deviation error of f_S will be less than ϵ with probability at least $1 - \delta$

The sample complexity depends on $\mathcal{H}, \epsilon$ and δ. In our case:

$$m(\epsilon, \mathcal{H}, \delta) = \frac{\log|\mathcal{H}| + \log\frac{1}{\delta}}{2\epsilon^2}$$

Structural risk minimization

This is a model selection approach to choosing a hypothesis space within a nested family of spaces: $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \subset \mathcal{H}_Q$

(For simplicity assume each set $\mathcal{H}_q$ is finite)

Let $f_{S,q}$ be the minimizer of $\mathcal{E}_S$ in $\mathcal{H}_q$. For each fixed q we have with confidence at least $1 - \delta$

$$\mathcal{E}(f_{S,q}) \leq \mathcal{E}_S(f_{S,q}) + \sqrt{\frac{\log|\mathcal{H}_q| + \log\frac{1}{\delta}}{2m}}$$

The **structural risk minimization** chooses the hypothesis space $\mathcal{H}_{q^*}$ which minimizes the r.h.s. of this inequality

Structural risk minimization (cont.)

For each fixed q we have

$$\mathcal{E}(f_{S,q}) \leq \mathcal{E}_S(f_{S,q}) + \sqrt{\frac{\log|\mathcal{H}_q| + \log\frac{1}{\delta}}{2m}}$$

The **structural risk minimization** chooses the hypothesis space $\mathcal{H}_{q^*}$ which minimizes the r.h.s. of this inequality

The expected error of function f_{S,q^*} (model) is bounded as

$$\mathcal{E}(f_{S,q^*}) \leq \mathcal{E}_S(f_{S,q^*}) + \sqrt{\frac{\log|\mathcal{H}_{q^*}| + \log Q + \log\frac{1}{\delta}}{2m}}$$

VC–dimension

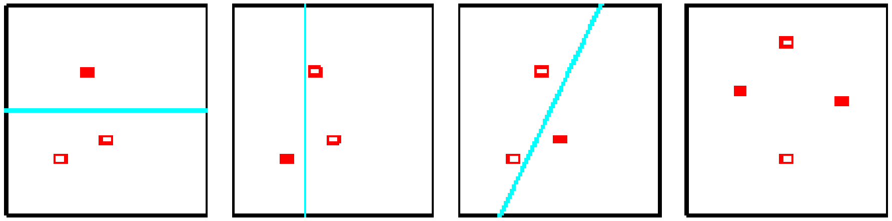
If $\mathcal{H}$ is an infinite set (for example the set of all linear classifiers) things become a bit more complicated but bounds similar to the above one can still be derived which contain a **complexity measure** of $\mathcal{H}$

For binary classification, the most widely used complexity measure is the **VC–dimension** (from Vapnik and Chervonenkis)

- The VC–dimension of a set of binary classifiers $\mathcal{H}$ is the largest number h of inputs $\mathbf{x}_1, \ldots, \mathbf{x}_h$ which can be shattered (classified) in all 2^h possible ways using classifiers in $\mathcal{H}$

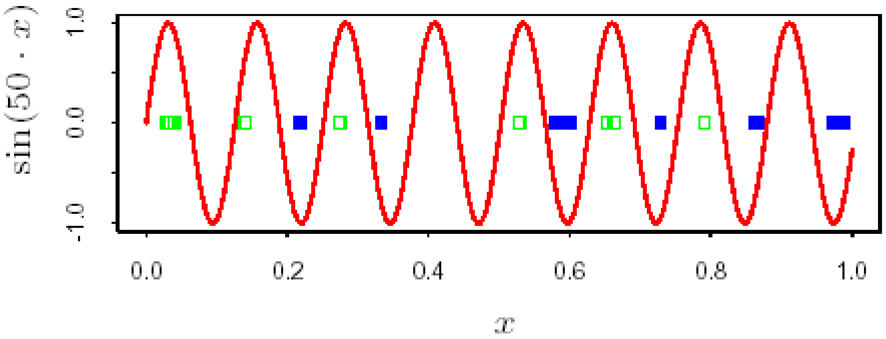
VC–dimension

The VC–dimension of lines in the plane is 3 (in $\mathbb{R}^d$ is $d+1$)



The VC–dimension of hyper-rectangles in $\mathbb{R}^d$ is $2d$ (exercise)

Example in which VC-dim is infinite $\mathcal{H} = \{\text{sign}(\sin ax) : a \in \mathbb{R}\}$



VC–bounds

Theorem: (V. & C.) with confidence at least $1 - \delta$ we have that

$$\mathcal{E}(f_S) \leq \mathcal{E}_S(f_S) + 2\sqrt{2\frac{h(\log\frac{2m}{h} + 1) + \log\frac{2}{\delta}}{m}}$$

Compare to the case that $\mathcal{H}$ is finite, where $h \leq \log N$ ($N := |\mathcal{H}|$)

$$\mathcal{E}(f_S) \leq \mathcal{E}_S(f_S) + \sqrt{\frac{\log N + \log\frac{1}{\delta}}{2m}}$$

- It may be that $h \ll \log N \Rightarrow VC$–bound is better
- When $h = \log N$, VC–bound is slightly worse but essentially the same: $O\left(\frac{h}{m}\log\frac{m}{h}\right)$ vs. $O\left(\frac{h}{m}\right)$

Bias/variance decomposition

Recall that $f^* = \text{argmin}_f \mathcal{E}(f)$

$$\mathcal{E}(f_S) - \mathcal{E}(f^*) = \underbrace{\mathcal{E}(f_S) - \mathcal{E}(f_{\mathcal{H}})}_{\text{variance}} + \underbrace{\mathcal{E}(f_{\mathcal{H}}) - \mathcal{E}(f^*)}_{\text{bias}}$$

where $f_{\mathcal{H}} = \text{argmin}_{f \in \mathcal{H}} \mathcal{E}(f)$. Using the VC–bound and the fact that $\mathcal{E}_S(f_S) \leq \mathcal{E}_S(f_{\mathcal{H}})$ we obtain that

$$\begin{aligned}
\mathcal{E}(f_S) - \mathcal{E}(f_{\mathcal{H}}) &= \mathcal{E}(f_S) - \mathcal{E}_S(f_S) + \mathcal{E}_S(f_S) - \mathcal{E}(f_{\mathcal{H}}) \\
&\leq \mathcal{E}(f_S) - \mathcal{E}_S(f_S) + \mathcal{E}_S(f_H) - \mathcal{E}(f_{\mathcal{H}}) \\
&\leq 4\sqrt{2\frac{h(\log\frac{2m}{h} + 1) + \log\frac{2}{\delta}}{m}}
\end{aligned}$$

Sample error and approximation error

$$\mathcal{E}(f_S) - \mathcal{E}(f^*) = \underbrace{\mathcal{E}(f_S) - \mathcal{E}(f_{\mathcal{H}})}_{\text{variance}} + \underbrace{\mathcal{E}(f_{\mathcal{H}}) - \mathcal{E}(f^*)}_{\text{bias}}$$

- the variance (or **sample error**) increases with the complexity of $\mathcal{H}$ and decreases with the sample size m

- the bias is independent of the sample and decreases with the complexity of $\mathcal{H}$. It measures the **approximation error** of $\mathcal{H}$ to f^*. For example, for the square loss we have:

$$\mathcal{E}(f) = E_{\mathbf{x},y}[(y - f(\mathbf{x}))^2] = \mathcal{E}(f^*) + \mathbf{E}_{\mathbf{x}}[(f(\mathbf{x}) - f^*(\mathbf{x}))^2]$$

Bias/variance decomposition

Another way to study the behavior of a learning algorithm is via the average generalization error (over the sampling of the training set S)

We define $\bar{f}(\mathbf{x}) = \mathbf{E}_S[f_S(\mathbf{x})]$. For the square loss we have that

$$\mathbf{E}_S\left[\mathcal{E}(f)\right] = \mathcal{E}(f^*) + \underbrace{\mathbf{E}_S[\|\bar{f} - f_S\|^2]}_{\text{variance}} + \underbrace{\|f^* - \bar{f}\|^2}_{\text{bias}}$$

where we have used the notation

$$\|f - g\|^2 := \mathbf{E}_{\mathbf{x}}\left[(f(\mathbf{x}) - g(\mathbf{x}))^2\right]$$

When $\bar{f} = f_{\mathcal{H}}$ the bias is the same as the previous notion of bias (approximation error)