GI01/M055: Supervised Learning

5. Learning Theory

November 16, 2009

John Shawe-Taylor

Today’s plan

- Model selection problem
- Generalisation error bounds
- Error bound analysis
- VC–dimension
- Bias/variance trade-off
- Support Vector Machines
- Kernel Principal Components Analysis

Bibliography: These lecture notes are available at:
http://www.cs.ucl.ac.uk/staff/J.Shawe-Taylor/courses/index-gi01.html

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

- Our goal is to minimize $\mathcal{E}$
- Optimal solution: $f^* := \text{argmin}_f \mathcal{E}(f)$
- 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: $V(y,z) = y \log(1+e^{-z}) + (1-y)\log(1+e^{z})$

We'll see further loss functions later in the course when we speak about support vector machines

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**

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

Supervised learning model (IV)

If we minimize $\mathcal{E}_{\text{emp}}$ 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 $\mathcal{E}_{\text{emp}}$ within $\mathcal{H}$. That is, our learning algorithm is:

$$f_S = \text{argmin}_{f \in \mathcal{H}} \mathcal{E}_{\text{emp}}(f)$$

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

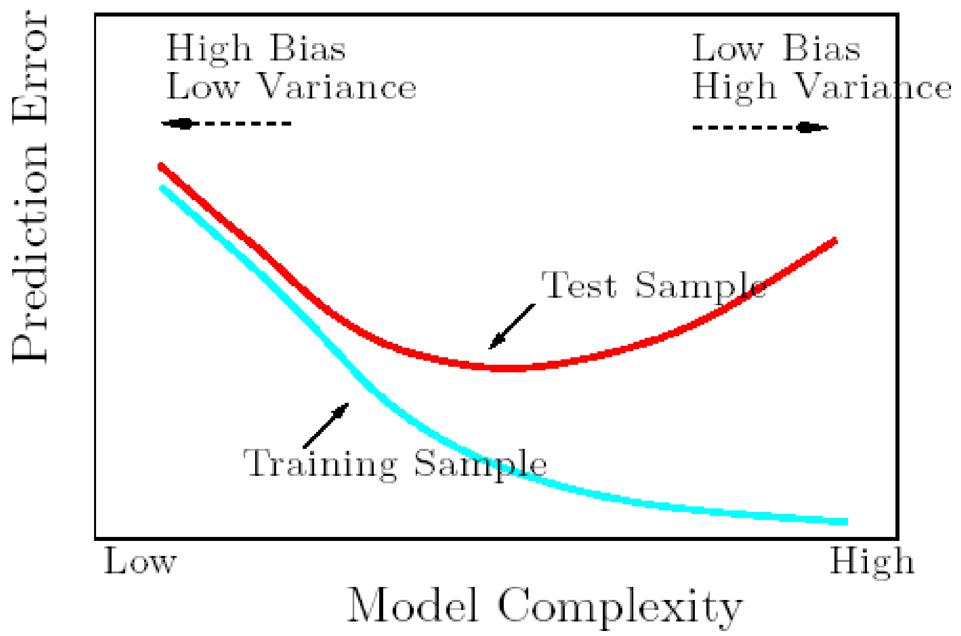
$$E_\lambda(f) := \frac{1}{m}\sum_{i=1}^{m} V(y_i, f(\mathbf{x}_i)) + \lambda R(f), \qquad \lambda > 0$$

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 or prediction error) 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)

Test set bound

- What does it mean if we observe an error rate of $\hat{c}_S$ on a sample S of size m?

- Can we say something about the true error rate c_D?

- If we assume sample is drawn independently and identically (i.i.d.) from the distribution that generates the test data, we have the following result:

Proposition 1 *Fix $\delta > 0$. With probability at least $1 - \delta$ over the generation of the sample S,*

$$c_D \leq \hat{c}_S + \sqrt{\frac{\ln(1/\delta)}{2m}}.$$

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**

Theories of learning

- Basic approach of SLT is to view learning from a statistical viewpoint.

- Aim of any theory is to model real/ artificial phenomena so that we can better understand/ predict/ exploit them.

- SLT is just one approach to understanding/ predicting/ exploiting learning systems, others include Bayesian inference, inductive inference, statistical physics, traditional statistical analysis.

Theories of learning cont.

- Each theory makes assumptions about the phenomenon of learning and based on these derives predictions of behaviour as well as algorithms that aim at optimising the predictions.

- Each theory has strengths and weaknesses – the better it captures the details of real world experience, the better the theory and the better the chances of it making accurate predictions and driving good algorithms.

General statistical considerations

- Statistical models (not including Bayesian) begin with an assumption that the data is generated by an underlying distribution P typically not given explicitly to the learner.

- If we are trying to classify cancerous tissue from healthy tissue, there are two distributions, one for cancerous cells and one for healthy ones.

General statistical considerations cont.

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

- Rather than accessing the distribution directly, statistical learning typically 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 $\mathcal{A}_{\mathfrak{F}}(S)$ from a function space $\mathfrak{F}$ in response to the training set S.

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

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

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

Generalisation of a learner

- For example, in the case of classification ℓ 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 the simple Parzen window classifier described by Bernhard Schölkopf: 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

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

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

- We plot the histogram and the average of the distribution 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:

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

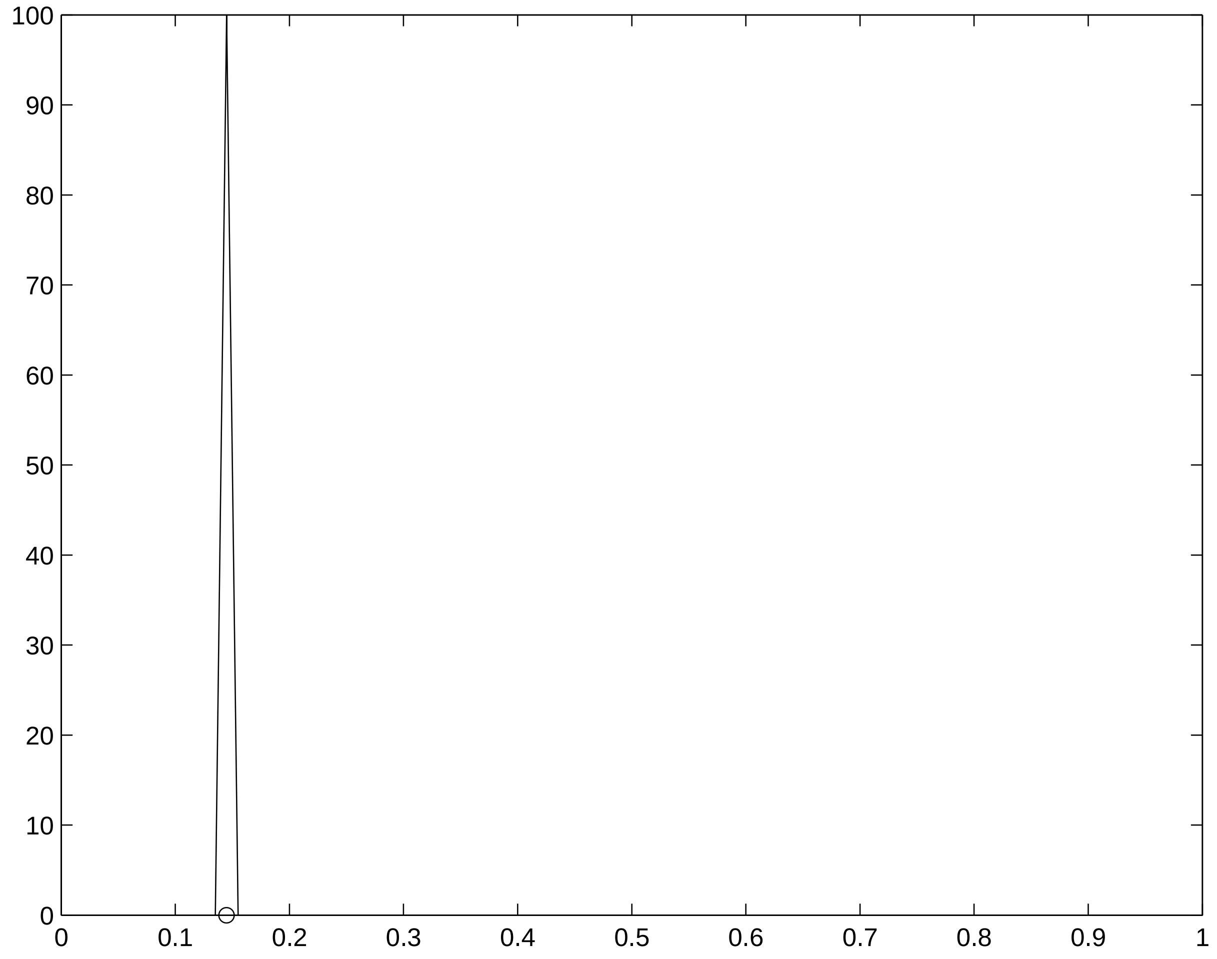
Example of Generalisation III

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

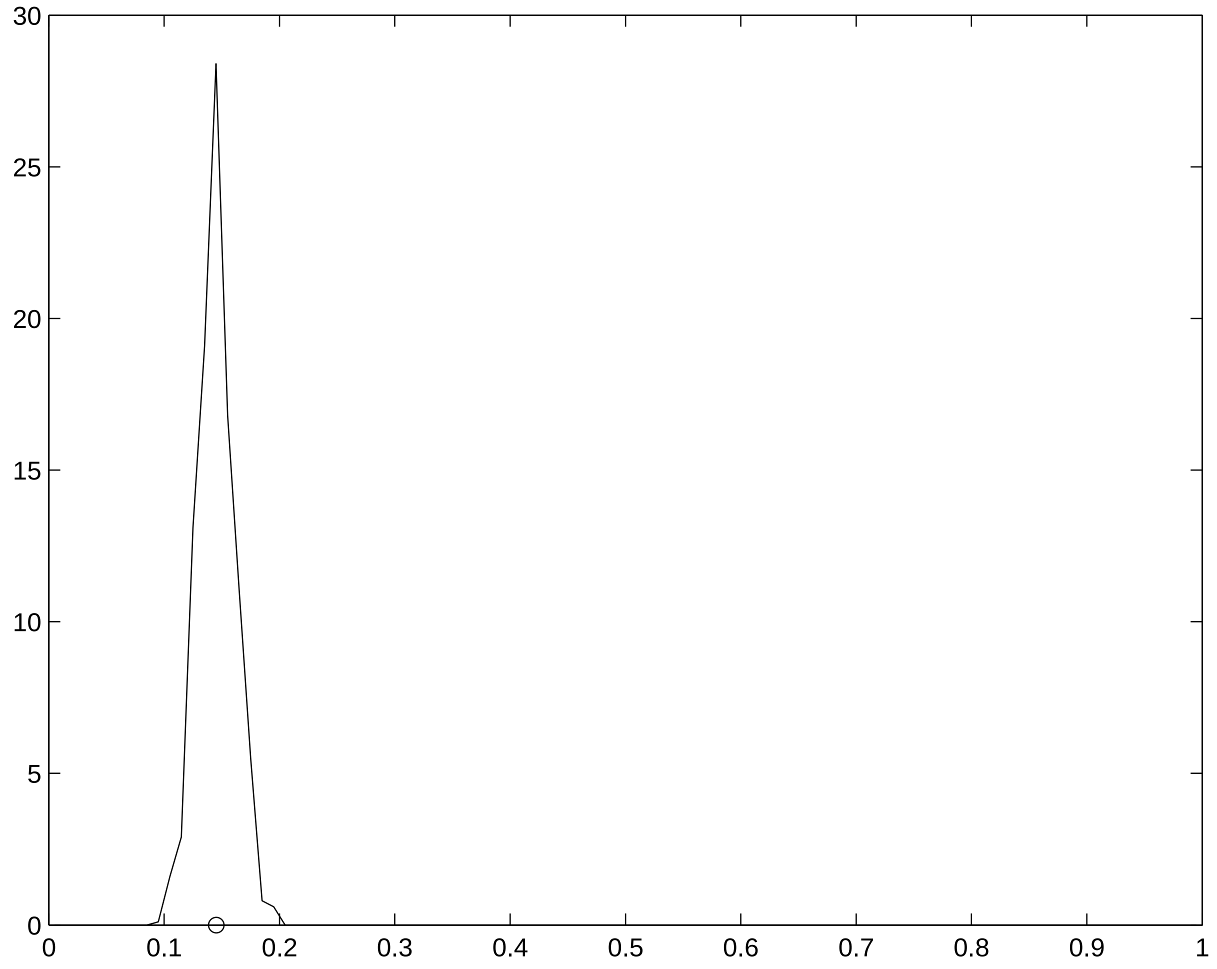
$$\begin{aligned}\mathbb{E}\left[\mathcal{A}_{\mathcal{F}}(S)\right] &= \mathbb{E}_S\left[\mathbf{w}_S^+ - \mathbf{w}_S^-\right]\\ &= \mathbb{E}_S\left[\mathbf{w}_S^+\right] - \mathbb{E}_S\left[\mathbf{w}_S^-\right]\\ &= \mathbb{E}_{y=+1}\left[\mathbf{x}\right] - \mathbb{E}_{y=-1}\left[\mathbf{x}\right],\end{aligned}$$

  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.

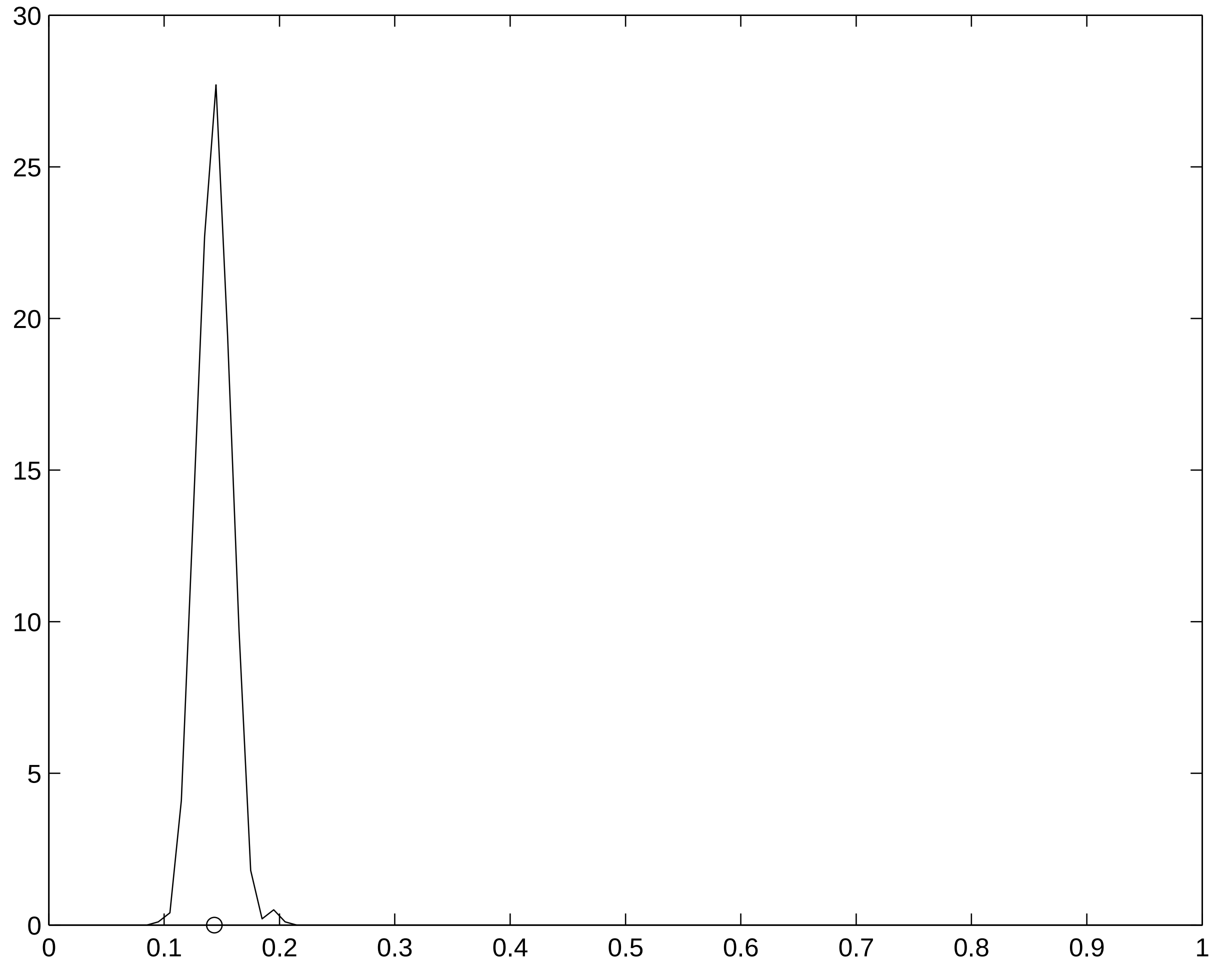
Error distribution: full dataset



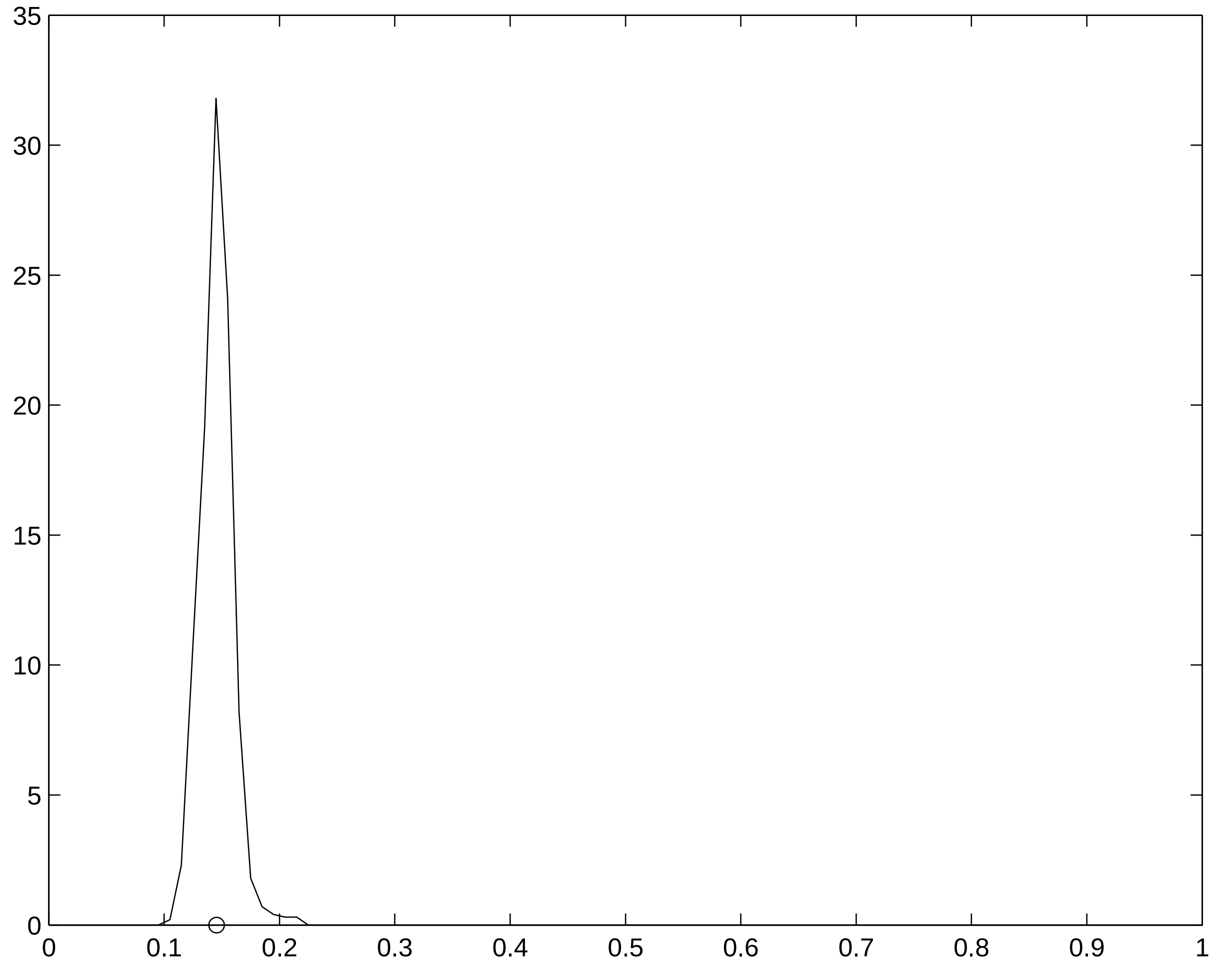
Error distribution: dataset size: 342



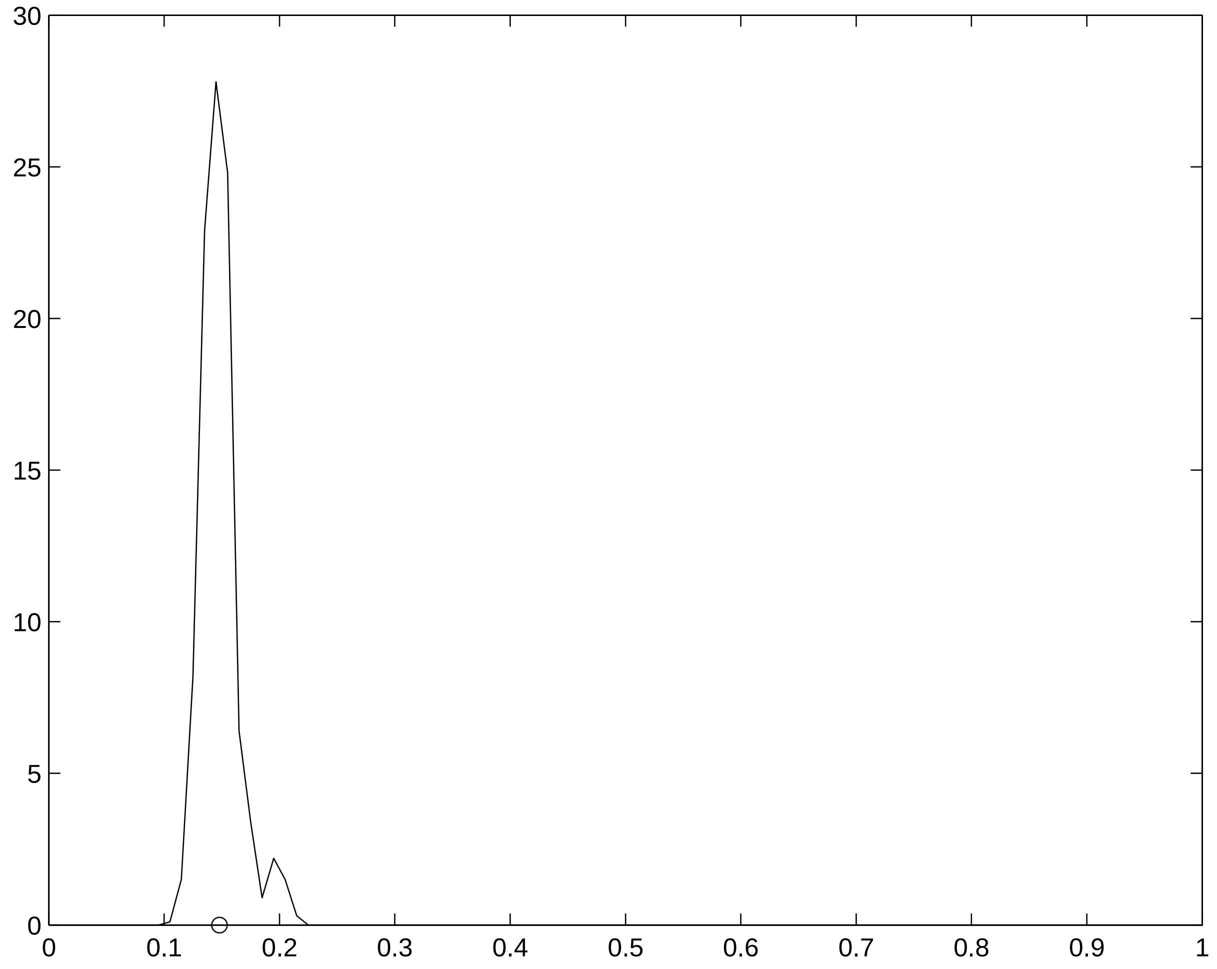
Error distribution: dataset size: 273



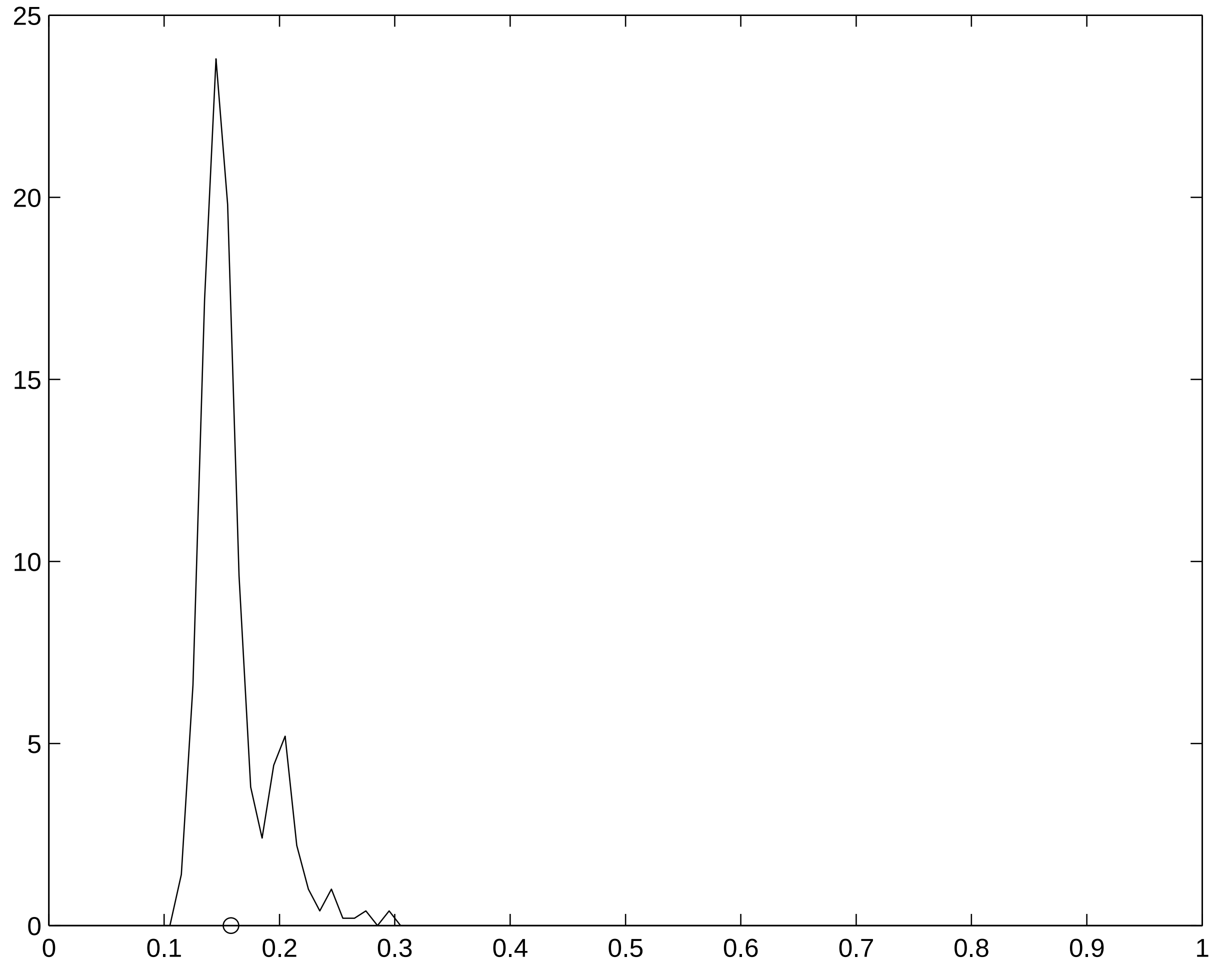
Error distribution: dataset size: 205



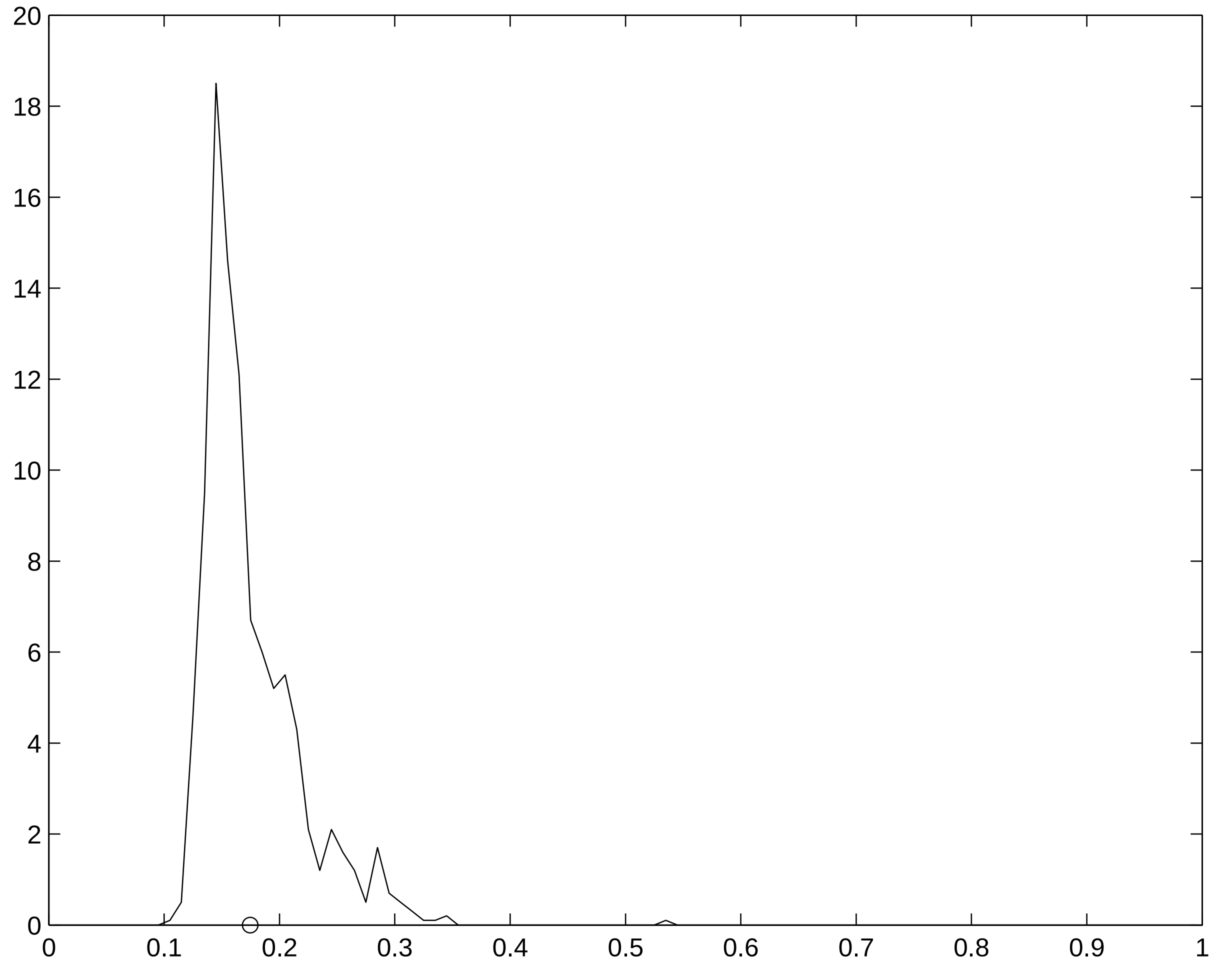
Error distribution: dataset size: 137



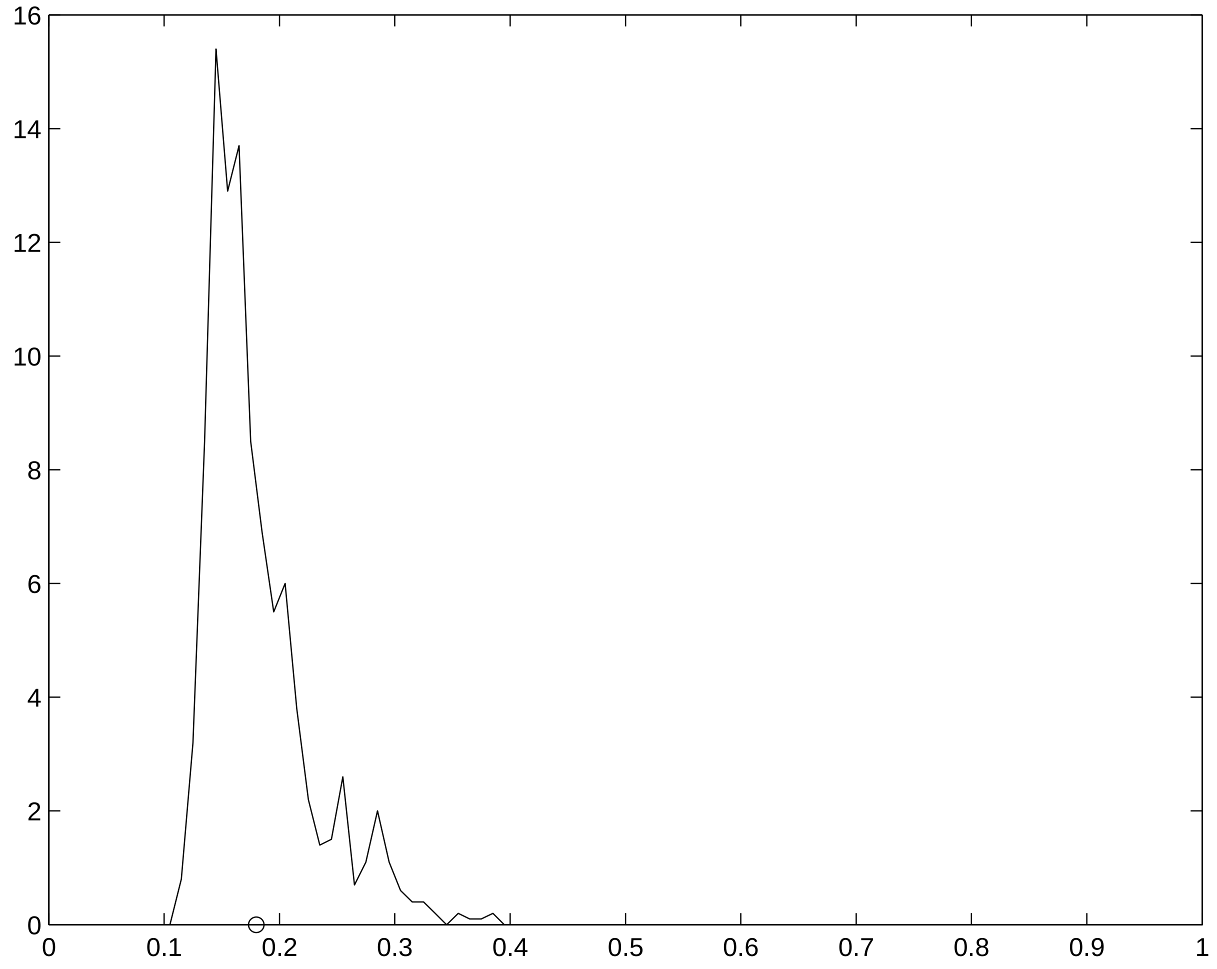
Error distribution: dataset size: 68



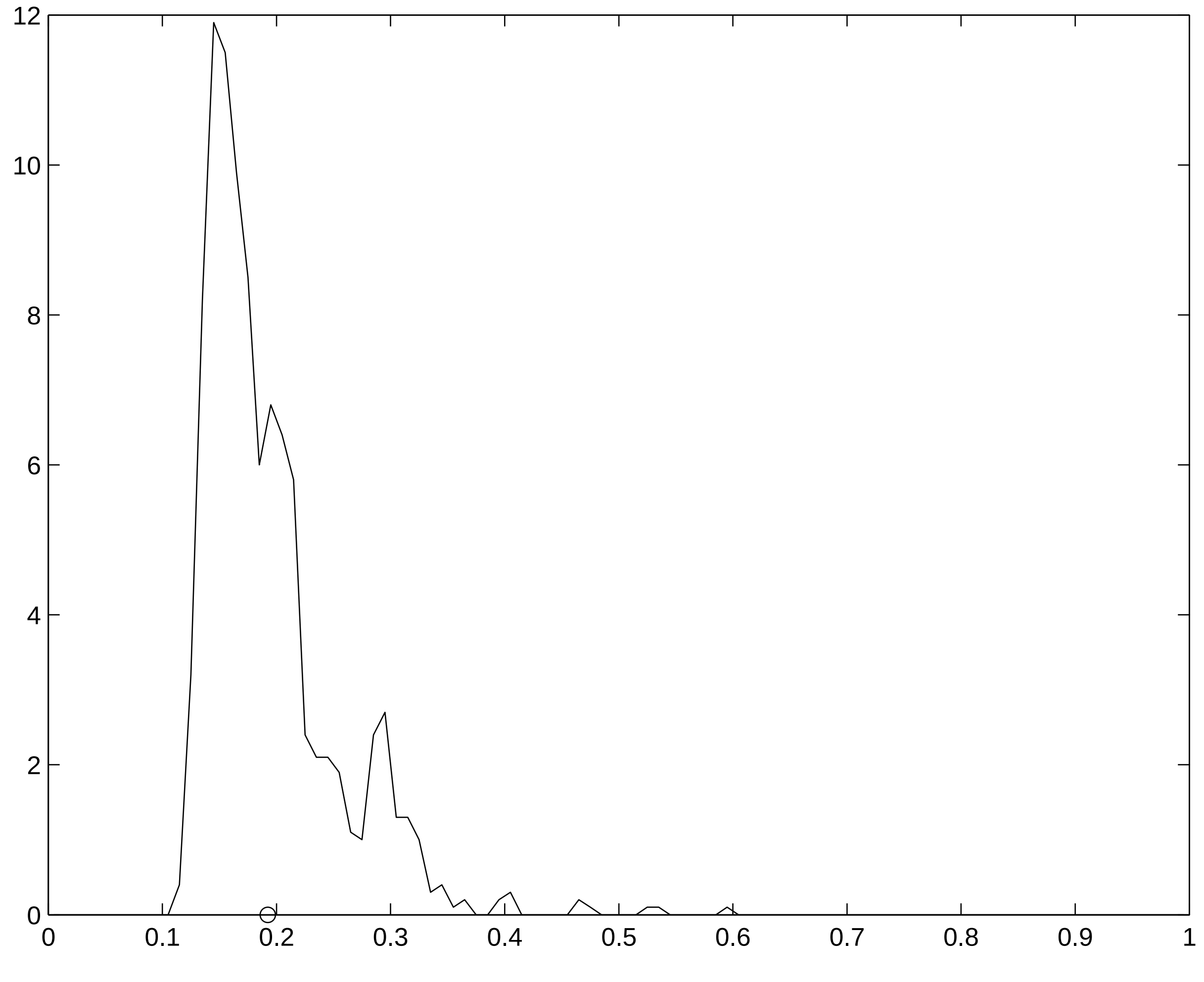
Error distribution: dataset size: 34



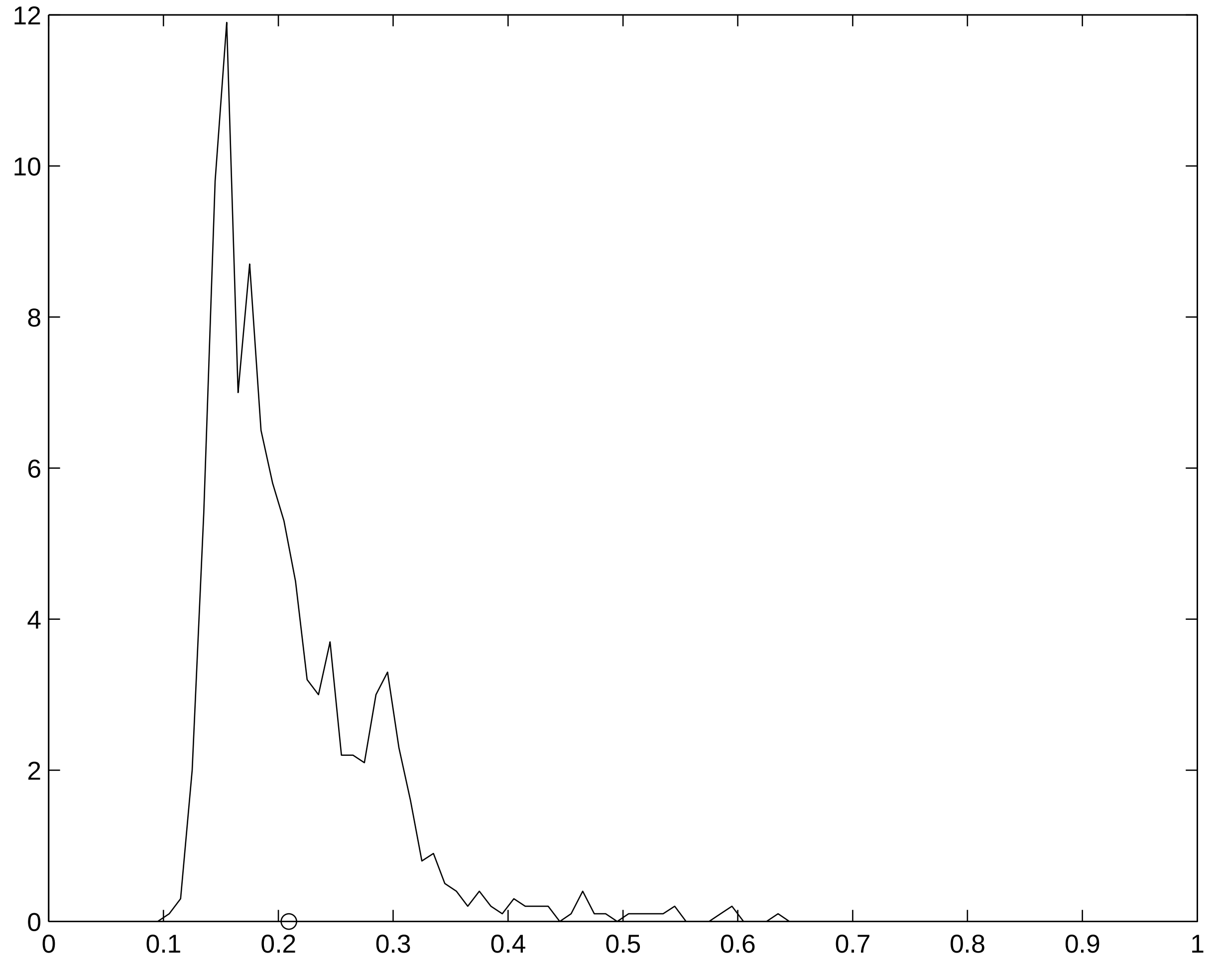
Error distribution: dataset size: 27



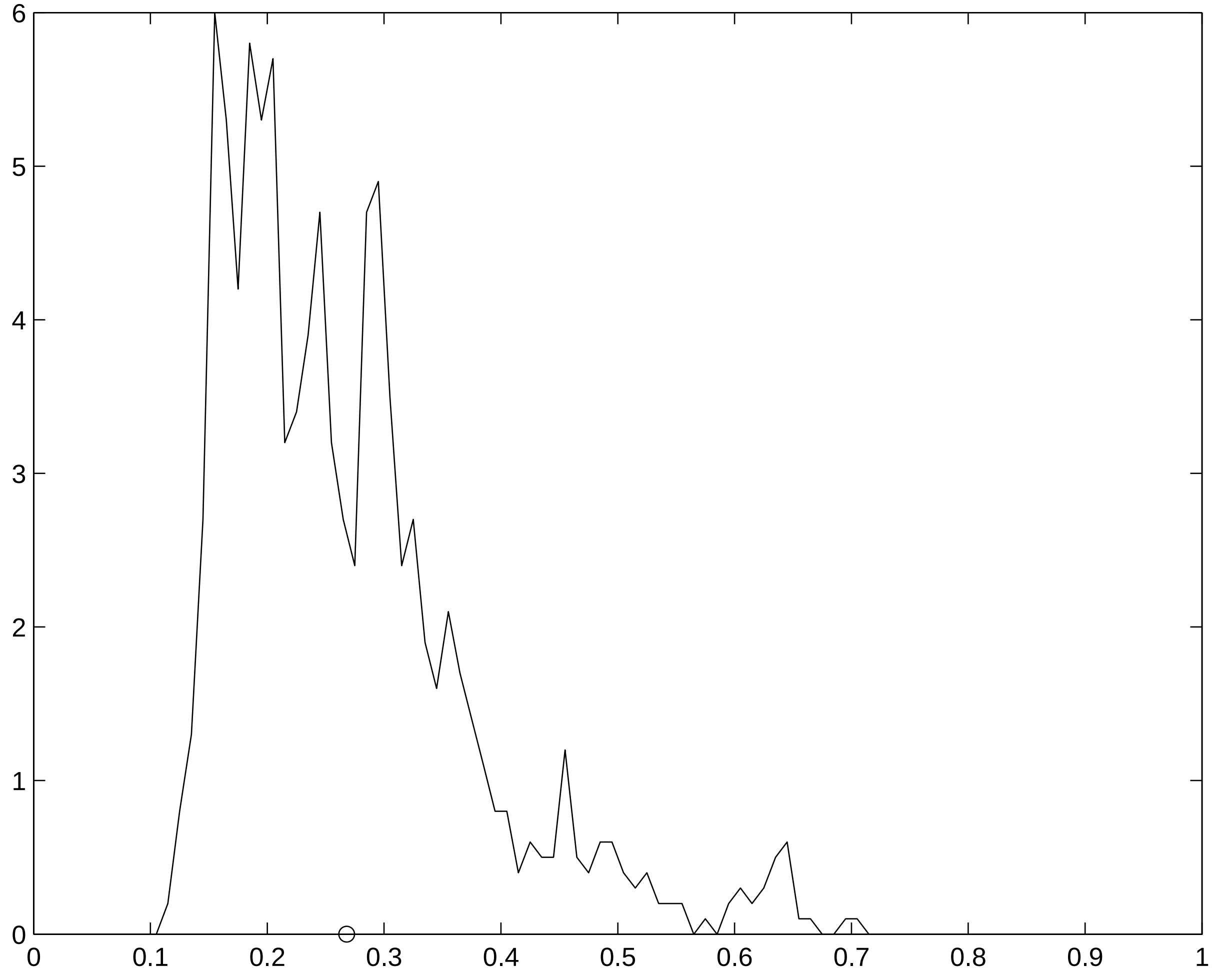
Error distribution: dataset size: 20



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

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

where

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

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 $\text{err}_S(f) = 0$, while true error is bad $\text{err}(f) > \epsilon$?

$$\begin{aligned} P_S\{\text{err}_S(f) = 0, \text{err}(f) > \epsilon\} &= (1 - \text{err}(f))^m \\ &\leq (1 - \epsilon)^m \\ &\leq \exp(-\epsilon m). \end{aligned}$$

so that choosing $\epsilon = \ln(1/t)/m$ ensures probability less than t.

Finite or Countable function classes

If we now consider a function class

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

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

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

then the probability of being misled by one of the functions is bounded by

$$P_S\left\{\exists f_n \colon \mathrm{err}_S(f_n) = 0, \mathrm{err}(f_n) > \frac{1}{m}\ln\left(\frac{1}{q_n\delta}\right)\right\} \leq \delta.$$

This uses the so-called union bound – the probability of the union of a set of events is at most the sum of the individual probabilities.

Finite or Countable function classes result

- The bound translates into a theorem: given $\mathcal{F}$ and q, 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

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

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

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 $V(y, f(\mathbf{x})) \in [0, 1]$ (one can always reduce to this case anyway)

Hoeffding's inequality gives

$$\text{Prob}\left(\mathcal{E}(f) - \mathcal{E}_{\text{emp}}(f) > \epsilon\right) \leq \exp(-2m\epsilon^2)$$

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

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

We call $\mathcal{D}(f) := \mathcal{E}(f) - \mathcal{E}_{\text{emp}}(f)$ 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 $\mathcal{E}_{\text{emp}}$

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

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 $A_n = \{|\mathcal{E}(f^{(n)}) - \mathcal{E}_{\text{emp}}(f^{(n)})| \geq \epsilon\}$ we conclude that

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

Thus, we have

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

Uniform bound

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

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

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

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

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

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

Sample complexity bound

$$\text{Prob}\,\{\mathcal{E}(f_S) - \mathcal{E}_{\text{emp}}(f_S) \geq \epsilon\} \leq |\mathcal{H}| \exp(-2m\epsilon^2)$$

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 $\mathcal{E}_{\text{emp}}$ in $\mathcal{H}_q$. For each fixed q we have with confidence at least $1 - \delta$

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

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

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

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

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

What if uncountably many functions?

- We need a way to convert from an infinite set to a finite one.
- Key idea is to replace measuring performance on a random test point with measuring on a second ‘ghost’ sample
- In this way the analysis is reduced to a finite set of examples and hence a finite set of classification functions.
- This step is often referred to as the ‘double sample trick’

Double sample trick

The result has the following form:

$$\begin{aligned} P^m\{\mathbf{X} \in X^m : \exists h \in H : \mathrm{err}_{\mathbf{X}}(h) = 0, \mathrm{err}(h) \geq \epsilon\} \\ \leq 2P^{2m}\{\mathbf{XY} \in X^{2m} : \exists h \in H : \\ \mathrm{err}_{\mathbf{X}}(h) = 0, \mathrm{err}_{\mathbf{Y}}(h) \geq \epsilon/2\} \end{aligned}$$

If we think of the first probability as being over $\mathbf{XY}$ the result concerns three events:

$$\begin{aligned} A(h) &:= \{\mathrm{err}_{\mathbf{X}}(h) = 0\} \\ B(h) &:= \{\mathrm{err}(h) \geq \epsilon\} \\ C(h) &:= \{\mathrm{err}_{\mathbf{Y}}(h) \geq \epsilon/2\} \end{aligned}$$

Double sample trick II

It is clear that

$$\begin{aligned} P^{2m}(C(h)|A(h)\&B(h)) &= P^{2m}(C(h)|B(h)) \\ &> 0.5 \end{aligned}$$

for reasonable m by a binomial tail bound.

Double sample trick II

Hence, we have

$$
\begin{aligned}
P^{2m}\{\mathbf{XY} \in X^{2m} &: \exists h \in H : A(h) \& C(h)\} \geq \\
P^{2m}\{\mathbf{XY} \in X^{2m} &: \exists h \in H : A(h) \& B(h) \& C(h)\} = \\
P^{2m}\{\mathbf{XY} \in X^{2m} &: \exists h \in H : A(h) \& B(h)\} \\
& P(C(h) | A(h) \& B(h))
\end{aligned}
$$

It follows that

$$
\begin{aligned}
P^{m}\{\mathbf{X} \in X^{m} &: \exists h \in H : A(h) \& B(h)\} \leq \\
& 2P^{2m}\{\mathbf{XY} \in X^{2m} : \exists h \in H : A(h) \& C(h)\}
\end{aligned}
$$

the required result.

How many functions on a finite sample?

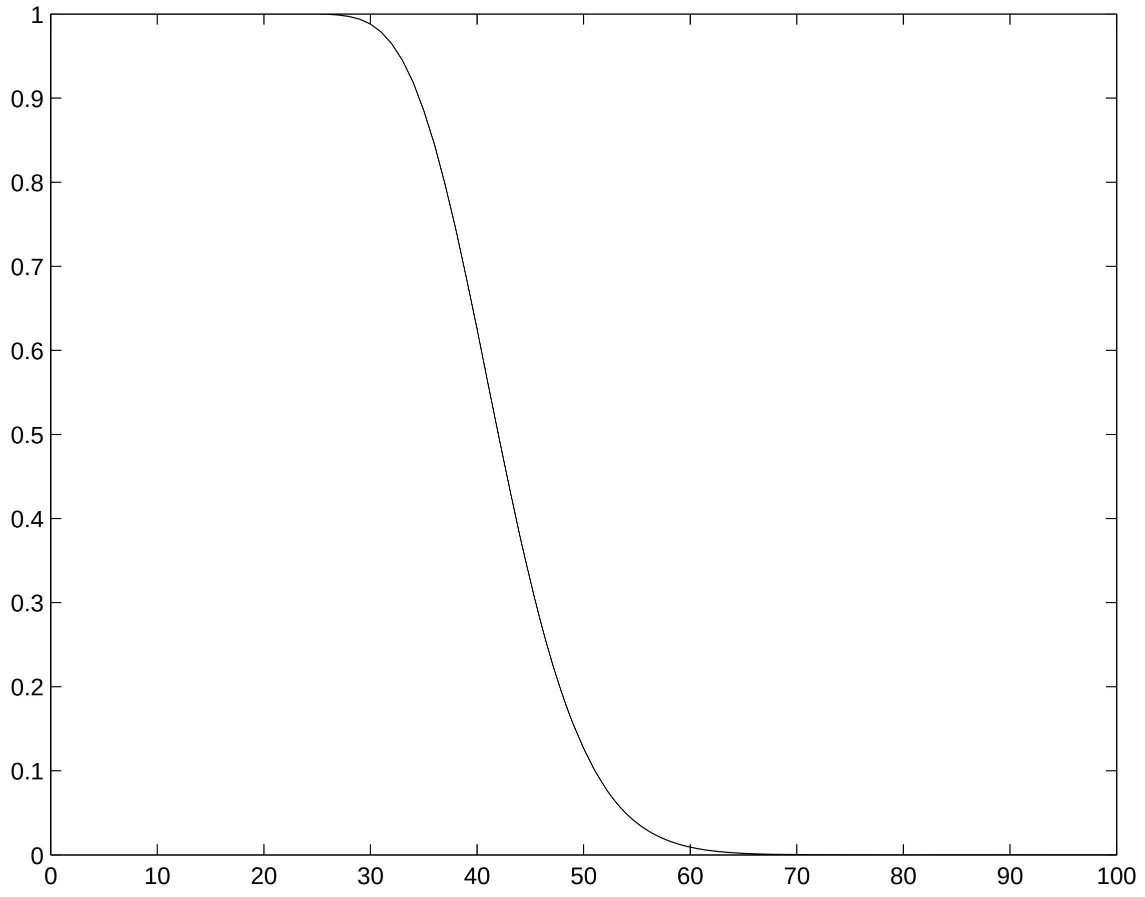
- Let H be a set of $\{-1, 1\}$ valued functions.

- The growth function $B_H(m)$ is the maximum cardinality of the set of functions H when restricted to m points – note that this cannot be larger than 2^m, i.e. $\log_2(B_H(m)) \leq m$

- For the statistics to work we want the number of functions to be much smaller than this.

Examining the growth function

Consider a plot of the ratio of the growth function $B_H(m)$ to 2^m for linear



Vapnik Chervonenkis dimension

- The Vapnik-Chervonenkis dimension is the point at which the graph stops being linear:

$$\begin{aligned}\text{VCdim}(H) = \max\{m \quad : \quad & \text{for some } \mathbf{x}_1, \ldots, \mathbf{x}_m, \\ & \text{for all } b \in \{-1, 1\}^m, \\ & \exists h_b \in H, h_b(\mathbf{x}_i) = b_i\}\end{aligned}$$

- For linear functions $\mathfrak{L}$ in $\mathbb{R}^n$, $\text{VCdim}(\mathfrak{L}) = n + 1$.

Sauer's Lemma

- Sauer's Lemma:

$$B_H(m) \leq \sum_{i=0}^{d} \binom{m}{i} \leq \left(\frac{em}{d}\right)^d,$$

where $m \geq d = \text{VCdim}(H)$.

Basic Theorem of SLT

We want to bound the probability that the training examples can mislead us about one of the functions we are considering using:

$$P^m\{\mathbf{X} \in X^m : \exists h \in H : \mathrm{err}_{\mathbf{X}}(h) = 0, \mathrm{err}(h) \geq \epsilon\}$$

$\longrightarrow$ double sample trick $\longrightarrow$

$$\leq 2P^{2m}\{\mathbf{XY} \in X^{2m} : \exists h \in H : \mathrm{err}_{\mathbf{X}}(h) = 0, \mathrm{err}_{\mathbf{Y}}(h) \geq \epsilon/2\}$$

$\longrightarrow$ union bound $\longrightarrow$

$$\leq 2B_H(2m)P^{2m}\{\mathbf{XY} \in X^{2m} : \mathrm{err}_{\mathbf{X}}(h) = 0, \mathrm{err}_{\mathbf{Y}}(h) \geq \epsilon/2\}$$

Final ingredient is known as symmetrisation.

Symmetrisation

- Consider generating a $2m$ sample S. Since the points are generated independently the probability of generating the same set of points in a different order is the same.

- Consider a fixed set Σ of permutations and each time we generate a sample we randomly permute it with a uniformly chosen element of Σ – gives probability distribution P_{Σ}^{2m}

Symmetrisation cont.

- Any event has equal probability under P^{2m} and P_{Σ}^{2m}, so that

$$P^{2m}(A) = P_{\Sigma}^{2m}(A) = \mathbb{E}^{2m}\left[P_{\sigma \sim \Sigma}(A)\right]$$

- Consider particular choice of $\boldsymbol{\Sigma}$ the permutations that swap/leave unchanged corresponding elements of the two samples $\mathbf{X}$ and $\mathbf{Y}$ – 2^m such permutations.

Completion of the proof

$$\begin{aligned}
P^{2m}\{\mathbf{XY} \in X^{2m}: \text{err}_{\mathbf{X}}(h) = 0, \text{err}_{\mathbf{Y}}(h) \geq \epsilon/2\} \\
\leq \mathbb{E}^{2m}\left[P_{\sigma\sim\Sigma}\{\text{err}_{\mathbf{X}}(h) = 0, \text{err}_{\mathbf{Y}}(h) \geq \epsilon/2 \text{ for } \sigma(\mathbf{XY})\}\right] \\
\leq \mathbb{E}^{2m}\left[2^{-\epsilon m/2}\right] \\
= 2^{-\epsilon m/2}
\end{aligned}$$

- Setting the right hand side equal to $\delta/(2B_H(2m))$ and inverting gives the bound on ϵ.

Final result

- Assembling the ingredients gives the result: with probability at least $1-\delta$ of random m samples the generalisation error of a function $h \in H$ chosen from a class H with VC dimension d with zero training error is bounded by

$$\epsilon = \epsilon(m, H, \delta) = \frac{2}{m}\left(d \log \frac{2em}{d} + \log \frac{2}{\delta}\right)$$

- Note that we can think of d as the complexity / capacity of the function class H.

Lower bounds

- VCdim *Characterises* Learnability in PAC setting: there exist distributions such that with probability at least δ over m random examples, the error of h is at least

$$\max\left(\frac{d-1}{32m}, \frac{1}{m}\log\left(\frac{1}{\delta}\right)\right).$$

Non-zero training error

- Very similar results can be obtained for non-zero training error.

- The main difference is the introduction of a square root to give a bound of the form

$$\epsilon(m, H, k, \delta) = k + O\left(\sqrt{\frac{d}{m}\log\frac{2em}{d}} + \sqrt{\frac{1}{m}\log\frac{2}{\delta}}\right)$$

  for k training errors, which is significantly worse than in the zero training error case.

- PAC-Bayes bounds now interpolate between these two.

Bias/variance decomposition

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

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

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

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 in statistics

In statistics 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)

Criticisms of PAC Theory

- The theory is certainly valid and the lower bounds indicate that it is not too far out – so can’t criticise as stands
- Criticism is that it doesn’t accord with experience of those applying learning.
- Mismatch between theory and practice.
- For example

Support Vector Machines (SVM)

One example of PAC failure is in analysing SVMs: linear functions in very high dimensional feature spaces.

1. kernel trick means we can work in an infinite dimensional feature space ($\Rightarrow$ infinite VC dimension) so that PAC result does not apply:

2. and YET very impressive performance

Support Vector Machines cont.

1. SVM seeks linear function in a feature space defined implicitly via a kernel κ:

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$$

2. For example the 1-norm SVM seeks $\mathbf{w}$ to solve

$$\begin{array}{ll} \min_{\mathbf{w},b,\gamma,\xi} & \|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \xi_i \\ \text{subject to} & y_i\left(\langle \mathbf{w}, \phi\left(\mathbf{x}_i\right)\rangle + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0, \\ & i = 1, \ldots, m. \end{array}$$

Margin in SVMs

- Intuition behind SVMs is that maximising the margin makes it possible to obtain good generalisation despite the high VC dimension

- The lower bound implies that we must be taking advantage of a benign distribution, since we know that in the worst case generalisation will be bad.

Margin in SVMs cont

- Hence, we require a theory that can give bounds that are sensitive to serendipitous distributions – in particular we conjecture that the margin is an indication of such ‘luckiness’.

- The proof approach will rely on using real-valued function classes. The margin gives an indication of the accuracy with which we need to approximate the functions when applying the statistics.

Covering Numbers

$\mathcal{F}$ a class of real functions defined on X and $\|\cdot\|_d$ a norm on $\mathcal{F}$, then

$$\mathcal{N}(\gamma, \mathcal{F}, \|\cdot\|_d)$$

is the smallest size set U_γ such that
for any $f \in \mathcal{F}$ there is a $u \in U_\gamma$ such that $\|f - u\|_d < \gamma$.

Covering Numbers cont.

For generalization bounds we need the γ-*growth function*,

$$\mathcal{N}^m(\gamma, \mathcal{F}) := \sup_{\mathbf{X} \in X^m} \mathcal{N}(\gamma, \mathcal{F}, \ell_\infty^{\mathbf{X}}).$$

where $\ell_\infty^{\mathbf{X}}$ gives the distance between two functions as the maximum difference between their outputs on the sample.

Second statistical result

- We want to bound the probability that the training examples can mislead us about one of the functions with margin bigger than fixed γ:

$$
\begin{aligned}
P^m\{\mathcal{X} \in X^m &: \exists f \in \mathcal{F} : \mathrm{err}_{\mathcal{X}}(f) = 0, m_{\mathcal{X}}(f) \geq \gamma, \mathrm{err}_P(f) \geq \epsilon\} \\
&\leq 2P^{2m}\{\mathcal{X}\mathcal{Y} \in X^{2m} : \exists f \in \mathcal{F} \text{ such that} \\
&\qquad \mathrm{err}_{\mathcal{X}}(f) = 0, m_{\mathcal{X}}(f) \geq \gamma, \mathrm{err}_{\mathcal{Y}}(f) \geq \epsilon/2\} \\
&\leq 2\mathcal{N}^{2m}(\gamma/2, \mathcal{F}) P^{2m}\{\mathcal{X}\mathcal{Y} \in X^{2m} : \text{ for fixed } f' \\
&\qquad m_{\mathcal{X}}(f') > \gamma/2, m_{\mathcal{Y}(\epsilon m/2)}(f') < \gamma/2\} \\
&\leq 2\mathcal{N}^{2m}(\gamma/2, \mathcal{F}) 2^{-\epsilon m/2} \leq \delta
\end{aligned}
$$

Second statistical result cont.

- inverting gives

$$\epsilon = \epsilon(m, \mathcal{F}, \delta, \gamma) = \frac{2}{m}\left(\log_2 \mathcal{N}^{2m}(\gamma/2, \mathcal{F}) + \log_2 \frac{2}{\delta}\right)$$

  i.e. with probability $1 - \delta$ over m random examples a margin γ hypothesis has error less than ϵ. Must apply for finite set of γ (‘do SRM over γ’).

Bounding the covering numbers

Have the following correspondences with the standard VC case (easy slogans):

Growth function	–	γ-growth function
Vapnik Chervonenkis dim	–	Fat shattering dim
Sauer's Lemma	–	Alon *et al.*

Covering numbers for linear functions

- For the case of linear functions there is a more direct route to bounding the covering numbers.

- We convert the $\gamma/2$ approximation on the sample problem into a classification problem, which is solvable with a margin of $\gamma/2$.

- It follows that if we use the perceptron algorithm to find a classifier, we will find a function satisfying the $\gamma/2$ approximation with just $8R^2/\gamma^2$ updates.

Covering numbers for linear functions

- This gives a sparse dual representation of the function. The covering is chosen as the set of functions with small sparse dual representations.

- Gives a bound on the size of the covering numbers of the form

$$\log_2 \mathcal{N}^{2m}(\gamma/2, \mathcal{F}) \leq k \log_2 \frac{e(2m+k-1)}{k}$$

where $k = \frac{8R^2}{\gamma^2}$.

Generalization of SVMs

For distribution with support in ball of radius R, (eg Gaussian Kernels $R = 1$) and margin γ, have bound:

$$\epsilon(m, \mathfrak{L}, \delta, \gamma) = \frac{2}{m}\left(k \log_2 \frac{e(2m + k - 1)}{k} + \log_2 \frac{m}{\delta}\right)$$

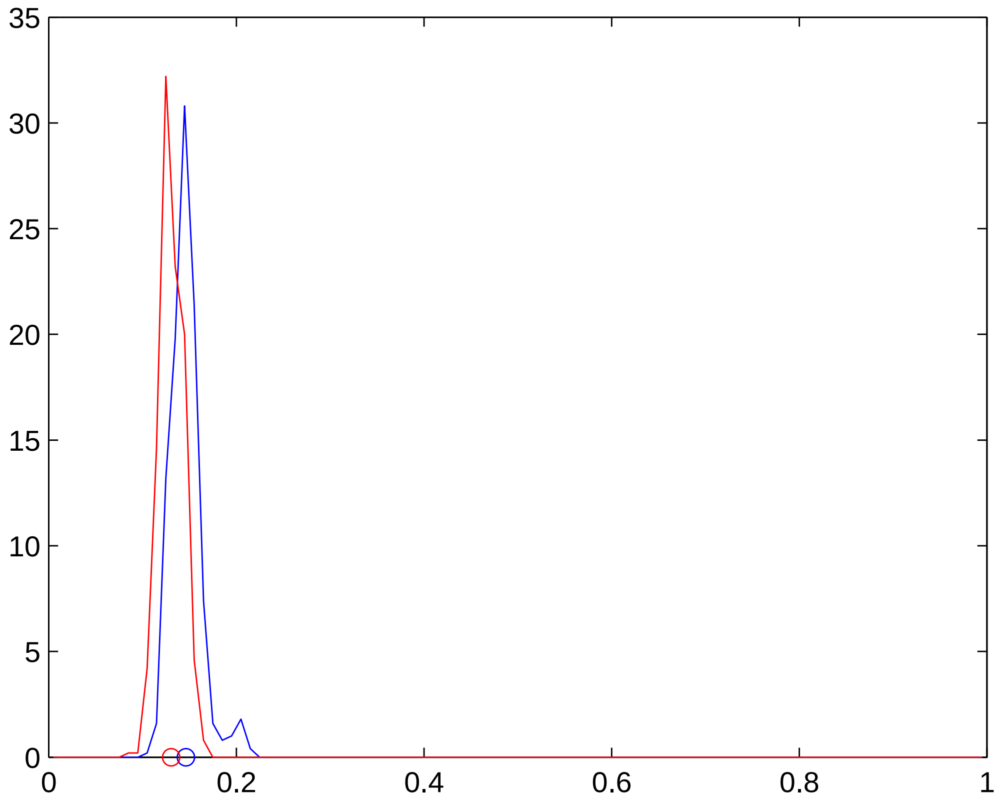
where $k = \frac{8R^2}{\gamma^2}$.

Controlling generalisation

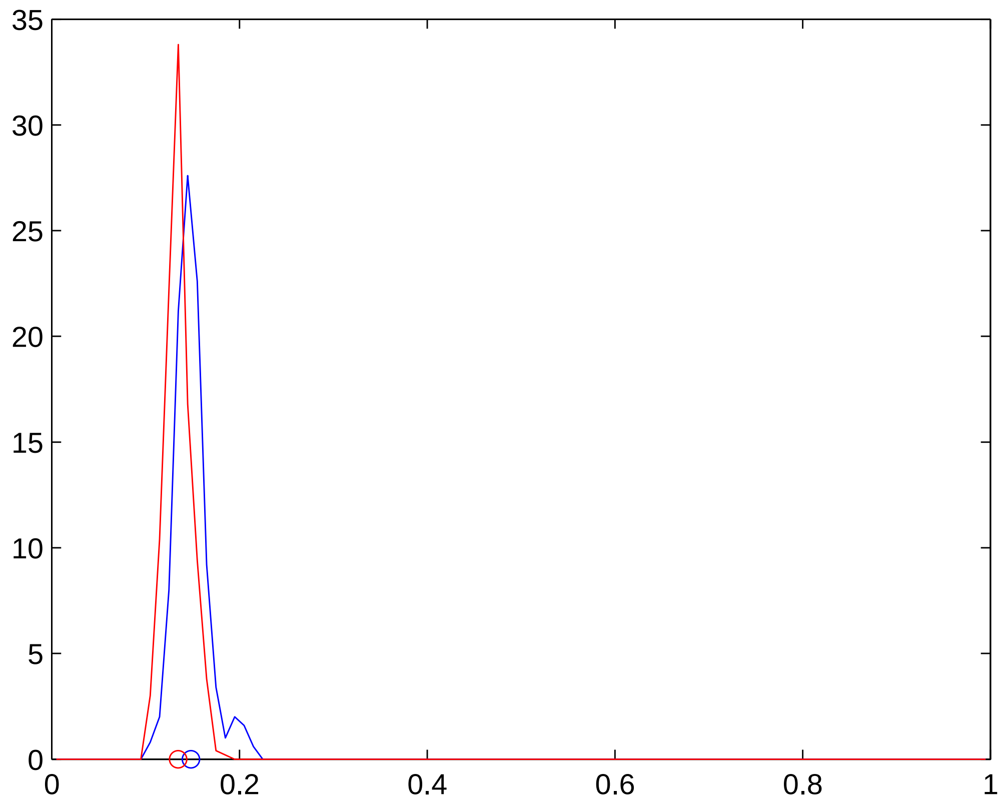
- Now consider using an SVM on the same data and compare the distribution of generalisations

- SVM distribution in red

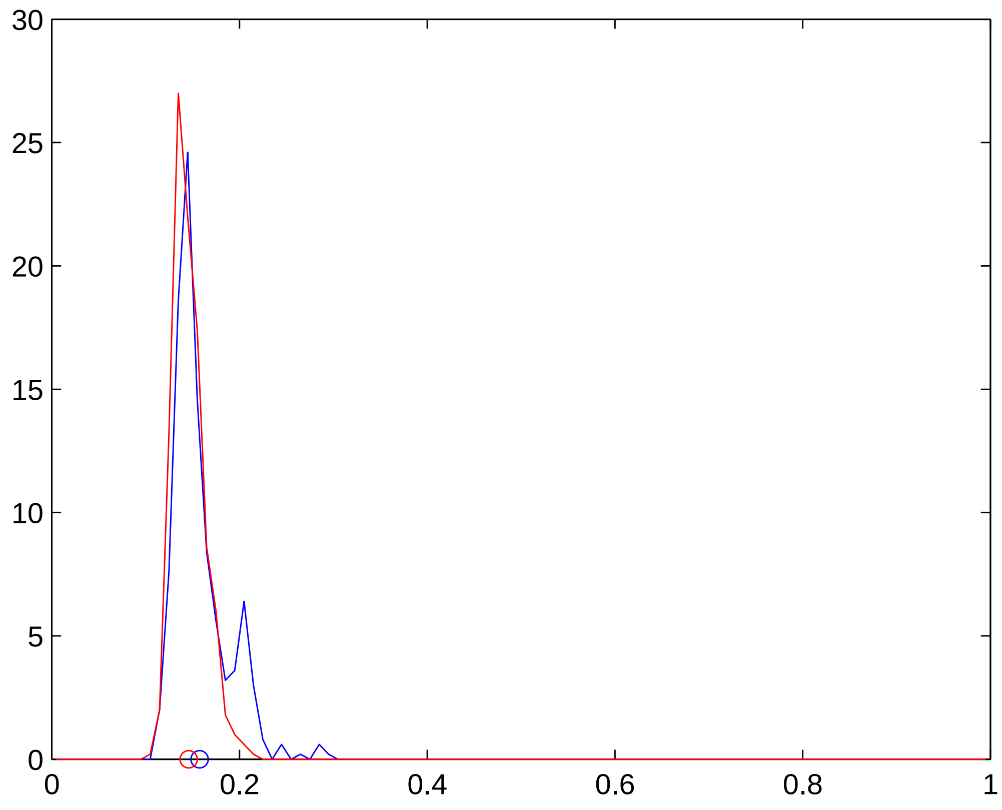
Error distribution: dataset size: 205



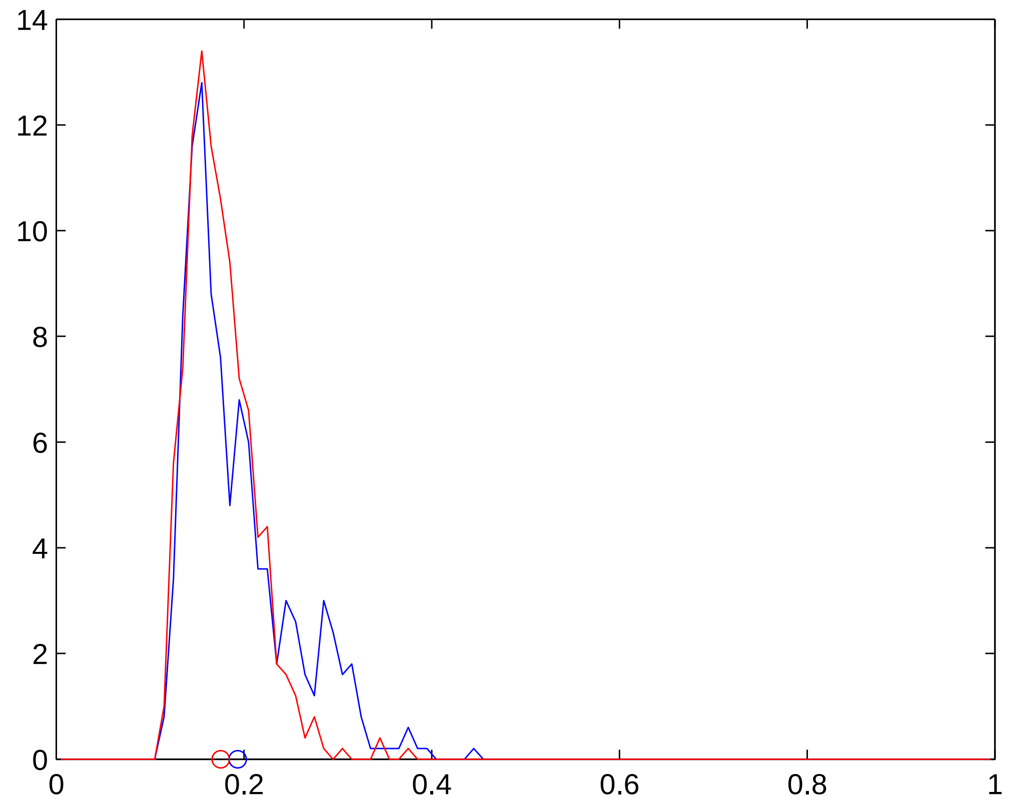
Error distribution: dataset size: 137



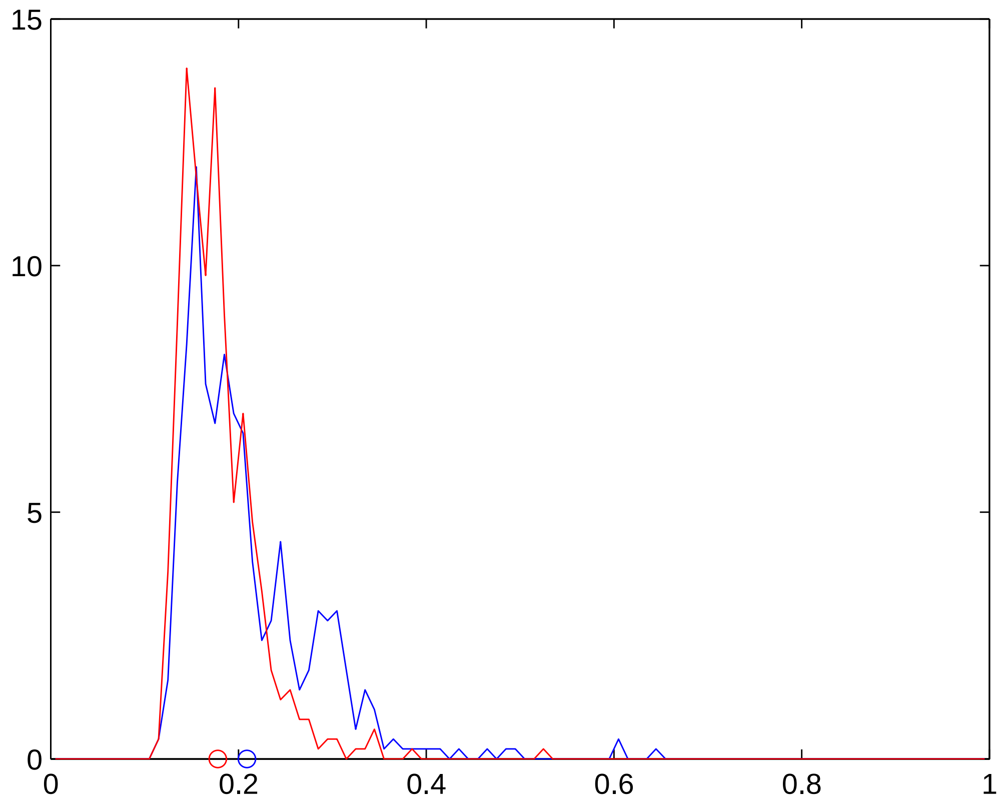
Error distribution: dataset size: 68



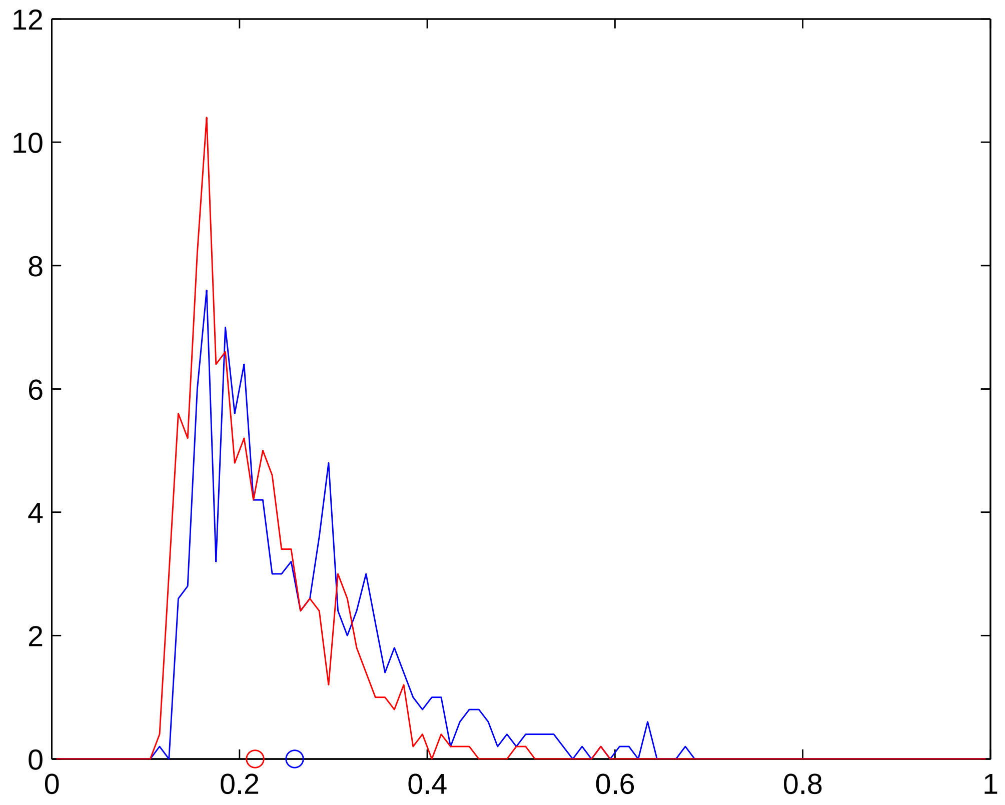
Error distribution: dataset size: 20



Error distribution: dataset size: 14



Error distribution: dataset size: 7



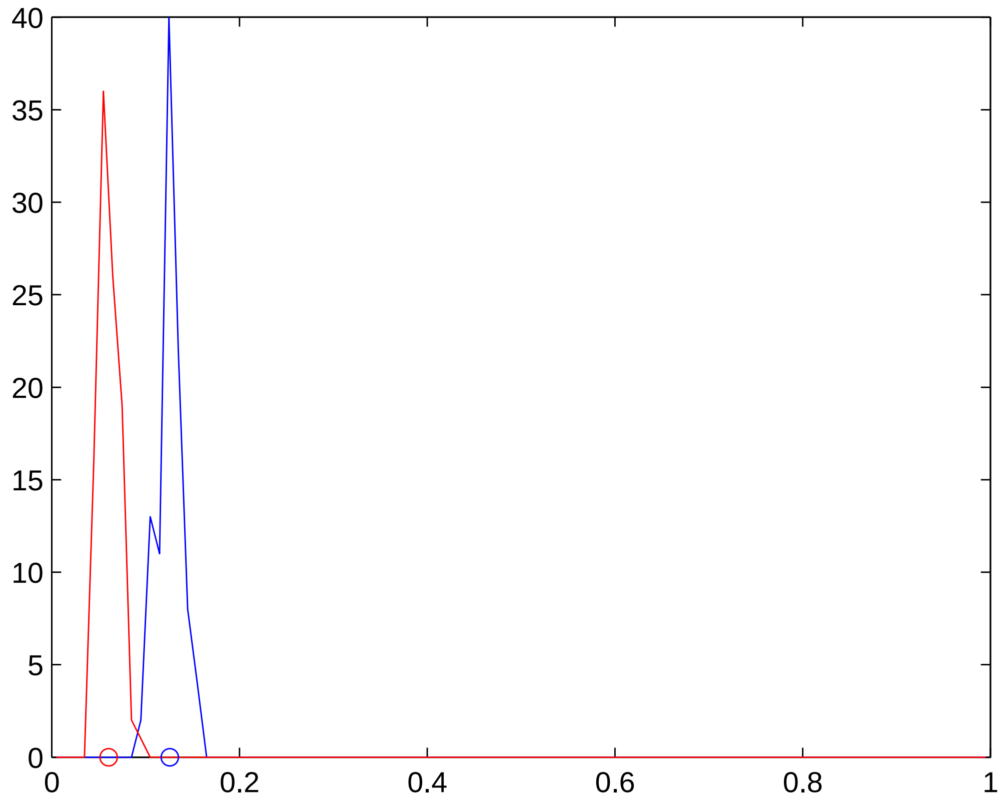
Using a kernel

- Can consider much higher dimensional spaces using the kernel trick

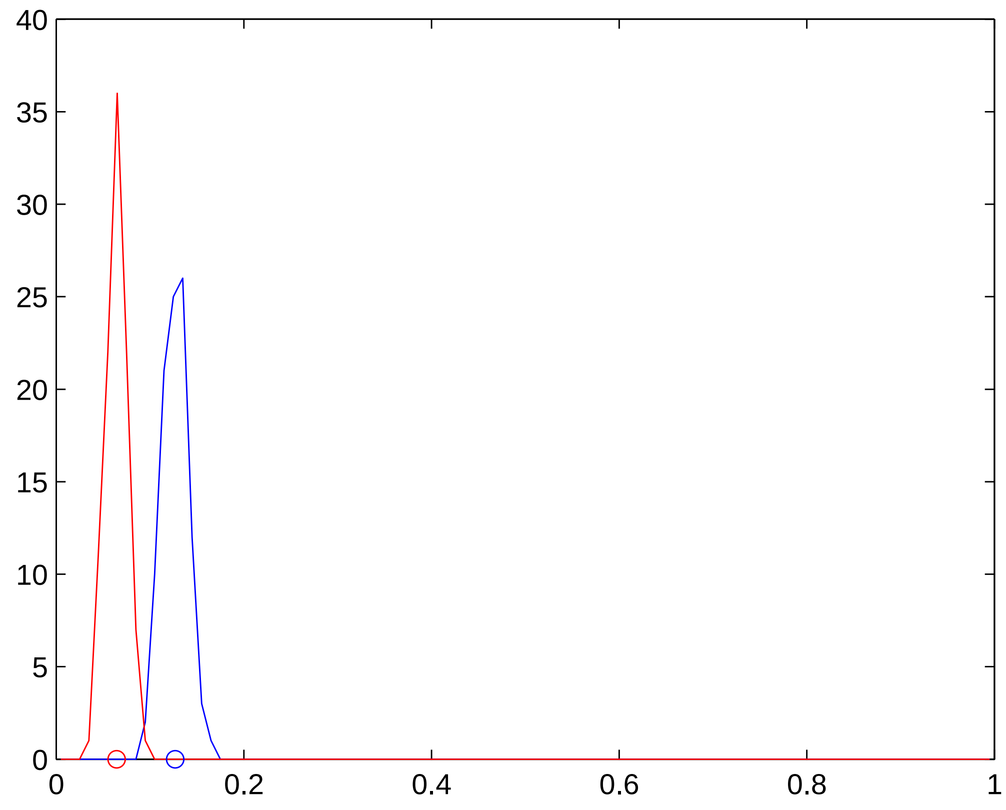
- Can even work in infinite dimensional spaces, eg using the Gaussian kernel:

$$\kappa(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right)$$

Error distribution: dataset size: 342



Error distribution: dataset size: 273



Subspace methods

- Principal components analysis: choose directions to maximise variance in the training data
- Canonical correlation analysis: choose directions to maximise correlations between two different views of the same objects
- Gram-Schmidt: greedily choose directions according to largest residual norms
- Partial least squares: greedily choose directions with maximal covariance with the target (will not cover this)

In all cases we need kernel versions in order to apply these methods in high-dimensional kernel defined feature spaces

Principal Components Analysis

- PCA is a subspace method – that is it involves projecting the data into a lower dimensional space.

- Subspace is chosen to ensure maximal variance of the projections:

$$\mathbf{w} = \mathsf{argmax}_{\mathbf{w}:\|\mathbf{w}\|=1}\mathbf{w}'\mathbf{X}'\mathbf{X}\mathbf{w}$$

- This is equivalent to maximising the Raleigh quotient:

$$\frac{\mathbf{w}'\mathbf{X}'\mathbf{X}\mathbf{w}}{\mathbf{w}'\mathbf{w}}$$

Principal Components Analysis

- We can optimise using Lagrange multipliers in order to remove the contraints:

$$L(\mathbf{w}, \lambda) = \mathbf{w}'\mathbf{X}'\mathbf{X}\mathbf{w} - \lambda \mathbf{w}'\mathbf{w}$$

  taking derivatives wrt $\mathbf{w}$ and setting equal to 0 gives:

$$\mathbf{X}'\mathbf{X}\mathbf{w} = \lambda \mathbf{w}$$

  implying $\mathbf{w}$ is an eigenvector of $\mathbf{X}'\mathbf{X}$.

- Note that

$$\lambda = \mathbf{w}'\mathbf{X}'\mathbf{X}\mathbf{w} = \sum_{i=1}^{m} \langle \mathbf{w}, \mathbf{x}_i \rangle^2$$

Principal Components Analysis

- So principal components analysis performs an eigenvalue decomposition of $\mathbf{X}'\mathbf{X}$ and projects into the space spanned by the first k eigenvectors

- Captures a total of

$$\sum_{i=1}^{k} \lambda_i$$

of the overall variance:

$$\sum_{i=1}^{m} \|\mathbf{x}_i\|^2 = \sum_{i=1}^{n} \lambda_i = \mathrm{tr}(\mathbf{K})$$

Kernel PCA

- We would like to find a dual representation of the principal eigenvectors and hence of the projection function.

- Suppose that $\mathbf{w}, \lambda \neq 0$ is an eigenvector/eigenvalue pair for $\mathbf{X}'\mathbf{X}$, then $\mathbf{X}\mathbf{w}, \lambda$ is for $\mathbf{X}\mathbf{X}'$:

$$(\mathbf{X}\mathbf{X}')\mathbf{X}\mathbf{w} = \mathbf{X}(\mathbf{X}'\mathbf{X})\mathbf{w} = \lambda\mathbf{X}\mathbf{w}$$

- and vice versa $\alpha, \lambda \rightarrow \mathbf{X}'\alpha, \lambda$

$$(\mathbf{X}'\mathbf{X})\mathbf{X}'\alpha = \mathbf{X}'(\mathbf{X}\mathbf{X}')\alpha = \lambda\mathbf{X}'\alpha$$

- Note that we get back to where we started if we do it twice.

Kernel PCA

- Hence, 1-1 correspondence between eigenvectors corresponding to non-zero eigenvalues, but note that if $\|\alpha\| = 1$

$$\|\mathbf{X}'\alpha\|^2 = \alpha'\mathbf{X}\mathbf{X}'\alpha = \alpha'\mathbf{K}\alpha = \lambda$$

so if $\alpha^i, \lambda_i, i = 1, \ldots, k$ are first k eigenvectors/values of $\mathbf{K}$

$$\frac{1}{\sqrt{\lambda_i}}\alpha^i$$

are dual representations of first k eigenvectors $\mathbf{w}^1, \ldots, \mathbf{w}^k$ of $\mathbf{X}'\mathbf{X}$ with same eigenvalues.

- Computing projections:

$$\langle \mathbf{w}^i, \phi(\mathbf{x}) \rangle = \frac{1}{\sqrt{\lambda_i}} \langle \mathbf{X}'\alpha^i, \phi(\mathbf{x}) \rangle = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{m} \alpha_j^i \kappa(\mathbf{x}_i, \mathbf{x})$$

Sample question

a) Define what is meant by a *valid* kernel function and give the positive semi-definite condition that ensures a kernel is valid?

b) Show that

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2$$

is a valid kernel for $\mathbf{x}, \mathbf{z} \in \mathbb{R}^d$.

c) Express the distance of the image of a point $\mathbf{x}$ from the centre of mass of a set of examples

$$S = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$$

in a kernel defined feature space.

Sample question (cont)

d) Using the result of part (c) give the algorithm in Matlab code or pseudocode that finds the indices of the test points that lie outside the smallest sphere containing all of the training data centred at the centre of mass. You may assume that the matrix $K(1 : m + n, 1 : m + n)$ has (i, j)th entry $\kappa(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \ldots, m + n$, where $\mathbf{x}_1, \ldots, \mathbf{x}_m$ are the training set and $\mathbf{x}_{m+1}, \ldots, \mathbf{x}_{m+n}$ are the test points.

e) Show that if we remove one point $\mathbf{x}_j$ from S, the centre of mass moves away from $\phi(\mathbf{x}_j)$ on the line from $\phi(\mathbf{x}_j)$ through the centre of mass by $\frac{1}{m-1}$ times the distance between them.

f) Hence or otherwise show that increasing the radius of the sphere by $\frac{m}{m-2}$ in the algorithm ensures that the leave one out error (an error occurs if the test point is not inside the sphere) on the training set is at most 1.