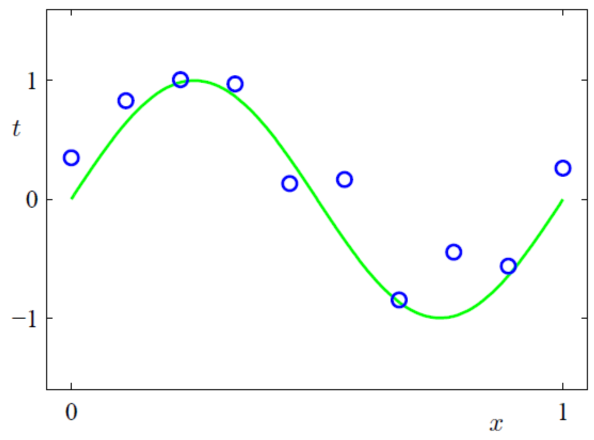
02457 Non-linear signal processing 2017 - Lecture 4

Lars Kai Hansen
Technical University of Denmark

Outline lecture 4

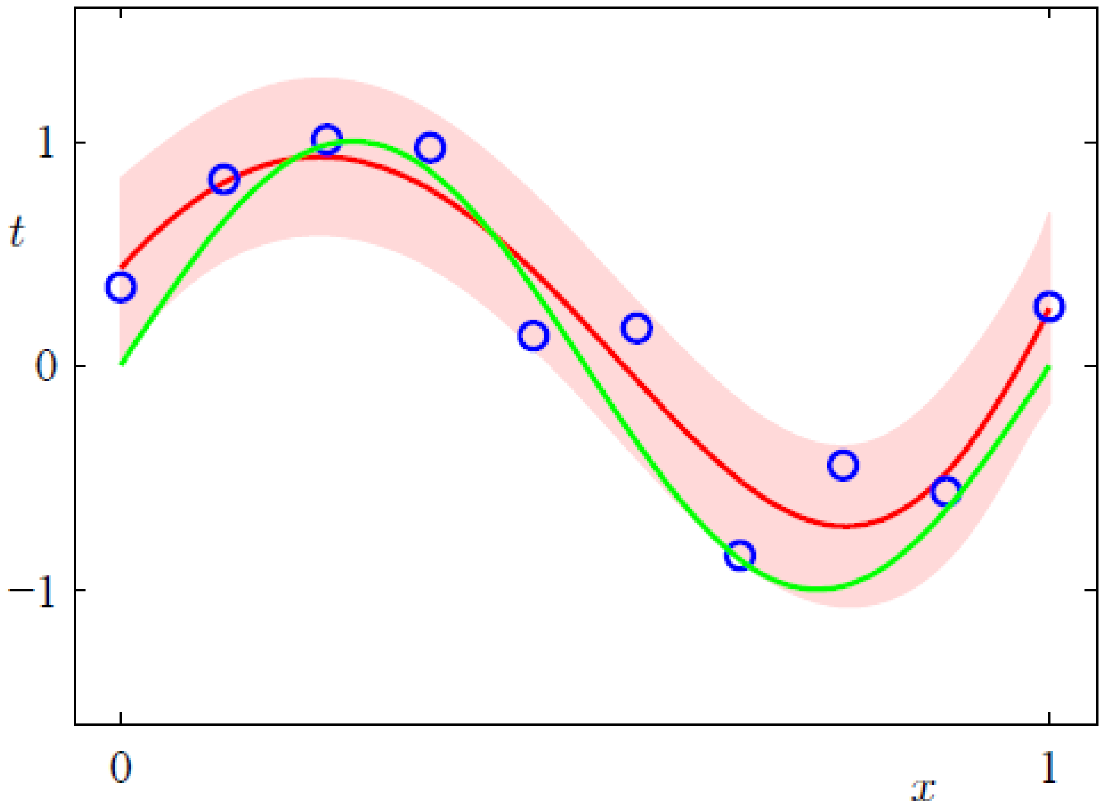


- Learning problem
- The likelihood function
- Linear regression
- Signal detection: Fisher linear discriminant

- The hidden agenda: Generalization
- The learning curve
- Priors: Weight decay
- Bias-variance dilemma

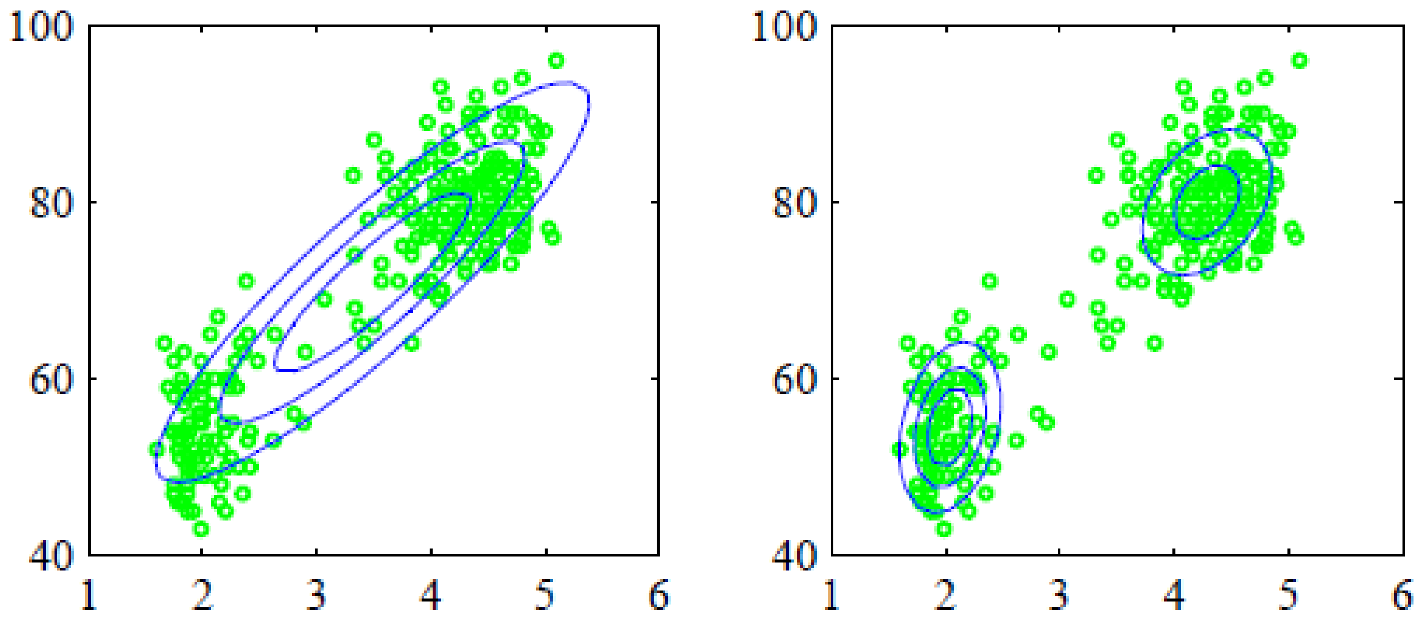
The supervised learning problem

- Supervised learning: Learning relations between sets of variables e.g. between input and output variables, conditional distributions $p(\text{output}|\text{input})$.



Unsupervised learning

Unsupervised learning: Learning the distribution of a set of variables $p(\text{input})$.



The Bayesian paradigm

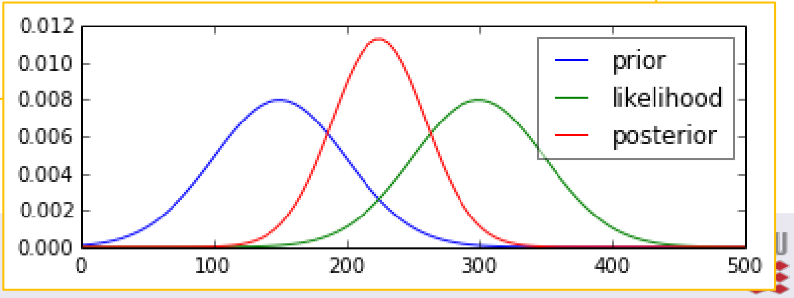
- The output density of the measured signals $(t, \mathbf{x})$ is modeled by a parameterized density: $p(t|\mathbf{x}) \sim p(t|\mathbf{x}, \mathbf{w})$.
- Let $\mathcal{D} = \{(t^1, \mathbf{x}^1), (t^2, \mathbf{x}^2), ..., (t^N, \mathbf{x}^N)\}$ be a *training set*
- Objective: Find the distribution of the parameter vector, $p(\mathbf{w}|\mathcal{D})$, hence the parameters are considered stochastic.

The likelihood function

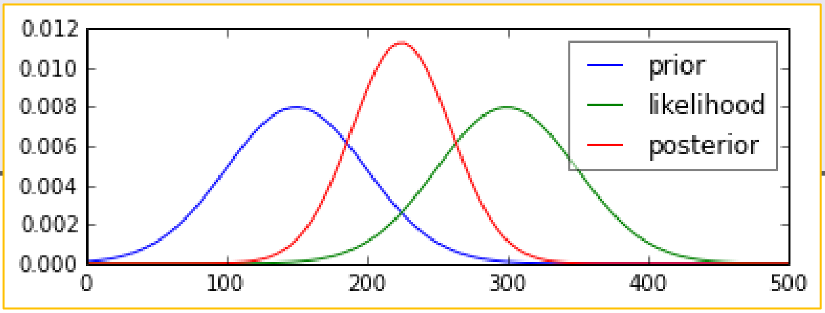
- Let $\mathcal{D} = \{(t^1, \mathbf{x}^1), (t^2, \mathbf{x}^2), ..., (t^N, \mathbf{x}^N)\}$ be the *training set*
- We use Bayes theorem

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$$

- The function $p(\mathcal{D}|\mathbf{w})$ is called the likelihood function (more correct the likelihood of the parameter vector $\mathbf{w}$). The density $p(\mathbf{w})$ is called the *a priori* or *prior* parameter distribution.



The likelihood function



$$\boxed{p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}}$$

- If the prior is “flat” in the neighborhood of the peak of $p(\mathcal{D}|\mathbf{w})$, we have

$$p(\mathbf{w}|\mathcal{D}) \propto p(\mathcal{D}|\mathbf{w})$$

- ...and finding the most probable parameters is equivalent to finding the maximum likelihood parameters.

Maximum likelihood and optimization

- For independent examples, $\mathcal{D} = \{(t^1, \mathbf{x}^1), (t^2, \mathbf{x}^2), ..., (t^N, \mathbf{x}^N)\}$, the likelihood function factorizes

$$p(\mathcal{D}|\mathbf{w}) = \prod_{n=1}^{N} p(t^n|\mathbf{x}^n, \mathbf{w})p(\mathbf{x}^n) = p(\mathcal{D}_t|\mathcal{D}_{\mathrm{x}}, \mathbf{w}) * p(\mathcal{D}_{\mathrm{x}})$$

- Many algorithms are based on minimizing an index or cost function

$$E(\mathbf{w}) = -\log p(\mathcal{D}_t|\mathcal{D}_{\mathrm{x}}, \mathbf{w}) = \sum_{n=1}^{N} -\log p(t^n|\mathbf{x}^n, \mathbf{w})$$

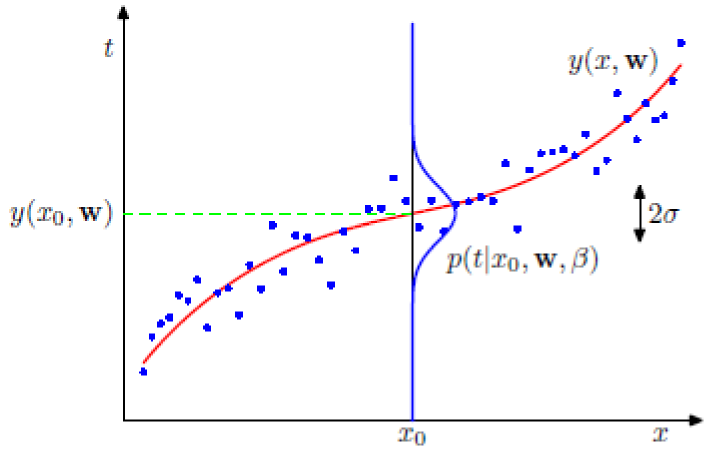
Maximum likelihood and least squares

$$t = y(\mathbf{x};\mathbf{w}) + \nu$$

$$p(t|\mathbf{x},\sigma^2,\mathbf{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(t - y(\mathbf{x};\mathbf{w}))^2\right)$$

$$p(\mathcal{D}_t|\mathcal{D}_{\mathbf{x}},\sigma^2,\mathbf{w}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(t^n - y(\mathbf{x}^n;\mathbf{w}))^2\right)$$

$$E(\mathbf{w},\sigma^2) = \frac{N}{2}\log 2\pi\sigma^2 + \frac{1}{2\sigma^2}\sum_{n=1}^{N}(t^n - y(\mathbf{x}^n;\mathbf{w}))^2$$

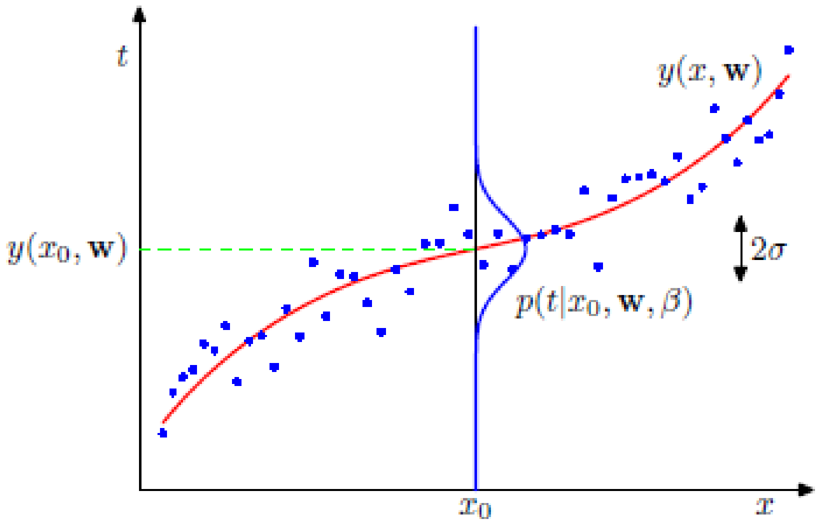


- Hence, maximizing the likelihood for Gaussian noise leads to a least squares problem (for $\mathbf{w}$).
- Note, the noise variance is given by ?

Estimate of noise variance in least squares

$$E(\mathbf{w}, \sigma^2) = \frac{N}{2} \log 2\pi\sigma^2 + \frac{1}{2\sigma^2} \sum_{n=1}^{N} (t^n - y(\mathbf{x}^n; \mathbf{w}))^2$$

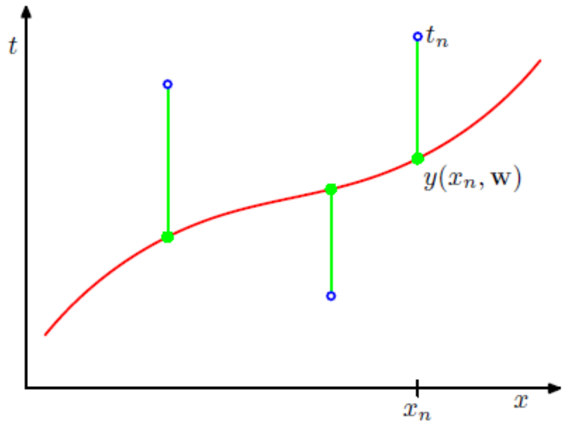
$$\widehat{\sigma^2} = \frac{1}{N} \sum_{n=1}^{N} (t^n - y(\mathbf{x}^n; \mathbf{w}))^2$$



Too optimistic for small N (variance too small)!

Linear regression

Figure 1.3 The error function (1.2) corresponds to (one half of) the sum of the squares of the displacements (shown by the vertical green bars) of each data point from the function $y(x, \mathbf{w})$.



- Let the function be linear

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$$

- Let a training set be given by $\mathcal{D} = \{(t^1, \mathbf{x}^1), ..., (t^N, \mathbf{x}^N)\}$, the sum-of-squares approximation error is given by

$$E = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}^n + w_0 - t^n)^2$$

- The optimal parameters are found by gradient based minimization,

$$\frac{\partial E}{\partial \mathbf{w}} = \sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}^n + w_0 - t^n)\mathbf{x}^n$$
$$\frac{\partial E}{\partial w_0} = \sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}^n + w_0 - t^n)$$

Linear regresion

- equations to solve

$$\sum_{n=1}^{N} (\mathbf{w}^T \mathbf{x}^n + w_0 - t^n)(\mathbf{x}^n)^T \;=\; 0$$
$$\sum_{n=1}^{N} (\mathbf{w}^T \mathbf{x}^n + w_0 - t^n) \;=\; 0$$

- the solution is given by in terms of $\boldsymbol{\mu} = (1/N)\sum \mathbf{x}^n$, and $\tau = (1/N)\sum t^n$

$$\mathbf{w} \;=\; \left(\frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}^n - \boldsymbol{\mu})(\mathbf{x}^n - \boldsymbol{\mu})^T \right)^{-1} \left(\frac{1}{N} \sum_{n=1}^{N} (t^n - \tau)\mathbf{x}^n \right)$$

$$w_0 \;=\; -\mathbf{w}^T \boldsymbol{\mu} + \tau$$

Signal detection: Bayes decision theory

- A signal detection system (or pattern classifier) provides a rule for assigning a measurement to a given signal category (class)
- Hence, a classifier divides measurement space (feature space) into disjoint regions $\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_c$, such that measurements that fall into region $\mathcal{R}_k$ are assigned with class $\mathcal{C}_k$.
- Boundaries between regions are denoted decision surfaces or decision boundaries

Signal Detection: Bayes decision theory

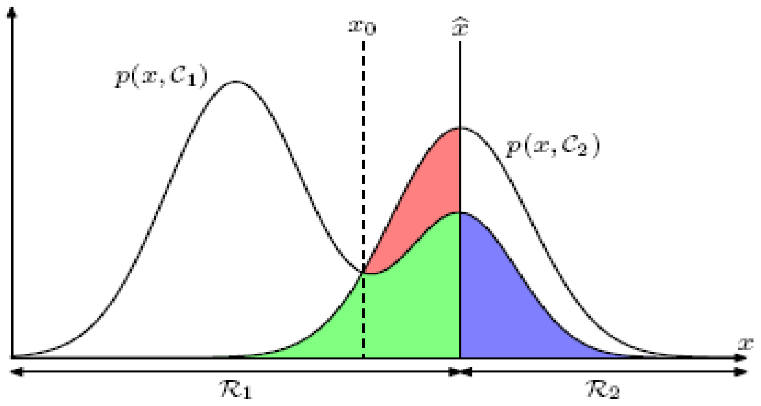


Figure 2: Schematic plot of the densities for a measured signal drawn from either of two populations $\mathcal{C}_1, \mathcal{C}_2$

$$\begin{aligned} P(\text{error}) &= P(x \in \mathcal{R}_2, \mathcal{C}_1) + P(x \in \mathcal{R}_1, \mathcal{C}_2) \\ &= P(x \in \mathcal{R}_2 | \mathcal{C}_1) P(\mathcal{C}_1) + P(x \in \mathcal{R}_1 | \mathcal{C}_2) P(\mathcal{C}_2) \\ &= \left(\int_{\mathcal{R}_2} p(x|\mathcal{C}_1) dx \right) P(\mathcal{C}_1) + \left(\int_{\mathcal{R}_1} p(x|\mathcal{C}_2) dx \right) P(\mathcal{C}_2) \end{aligned}$$

- The probability of error is minimized if we assign points to $\mathcal{R}_1$, whenever $p(x|\mathcal{C}_1)P(\mathcal{C}_1) > p(x|\mathcal{C}_2)P(\mathcal{C}_2)$

Signal detection

Discriminant Functions

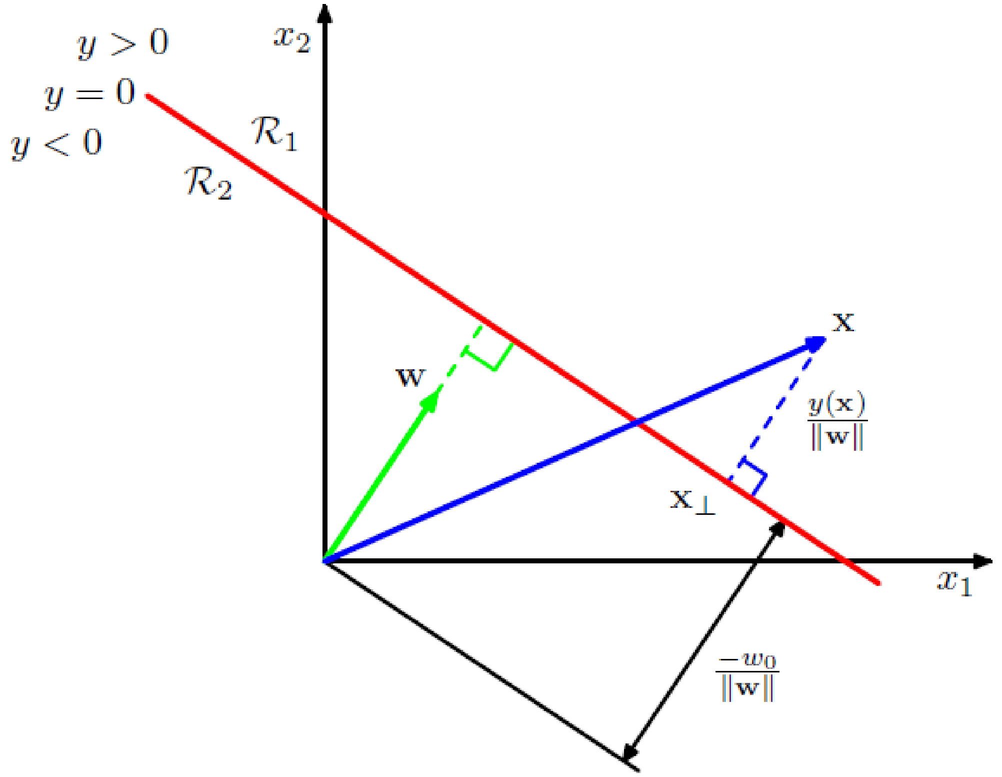
A discriminant is a function that takes an input vector $\mathbf{x}$ and assigns it to one of K classes, denoted $\mathcal{C}_k$. In this chapter, we shall restrict attention to *linear discriminants*, namely those for which the decision surfaces are hyperplanes. To simplify the discussion, we consider first the case of two classes and then investigate the extension to $K > 2$ classes.

The simplest representation of a linear discriminant function

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

Linear discriminant

Figure 4.1 Illustration of the geometry of a linear discriminant function in two dimensions. The decision surface, shown in red, is perpendicular to $\mathbf{w}$, and its displacement from the origin is controlled by the bias parameter w_0. Also, the signed orthogonal distance of a general point $\mathbf{x}$ from the decision surface is given by $y(\mathbf{x})/\|\mathbf{w}\|$.



Multiple classes

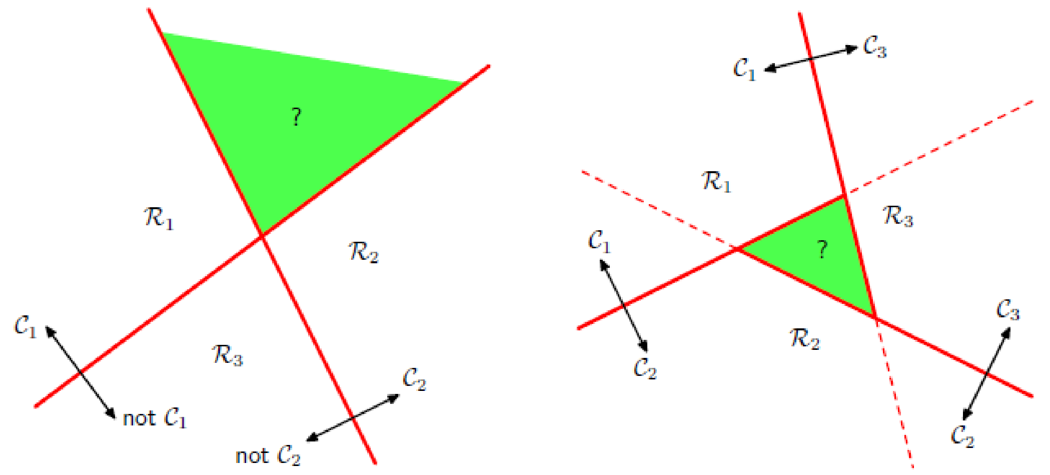


Figure 4.2 Attempting to construct a K class discriminant from a set of two class discriminants leads to ambiguous regions, shown in green. On the left is an example involving the use of two discriminants designed to distinguish points in class $\mathcal{C}_k$ from points not in class $\mathcal{C}_k$. On the right is an example involving three discriminant functions each of which is used to separate a pair of classes $\mathcal{C}_k$ and $\mathcal{C}_j$.

We can avoid these difficulties by considering a single K-class discriminant comprising K linear functions of the form

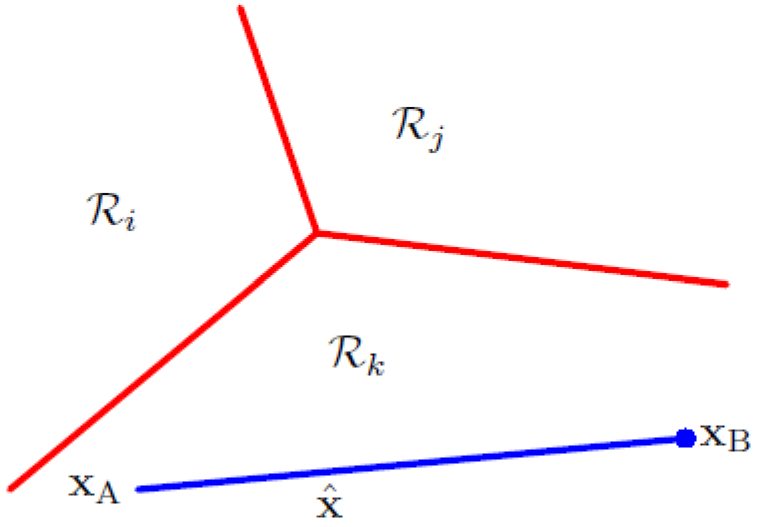
$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0} \tag{4.9}$$

and then assigning a point $\mathbf{x}$ to class $\mathcal{C}_k$ if $y_k(\mathbf{x}) > y_j(\mathbf{x})$ for all $j \neq k$. The decision boundary between class $\mathcal{C}_k$ and class $\mathcal{C}_j$ is therefore given by $y_k(\mathbf{x}) = y_j(\mathbf{x})$ and hence corresponds to a $(D-1)$-dimensional hyperplane defined by

$$(\mathbf{w}_k - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{k0} - w_{j0}) = 0. \tag{4.10}$$

Multiple linear decision surfaces

Figure 4.3 Illustration of the decision regions for a multiclass linear discriminant, with the decision boundaries shown in red. If two points $\mathbf{x}_A$ and $\mathbf{x}_B$ both lie inside the same decision region $\mathcal{R}_k$, then any point $\widehat{\mathbf{x}}$ that lies on the line connecting these two points must also lie in $\mathcal{R}_k$, and hence the decision region must be singly connected and convex.



$$\widehat{\mathbf{x}} = \lambda \mathbf{x}_A + (1-\lambda)\mathbf{x}_B$$

$$y_k(\widehat{\mathbf{x}}) = \lambda y_k(\mathbf{x}_A) + (1-\lambda) y_k(\mathbf{x}_B).$$

Fisher linear discriminant

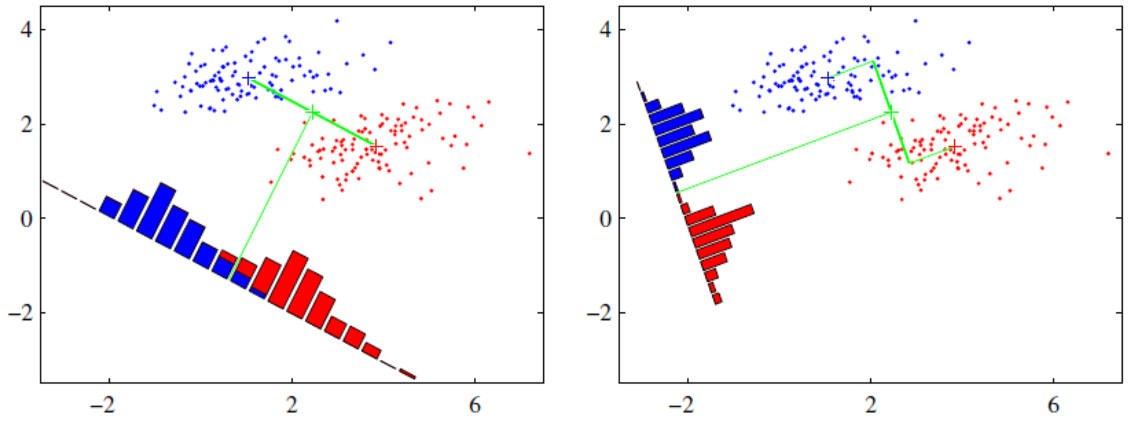


Figure 4.6 The left plot shows samples from two classes (depicted in red and blue) along with the histograms resulting from projection onto the line joining the class means. Note that there is considerable class overlap in the projected space. The right plot shows the corresponding projection based on the Fisher linear discriminant, showing the greatly improved class separation.

Consider first of all the case of two classes. The posterior probability for class $\mathcal{C}_1$ can be written as

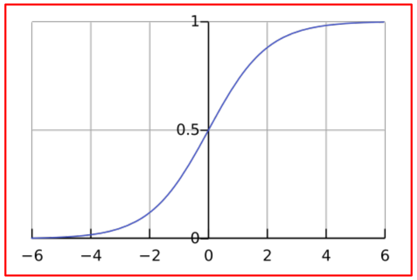
$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)+p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{1}{1+\exp(-a)} = \sigma(a) \end{aligned} \tag{4.57}$$

where we have defined

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$

and $\sigma(a)$ is the *logistic sigmoid* function defined by

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$



Fisher linear discriminant

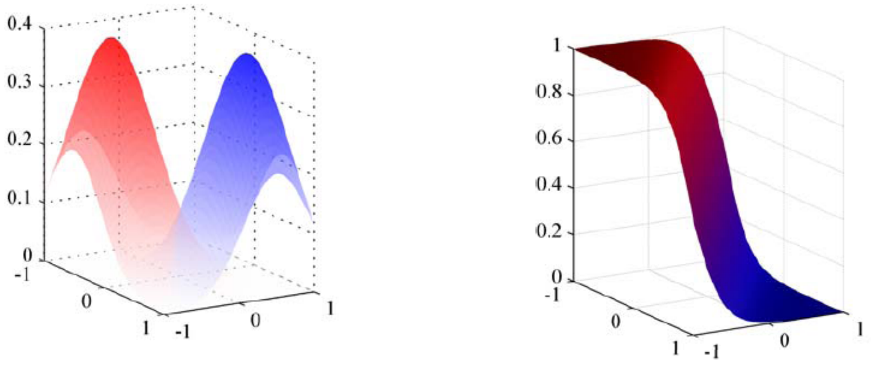


Figure 4.10 The left-hand plot shows the class-conditional densities for two classes, denoted red and blue. On the right is the corresponding posterior probability $p(\mathcal{C}_1|\mathbf{x})$, which is given by a logistic sigmoid of a linear function of $\mathbf{x}$. The surface in the right-hand plot is coloured using a proportion of red ink given by $p(\mathcal{C}_1|\mathbf{x})$ and a proportion of blue ink given by $p(\mathcal{C}_2|\mathbf{x}) = 1 - p(\mathcal{C}_1|\mathbf{x})$.

Let us assume that the class-conditional densities are Gaussian and then explore the resulting form for the posterior probabilities. To start with, we shall assume that all classes share the same covariance matrix. Thus the density for class $\mathcal{C}_k$ is given by

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}. \tag{4.64}$$

Consider first the case of two classes. From (4.57) and (4.58), we have

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0) \tag{4.65}$$

where we have defined

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{4.66}$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}. \tag{4.67}$$

Generalization

Generalization – the hidden agenda

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

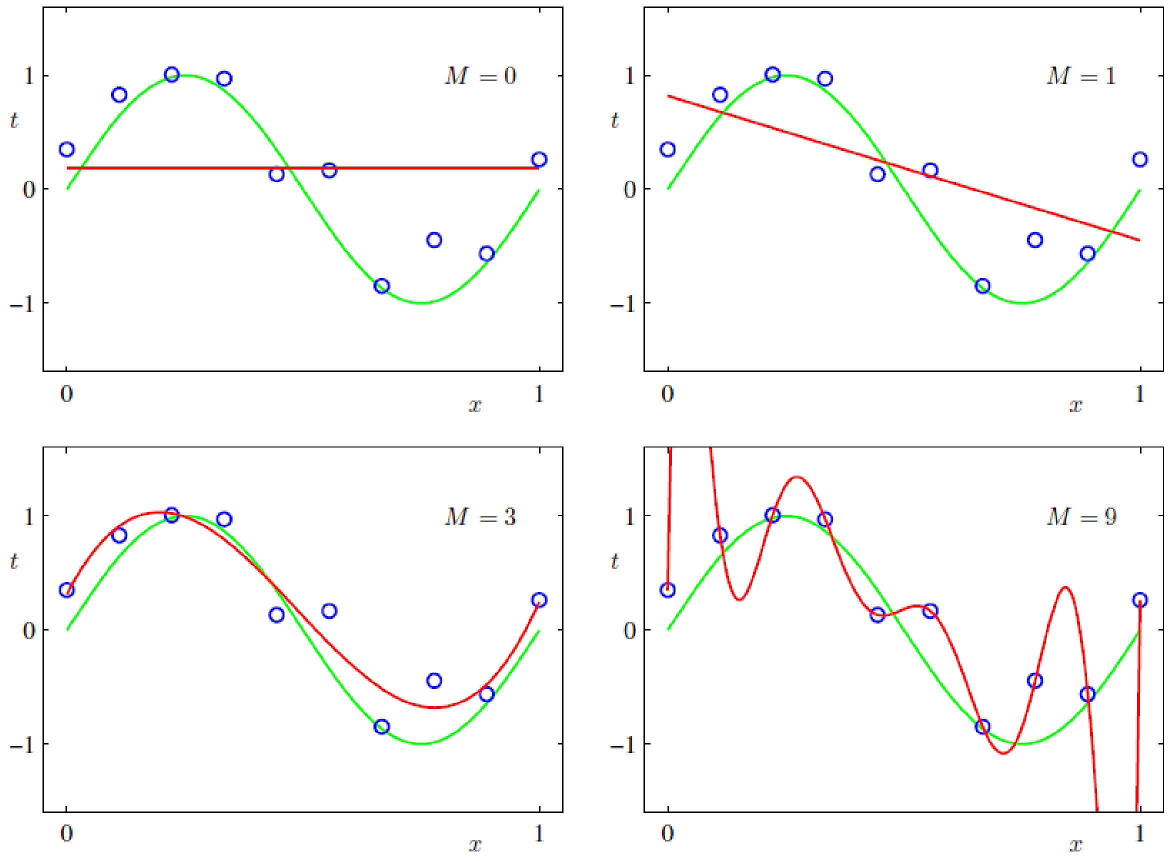


Figure 1.4 Plots of polynomials having various orders M, shown as red curves, fitted to the data set shown in Figure 1.2.

Training error vs test error

- Let a training set of independent examples be given by $\mathcal{D} = \{(t^1, \mathbf{x}^1), ..., (t^N, \mathbf{x}^N)\}$.
- The *training error pr. example* of the model $p(t|\mathbf{x}, \mathbf{w})$ is given by

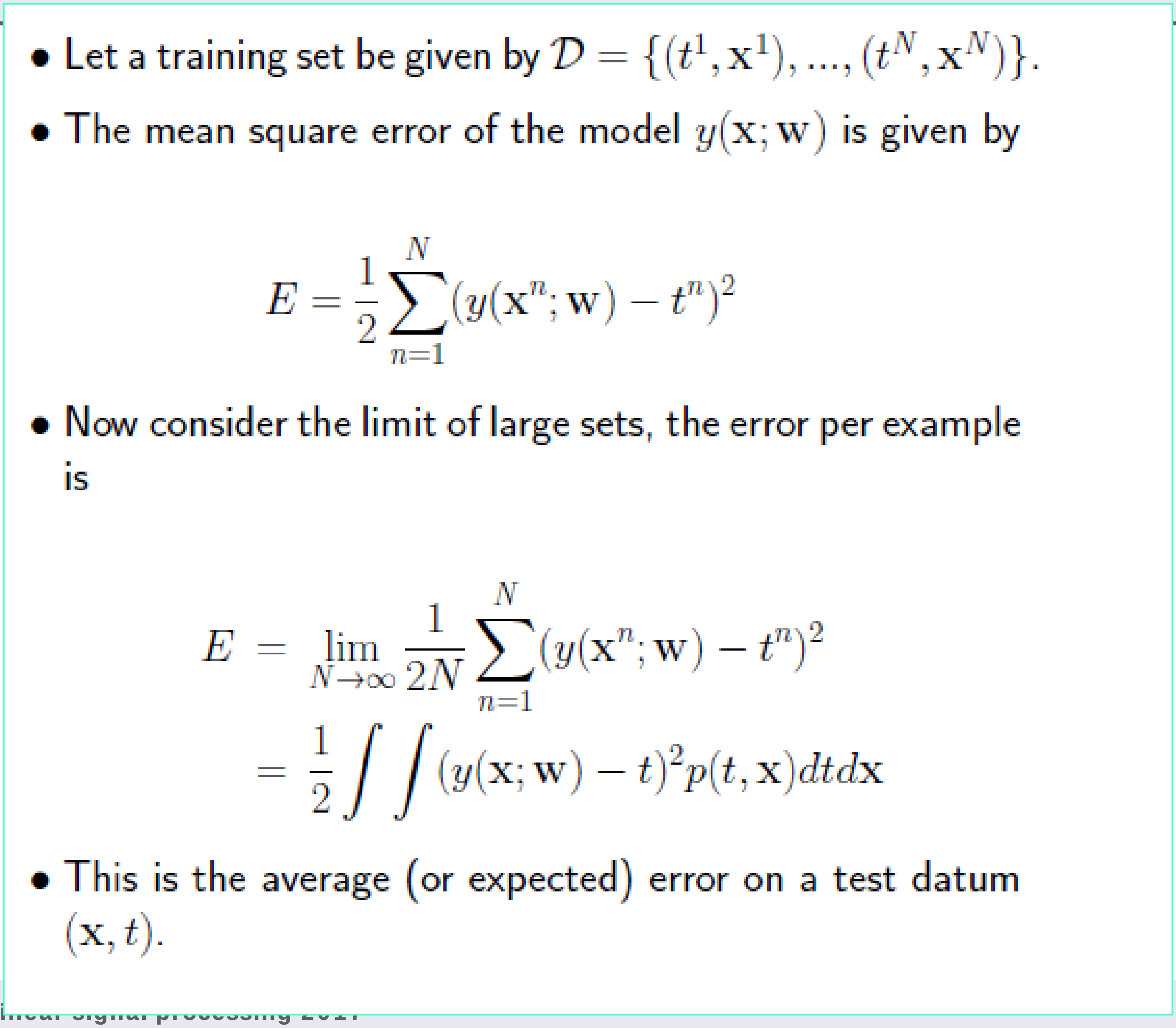
$$E = \frac{1}{N} \sum_{n=1}^{N} - \log p(t^n|\mathbf{x}^n, \mathbf{w})$$

  this is what we use to find good parameters $\mathbf{w}$.
- However, what we really want is that the probability of future data points is high, i.e., that the typical cost

$$E^k = - \log p(t^k|\mathbf{x}^k, \mathbf{w})$$

  is low. A model that assigns high probability to all future data point is close to the true model, hence, *a good generalizer*.

Limiting test / training error = generalization error



- Let a training set be given by $\mathcal{D} = \{(t^1, \mathbf{x}^1), ..., (t^N, \mathbf{x}^N)\}$.
- The mean square error of the model $y(\mathbf{x}; \mathbf{w})$ is given by

$$E = \frac{1}{2} \sum_{n=1}^{N} (y(\mathbf{x}^n; \mathbf{w}) - t^n)^2$$

- Now consider the limit of large sets, the error per example is

$$\begin{aligned} E &= \lim_{N \to \infty} \frac{1}{2N} \sum_{n=1}^{N} (y(\mathbf{x}^n; \mathbf{w}) - t^n)^2 \\ &= \frac{1}{2} \int \int (y(\mathbf{x}; \mathbf{w}) - t)^2 p(t, \mathbf{x}) dt d\mathbf{x} \end{aligned}$$

- This is the average (or expected) error on a test datum $(\mathbf{x}, t)$.

The generalization error

- So, let us define *the generalization error*:

$$E = \lim_{M\to\infty} \frac{1}{M} \sum_{k=1}^{M} - \log p(t^k|\mathrm{x}^k, \mathrm{w})$$

$$= \iint -\log[p(t|\boldsymbol{x},\boldsymbol{w})]p(t|\boldsymbol{x})p(\boldsymbol{x})dtd\boldsymbol{x}$$

This is the average (or expected) error on a test datum (t, x).

Generalization

Model capacity and test errors

- The Generalization error depends on the interplay between model flexibility and training set size
- The learning curve is the relation between generalization and training set size: $E_{\text{test}}(N)$ vs. N.
- The generalization error is determined by the complexity of the model and the amount of data N.
- The model complexity is controlled by *regularization* and by *parameter pruning*

Cross validation

Figure 1.18 The technique of S-fold cross-validation, illustrated here for the case of $S = 4$, involves taking the available data and partitioning it into S groups (in the simplest case these are of equal size). Then $S - 1$ of the groups are used to train a set of models that are then evaluated on the remaining group. This procedure is then repeated for all S possible choices for the held-out group, indicated here by the red blocks, and the performance scores from the S runs are then averaged.

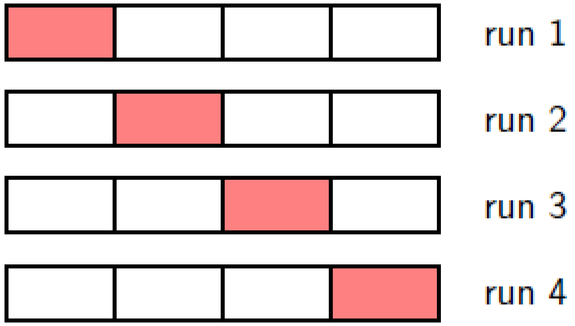
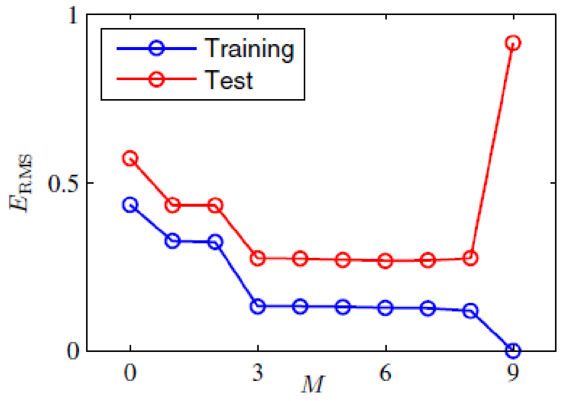


Figure 1.5 Graphs of the root-mean-square error, defined by (1.3), evaluated on the training set and on an independent test set for various values of M.



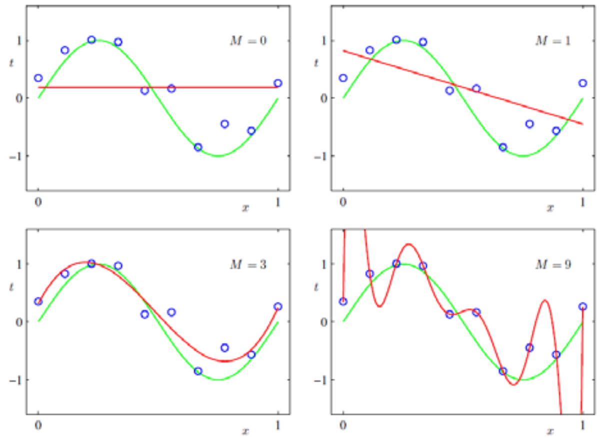
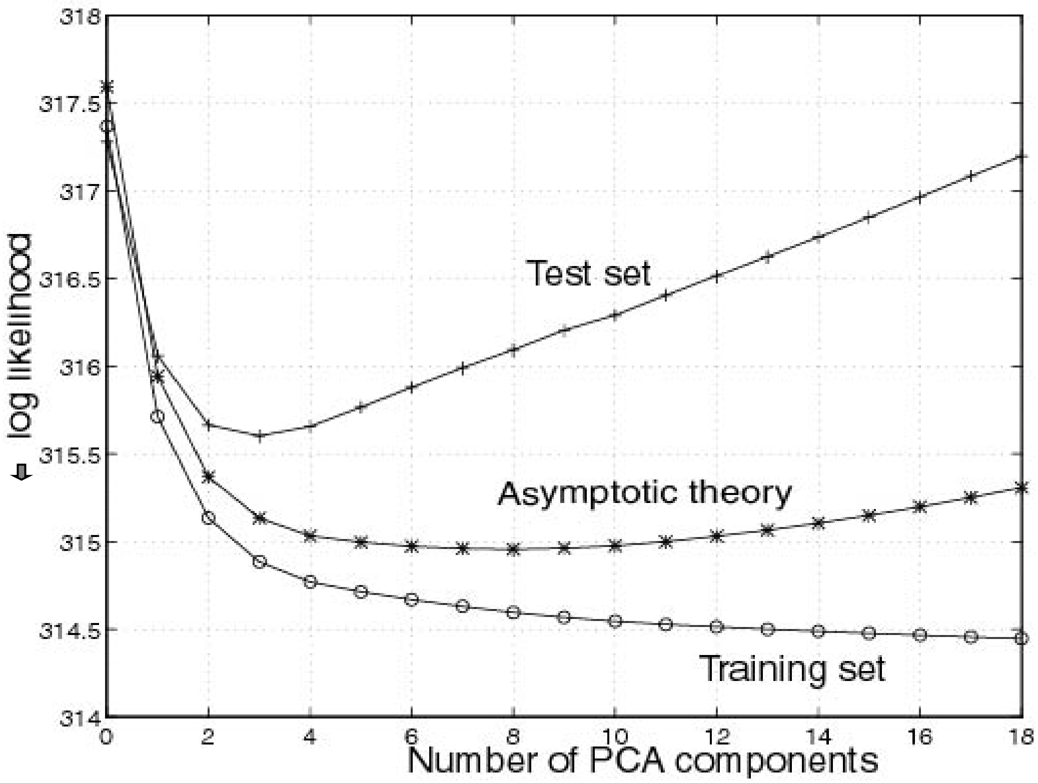


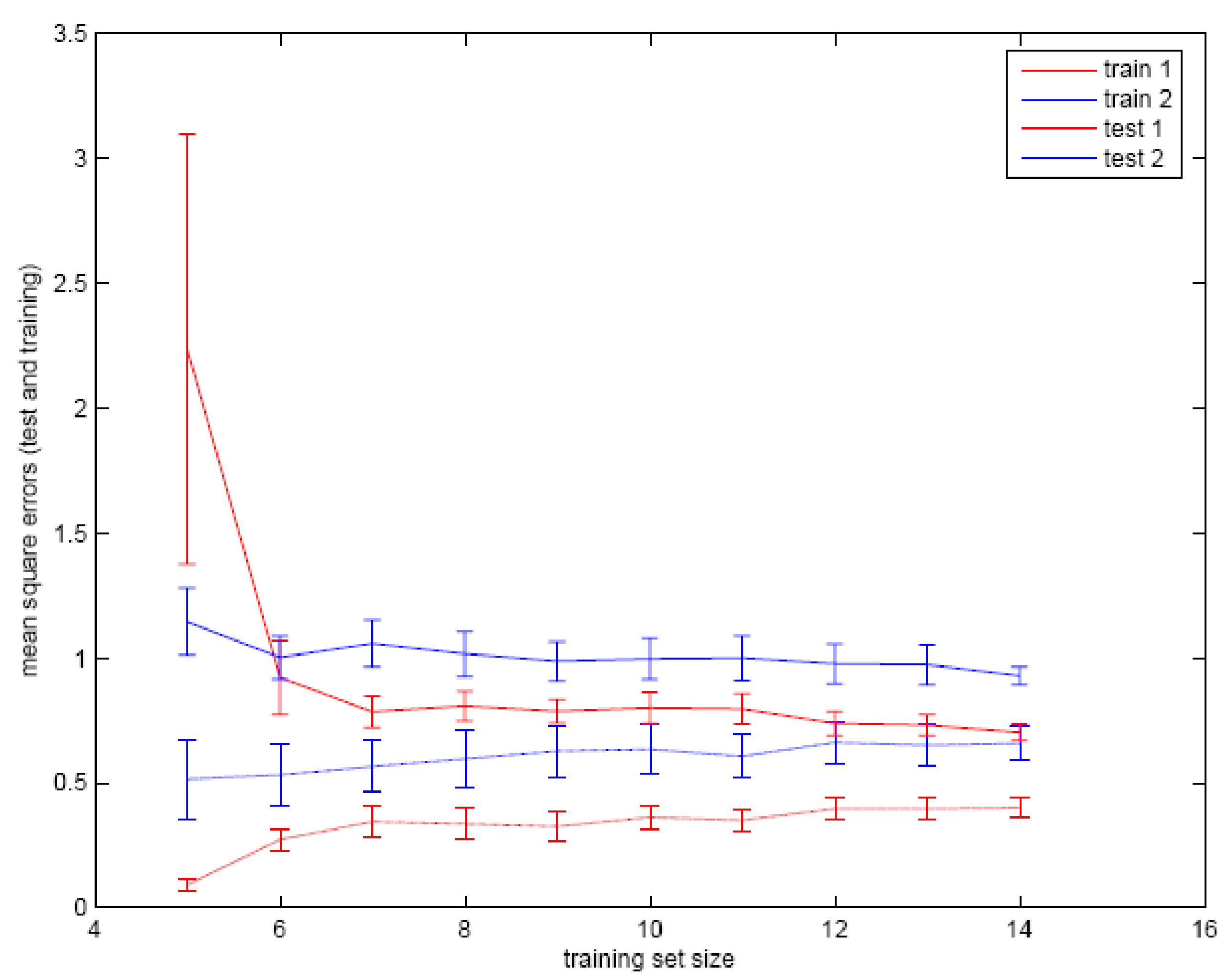
Figure 1.4 Plots of polynomials having various orders M, shown as red curves, fitted to the data set shown in Figure 1.2.

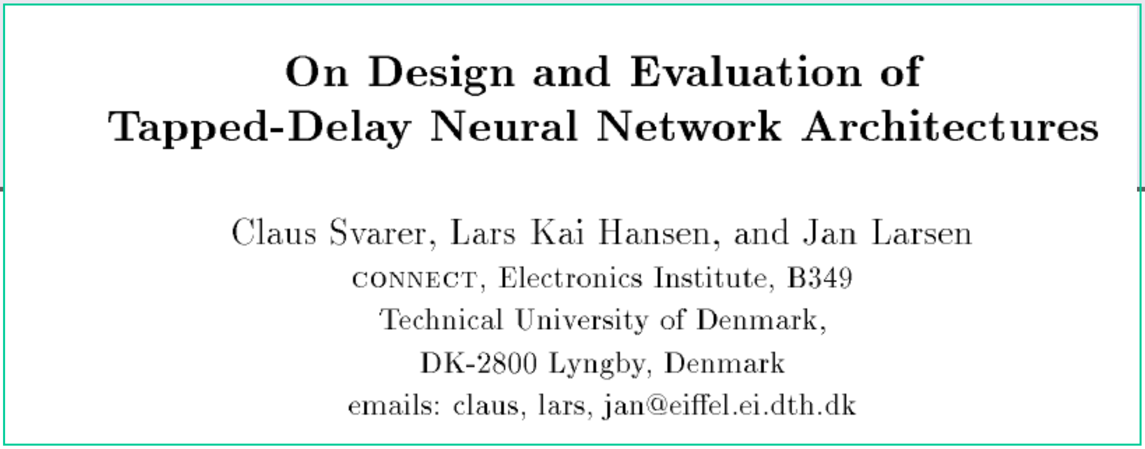
Bias-variance trade-off as function of PCA dimension



Hansen et al. *NeuroImage* (1999)

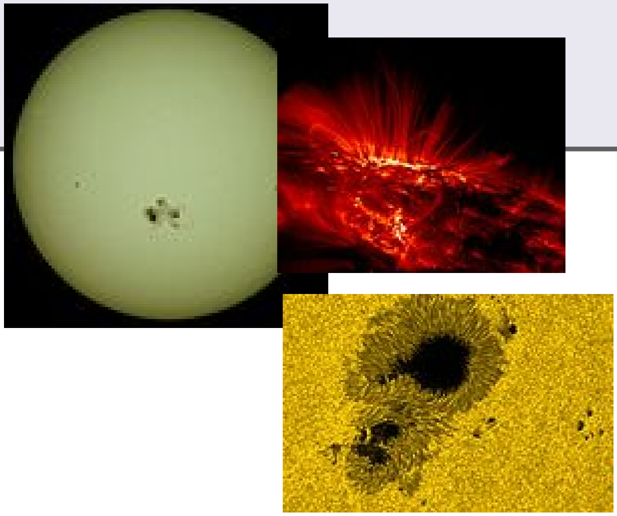
Learning curve: Test error vs sample size

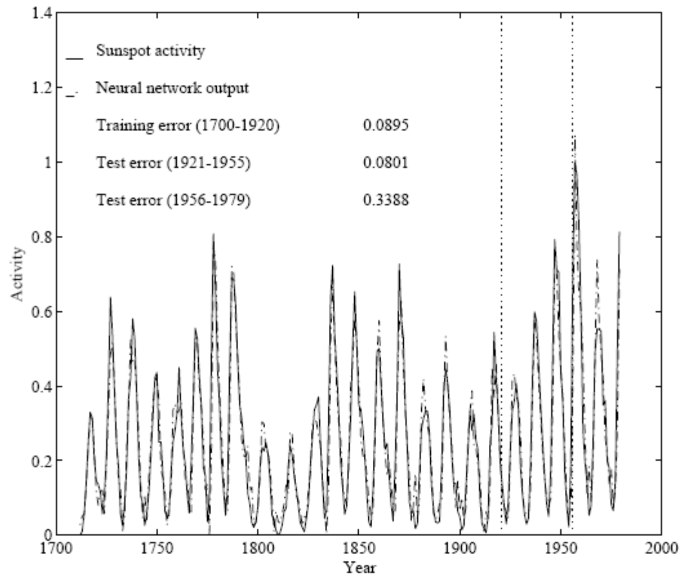


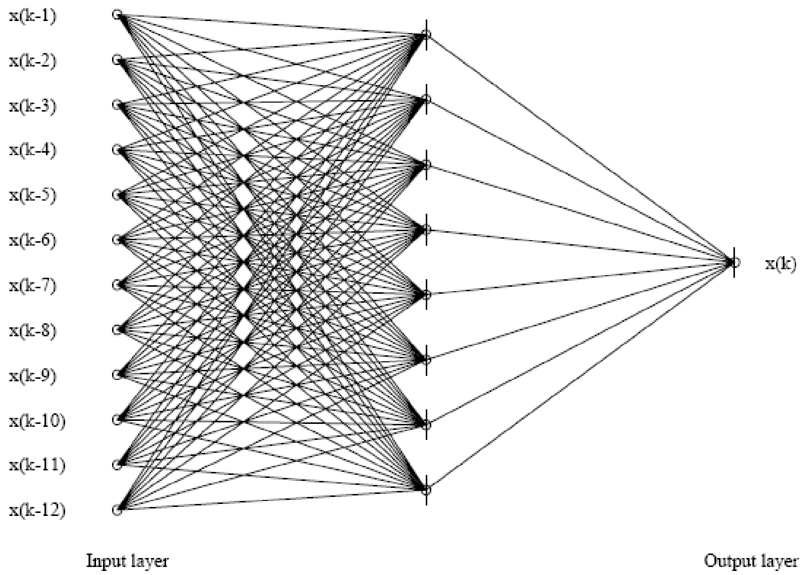


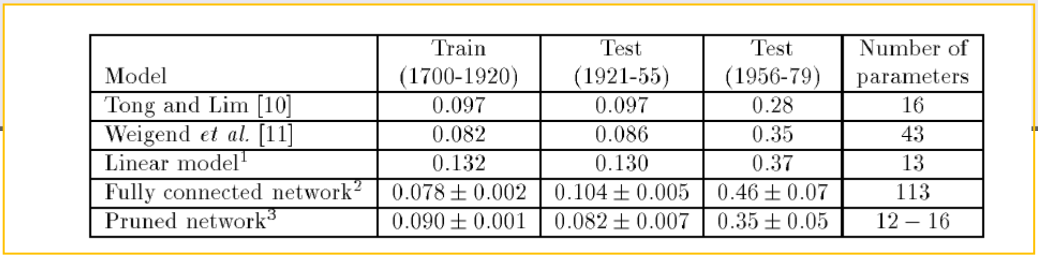
On Design and Evaluation of Tapped-Delay Neural Network Architectures

Claus Svarer, Lars Kai Hansen, and Jan Larsen
CONNECT, Electronics Institute, B349
Technical University of Denmark,
DK-2800 Lyngby, Denmark
emails: claus, lars, jan@eiffel.ei.dth.dk

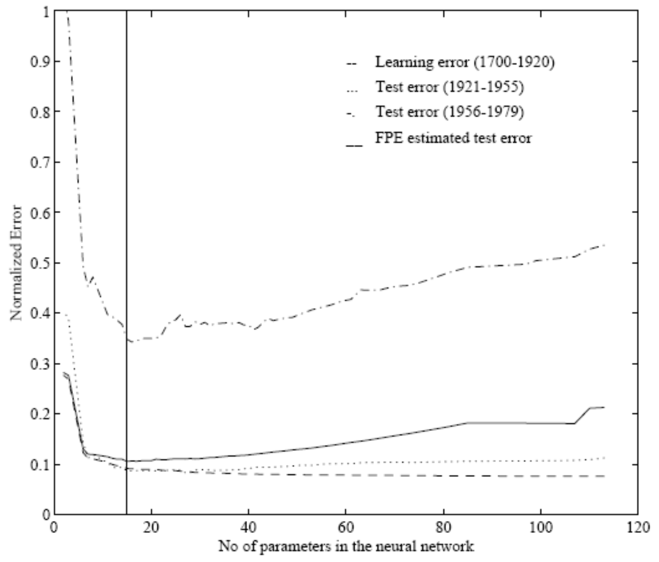


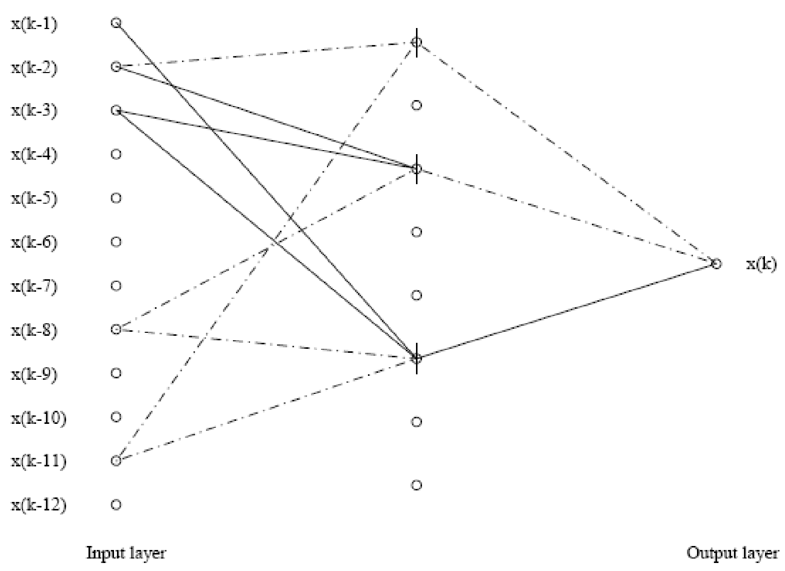




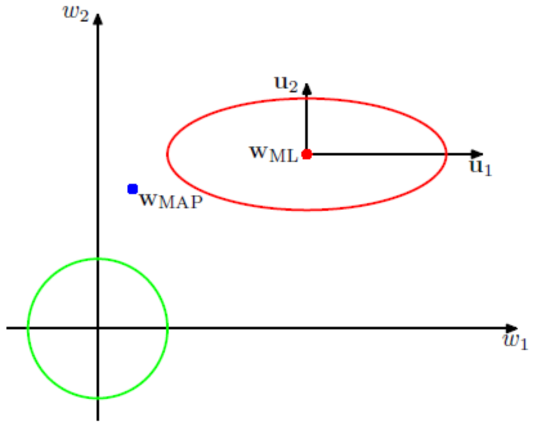


Model	Train (1700-1920)	Test (1921-55)	Test (1956-79)	Number of parameters
Tong and Lim [10]	0.097	0.097	0.28	16
Weigend *et al.* [11]	0.082	0.086	0.35	43
Linear model[1]	0.132	0.130	0.37	13
Fully connected network[2]	0.078 ± 0.002	0.104 ± 0.005	0.46 ± 0.07	113
Pruned network[3]	0.090 ± 0.001	0.082 ± 0.007	0.35 ± 0.05	$12 - 16$





Bayesian priors and regularization



In Section 1.1, we introduced the idea of adding a regularization term to an error function in order to control over-fitting, so that the total error function to be minimized takes the form

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w}) \tag{3.24}$$

where λ is the regularization coefficient that controls the relative importance of the data-dependent error $E_D(\mathbf{w})$ and the regularization term $E_W(\mathbf{w})$. One of the simplest forms of regularizer is given by the sum-of-squares of the weight vector elements

$$E_W(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}. \tag{3.25}$$

If we also consider the sum-of-squares error function given by

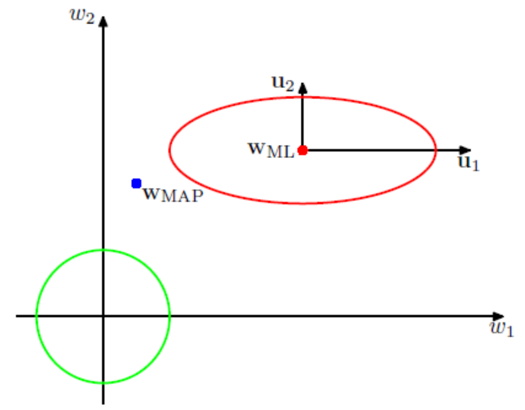
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 \tag{3.26}$$

then the total error function becomes

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}. \tag{3.27}$$

Regularization (by "weight decay")

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$



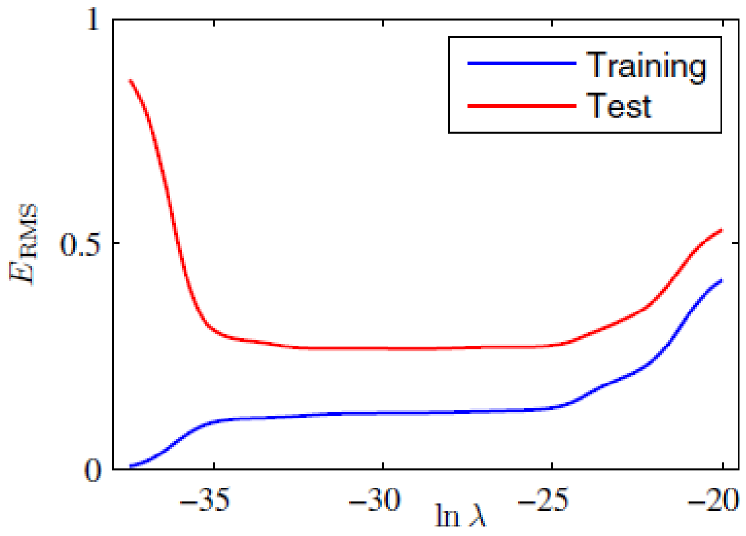
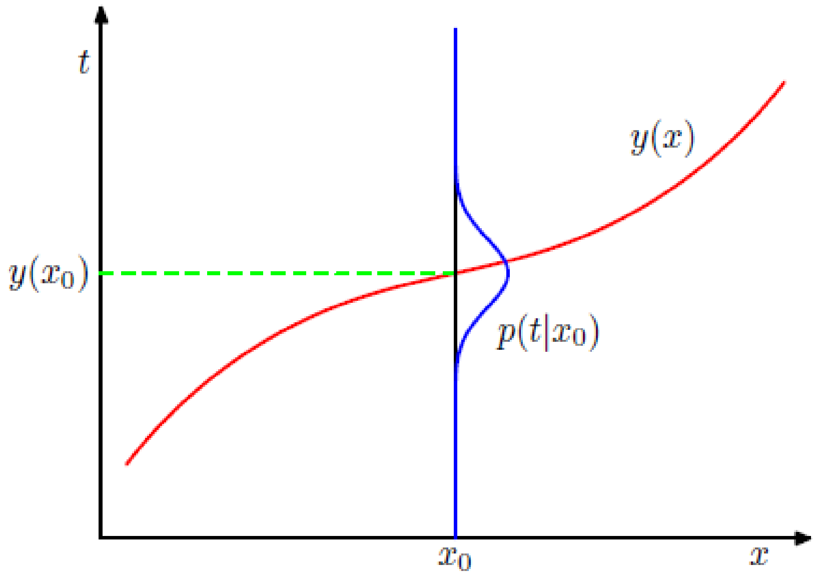


Figure 1.8 Graph of the root-mean-square error (1.3) versus $\ln\lambda$ for the $M = 9$ polynomial.

Optimal regression (minimize test error)

Figure 1.28 The regression function $y(x)$, which minimizes the expected squared loss, is given by the mean of the conditional distribution $p(t|x)$.



$$\mathbb{E}[L] = \iint \{y(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t.$$

$$\int t p(t|\mathbf{x})\,\mathrm{d}t = \mathbb{E}_t[t|\mathbf{x}]$$

$$\begin{aligned}\{y(\mathbf{x}) - t\}^2 &= \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}] + \mathbb{E}[t|\mathbf{x}] - t\}^2 \\ &= \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}^2 + 2\{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}\{\mathbb{E}[t|\mathbf{x}] - t\} + \{\mathbb{E}[t|\mathbf{x}] - t\}^2\end{aligned}$$

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int \{\mathbb{E}[t|\mathbf{x}] - t\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x}. \qquad (1.90)$$

Bias-variance dilemma

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x})\,\mathrm{d}t.$$

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t. \qquad (3.37)$$

Consider the integrand of the first term in (3.37), which for a particular data set $\mathcal{D}$ takes the form

$$\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2. \qquad (3.38)$$

Because this quantity will be dependent on the particular data set $\mathcal{D}$, we take its average over the ensemble of data sets. If we add and subtract the quantity $\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]$

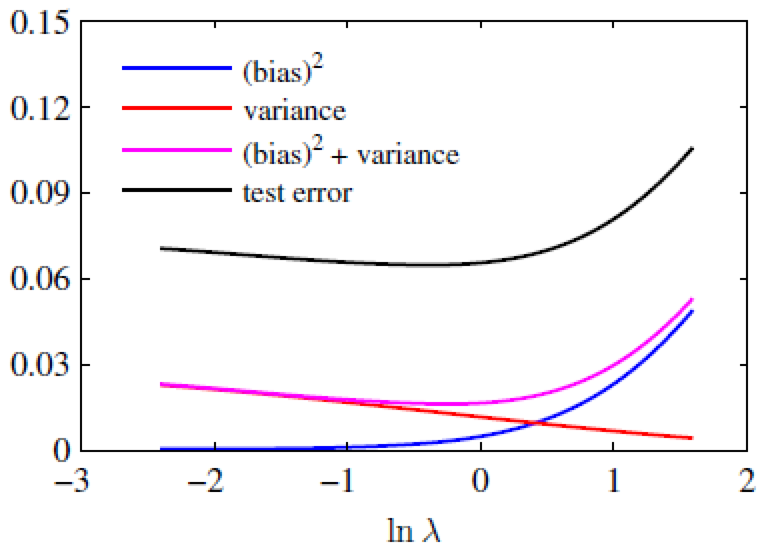
$$\begin{aligned}
&\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\
&\quad = \{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\
&\quad\quad + 2\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}. \qquad (3.39)
\end{aligned}$$

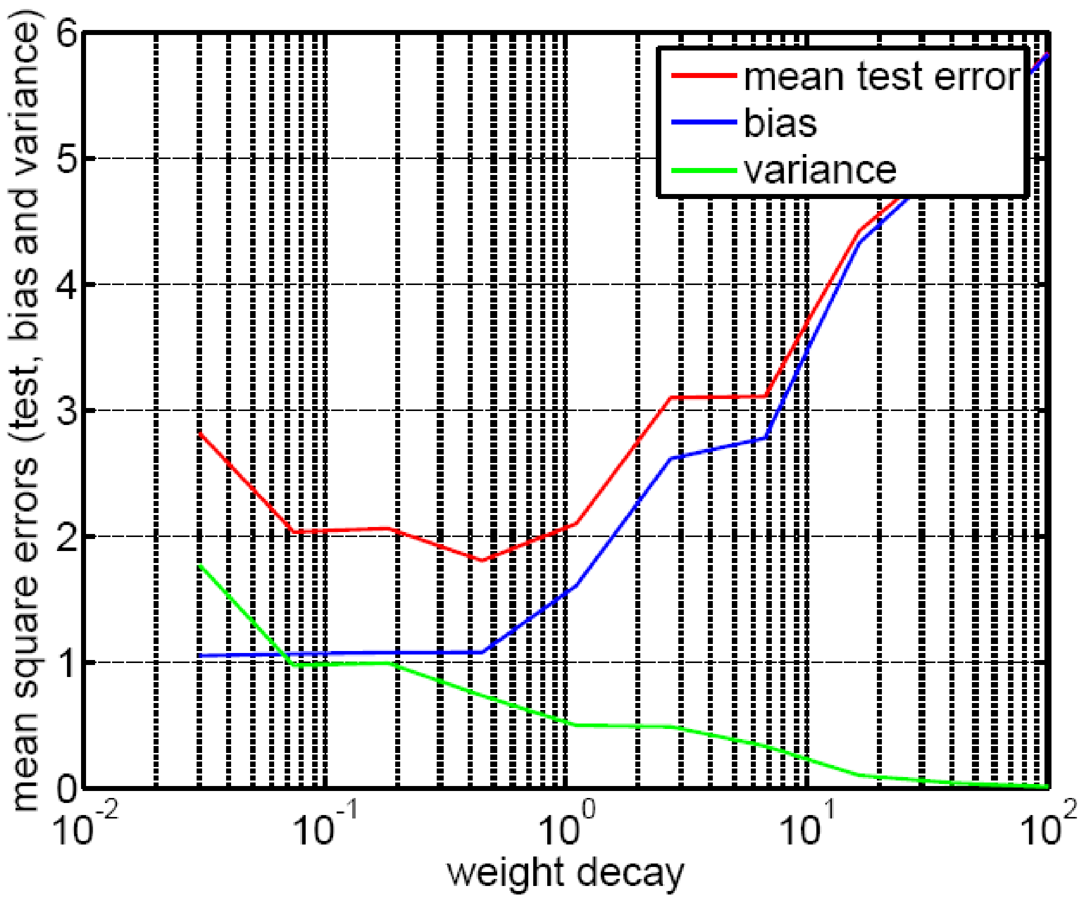
We now take the expectation of this expression with respect to $\mathcal{D}$ and note that the final term will vanish, giving

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2\right] \\
&\quad = \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2\right]}_{\text{variance}}. \qquad (3.40)
\end{aligned}$$

Bias – variance dilemma

Figure 3.6 Plot of squared bias and variance, together with their sum, corresponding to the results shown in Figure 3.5. Also shown is the average test set error for a test data set size of 1000 points. The minimum value of $(\text{bias})^2 + \text{variance}$ occurs around $\ln\lambda = -0.31$, which is close to the value that gives the minimum error on the test data.




mean test error
bias
variance
mean square errors (test, bias and variance)
weight decay
0
1
2
3
4
5
6
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}