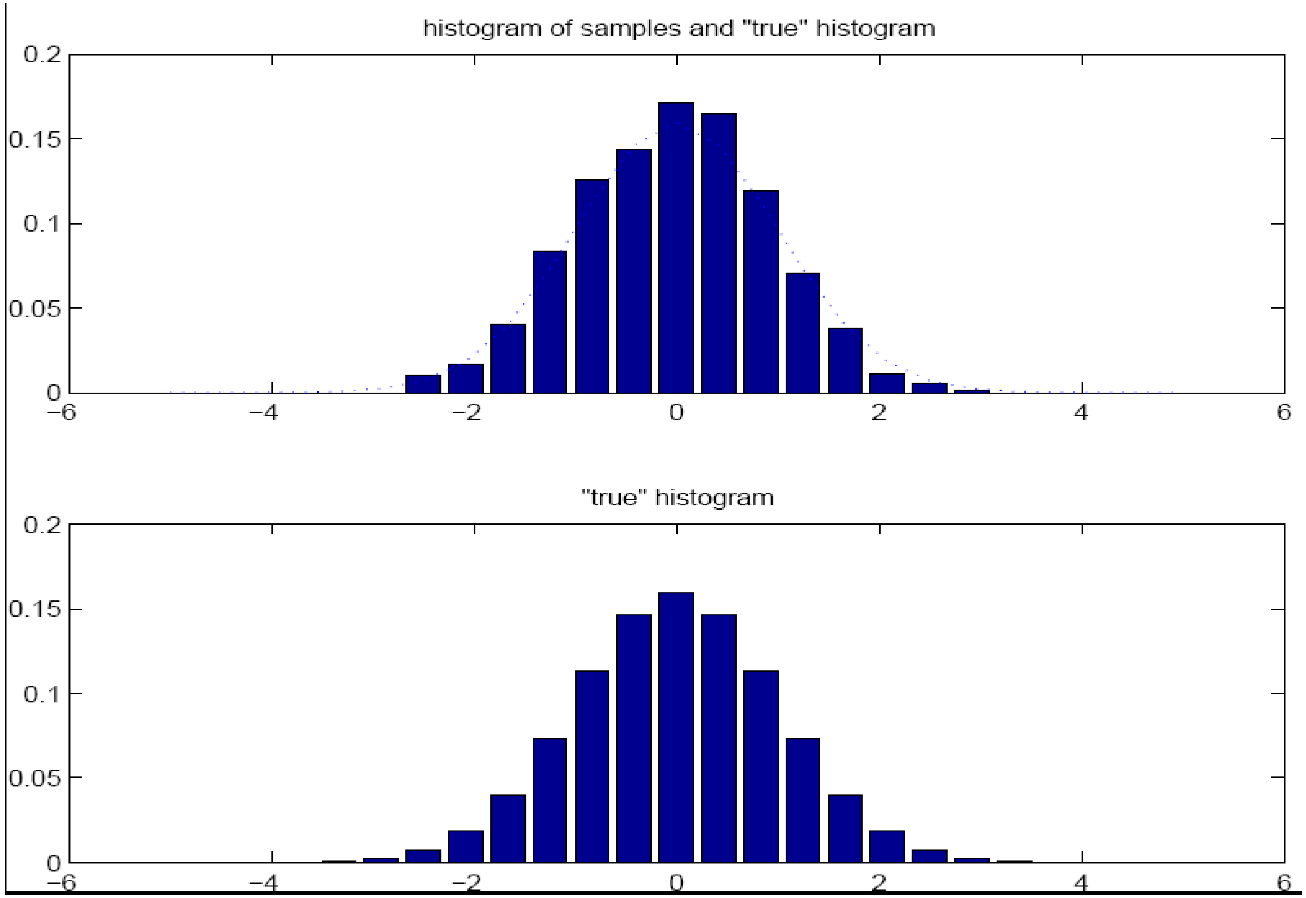
02457 Non-linear signal processing

2017 - Lecture 13 review

Outline review lecture 13

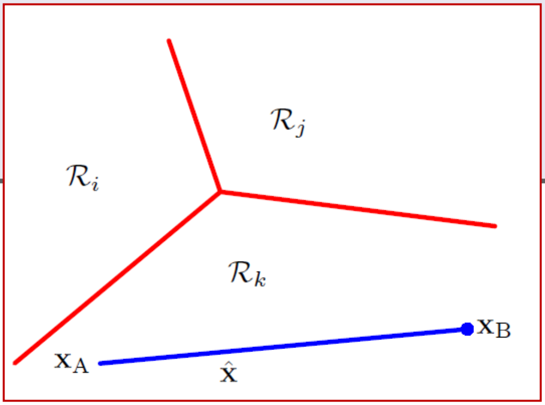
- Learning and generalization
- Bayesian linear model
- Neural networks
- Dynamic linear model – forward algorithm
  Teacher-student, sun spots
- Exam & goodbye

Models vs data



$$\mu = \int_{-\infty}^{\infty} x \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) dx \qquad \mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

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

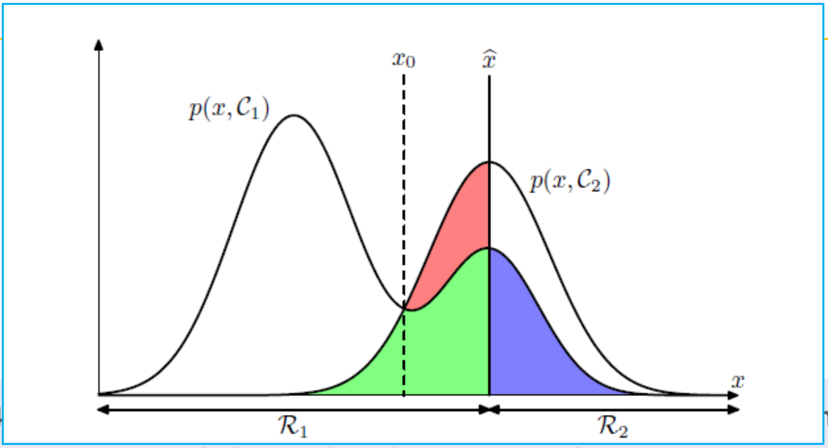
$$P(\mathcal{C}_k, \mathbf{x}) = p(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k)$$
$$P(\mathcal{C}_k, \mathbf{x}) = P(\mathcal{C}_k|\mathbf{x})p(\mathbf{x})$$

$$\boxed{P(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k)}{p(\mathbf{x})}}$$

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{P(\mathcal{C}_k|X^l)p(\mathbf{x})}{P(\mathcal{C}_k)}$$

$$\sum_{k=1}^{c} P(\mathcal{C}_k|\mathbf{x}) = 1$$

$$\sum_{k=1}^{c} p(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k) = p(\mathbf{x})$$



Signal detection / Classification / Signals and noise

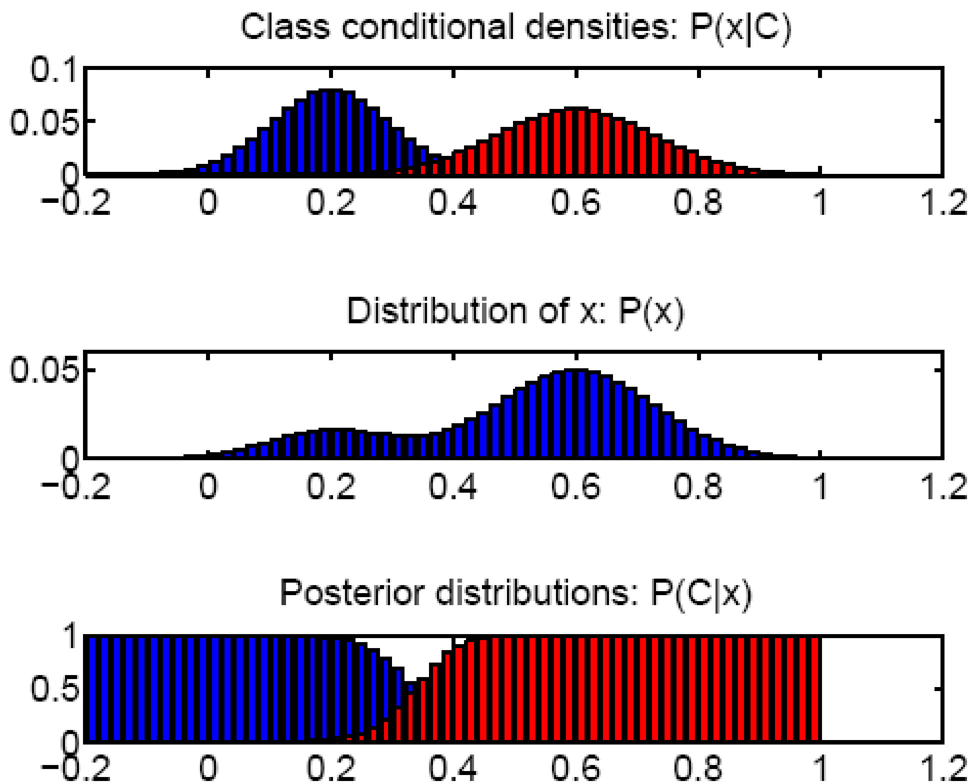


Figure 1: Schematic plot of the histograms for a measured signal drawn from either of two populations $\mathcal{C}_1, \mathcal{C}_2$, density of x, and the corresponding posteriors $P(\mathcal{C}|X)$'s

Learning from data

- Supervised learning: Learning relations between sets of variables e.g. between input and output variables, conditional distributions $p(\mathrm{output}|\mathrm{input})$.
- Unsupervised learning: Learning the distribution of a set of variables $p(\mathrm{input})$.

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 $\chi = \{(t^1, \mathbf{x}^1), (t^2, \mathbf{x}^2), ..., (t^N, \mathbf{x}^N)\}$ be a *training set*
- Objective: Find the distribution of the parameter vector, $p(\mathbf{w}|\chi)$, hence the parameters are considered stochastic.

The likelihood function

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

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

- The function $p(\chi|\mathbf{w})$ is called the likelihood function (more correct the likelihood of the parameter vector θ). The density $p(\mathbf{w})$ is called the a $priori$ or $prior$ parameter distribution.
- If the prior is "flat" in the neighborhood of the peak of $p(\chi|\mathbf{w})$, we have

Maximum likelihood & optimization

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

$$p(\chi|\mathbf{w}) = \prod_{n=1}^{N} p(t^n|\mathbf{x}_n, \mathbf{w})p(\mathbf{x}^n) = p(\chi_t|\chi_{\mathbf{x}}, \mathbf{w}) * p(\chi_{\mathbf{x}})$$

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

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

Bayesian linear learning

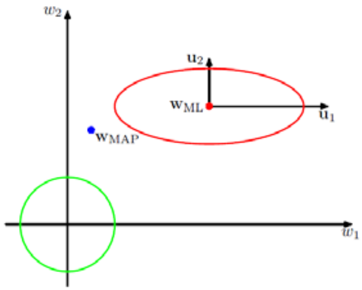


Figure 3.15 Contours of the likelihood function (red) and the prior (green) in which the axes in parameter space have been rotated to align with the eigenvectors $\mathbf{u}_i$ of the Hessian. For $\alpha = 0$, the mode of the posterior is given by the maximum likelihood solution $\mathbf{w}_{\mathrm{ML}}$, whereas for nonzero α the mode is at $\mathbf{w}_{\mathrm{MAP}} = \mathbf{m}_N$. In the direction w_1 the eigenvalue λ_1, defined by (3.87), is small compared with α and so the quantity $\lambda_1/(\lambda_1 + \alpha)$ is close to zero, and the corresponding MAP value of w_1 is also close to zero. By contrast, in the direction w_2 the eigenvalue λ_2 is large compared with α and so the quantity $\lambda_2/(\lambda_2+\alpha)$ is close to unity, and the MAP value of w_2 is close to its maximum likelihood value.

Let $\mathcal{D} = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), ..., (\mathbf{x}_N, t_N)\}$ be a data set of N samples with $\mathbf{x} \in \mathbb{R}^d$.

$$\boxed{t = \mathbf{w}^\top \mathbf{x} + \epsilon} \qquad p(\mathcal{D}|\mathbf{w}, \beta) = \left(\sqrt{\frac{\beta}{2\pi}}\right)^N \exp\left(-\frac{\beta}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^\top \mathbf{x}_n)^2\right)$$

assign a standard Gaussian prior to the weights $\mathbf{w} \sim \mathcal{N}(0, \alpha^{-1}\mathbf{I})$

$$\begin{aligned} p(\mathbf{w}|\alpha, \beta, \mathcal{D}) &= \frac{p(\mathcal{D}|\mathbf{w}, \beta)p(\mathbf{w}|\alpha)}{p(\mathcal{D}|\alpha, \beta)} \\ &\propto \left(\sqrt{\frac{\beta}{2\pi}}\right)^N \exp\left(-\frac{\beta}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^\top \mathbf{x}_n)^2\right) \left(\sqrt{\frac{\alpha}{2\pi}}\right)^d \exp\left(-\frac{\alpha}{2}||\mathbf{w}||^2\right) \end{aligned}$$

The product between $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, is proportional to $\mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ with mean vector and covariance matrix given by,

$$\begin{aligned} \boldsymbol{\mu}_p &= \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1} \left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right) \\ \boldsymbol{\Sigma}_p &= \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}. \end{aligned}$$

Bayesian linear learning

The product between $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, is proportional to $\mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ with mean vector and covariance matrix given by,

$$\begin{aligned}
\boldsymbol{\mu}_p &= \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1} \left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right) \\
\boldsymbol{\Sigma}_p &= \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}.
\end{aligned}$$

$$\begin{aligned}
p(\mathbf{w}|\alpha, \beta, \mathcal{D}) &= \frac{p(\mathcal{D}|\mathbf{w}, \beta) p(\mathbf{w}|\alpha)}{p(\mathcal{D}|\alpha, \beta)} \\
&\propto \left(\sqrt{\frac{\beta}{2\pi}}\right)^N \exp\left(-\frac{\beta}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^\top \mathbf{x}_n)^2\right) \left(\sqrt{\frac{\alpha}{2\pi}}\right)^d \exp\left(-\frac{\alpha}{2}||\mathbf{w}||^2\right)
\end{aligned}$$

In this case the prior is given by $\boldsymbol{\mu}_2 \equiv \boldsymbol{\mu}_{\text{prior}} = \mathbf{0}$ and $\boldsymbol{\Sigma}_2 \equiv \boldsymbol{\Sigma}_{\text{prior}} = \alpha^{-1}\mathbf{I}$. For the likelihood a bit of algebra leads to,

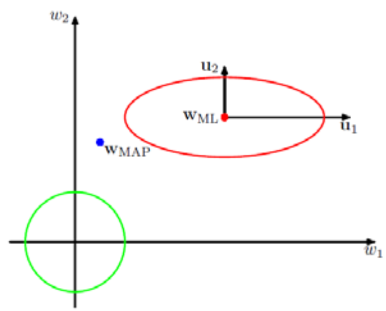
$$\begin{aligned}
\boldsymbol{\mu}_1 &\equiv \left(\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top\right)^{-1} \sum_{n=1}^{N} \mathbf{x}_n t_n \\
\boldsymbol{\Sigma}_1 &\equiv \left(\beta \sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top\right)^{-1}. \qquad (3)
\end{aligned}$$

Bayesian linear learning

Hence the posterior mean vector and covariance matrix are found as

$$
\begin{aligned}
\boldsymbol{\mu}_p &\equiv \left(\alpha \mathbf{I} + \beta \sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top \right)^{-1} \beta \sum_{n=1}^{N} \mathbf{x}_n t_n \\
\boldsymbol{\Sigma}_p &\equiv \left(\alpha \mathbf{I} + \beta \sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top \right)^{-1} .
\end{aligned}
$$

Figure 3.15 Contours of the likelihood function (red) and the prior (green) in which the axes in parameter space have been rotated to align with the eigenvectors $\mathbf{u}_i$ of the Hessian. For $\alpha = 0$, the mode of the posterior is given by the maximum likelihood solution $\mathbf{w}_{\mathrm{ML}}$, whereas for nonzero α the mode is at $\mathbf{w}_{\mathrm{MAP}} = \mathbf{m}_N$. In the direction w_1 the eigenvalue λ_1, defined by (3.87), is small compared with α and so the quantity $\lambda_1/(\lambda_1 + \alpha)$ is close to zero, and the corresponding MAP value of w_1 is also close to zero. By contrast, in the direction w_2 the eigenvalue λ_2 is large compared with α and so the quantity $\lambda_2/(\lambda_2+\alpha)$ is close to unity, and the MAP value of w_2 is close to its maximum likelihood value.



The predictive density is computed as

$$
p(t_{N+1}|\mathbf{x}_{N+1}, \mathcal{D}) = \int p(t_{N+1}|\mathbf{x}_{N+1}, \mathbf{w}) p(\mathbf{w}|\mathcal{D}) d\mathbf{w}
$$

This is again a normal distribution. We note

$$
t_{N+1} = \mathbf{w}_N^\top \mathbf{x}_{N+1} + \epsilon_{N+1}, \quad \text{and} \quad \mathbf{w}_N \sim \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p),
$$

which leads to the predictive mean and variance,

$$
\begin{aligned}
\mu_{t_{N+1}} &= \boldsymbol{\mu}_p^\top \mathbf{x}_{N+1}, \\
\sigma^2_{t_{N+1}} &= \beta^{-1} + \mathbf{x}_{N+1}^\top \boldsymbol{\Sigma}_p \mathbf{x}_{N+1} .
\end{aligned}
$$

The generalization error: "The Hidden agenda"

- 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*.

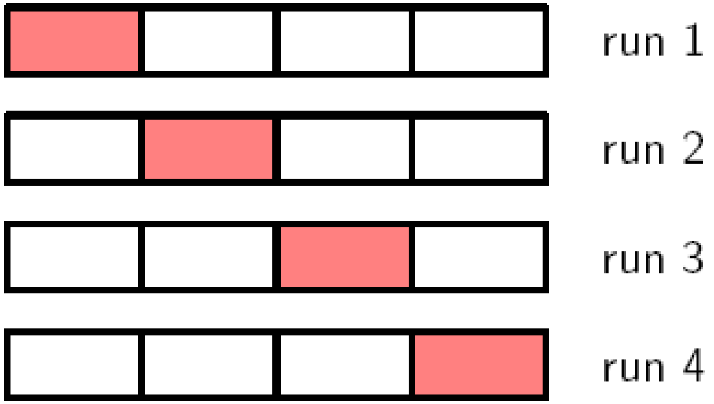
- So, let us define *the generalization error*:

$$E = \lim_{M\to\infty} \frac{1}{M}\sum_{k=1}^{M} -\log p(t^k|\mathbf{x}^k, \mathbf{w})$$
$$= \int\int -\log[p(t^k|\mathbf{x}^k, \mathbf{w})]p(t|\mathbf{x})dt p(\mathbf{x})d\mathbf{x}$$

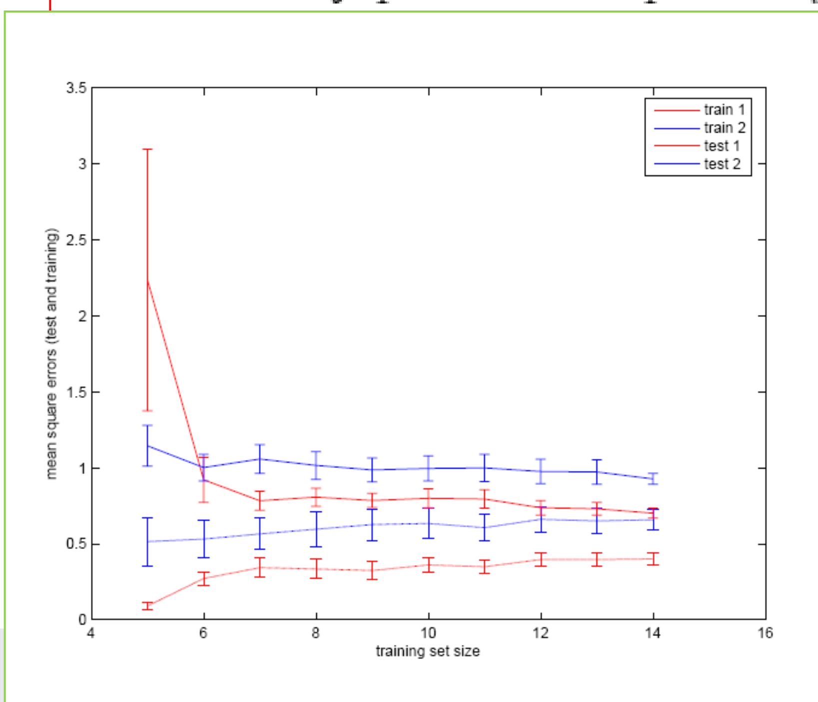
  This is the average (or expected) error on a test datum $(t, \mathbf{x})$.

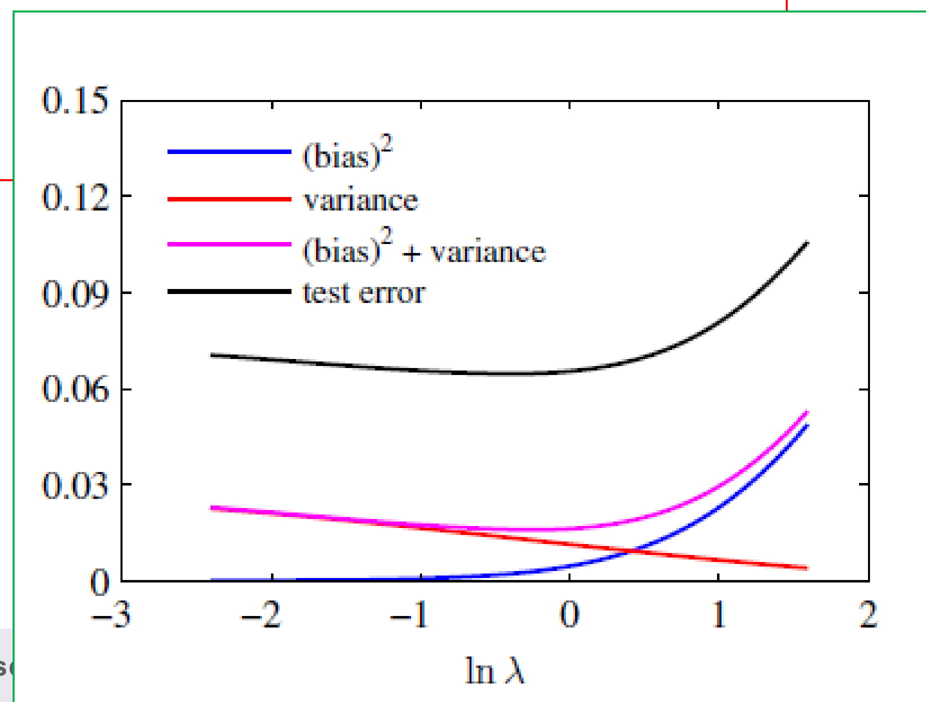
Generalization errors can not be measured, but can be estimated using a finite *test set*

The bias-variance trade-off quantities can be estimated by drawing multiple training sets (can in fact be overlapping i.e. cross-validation)



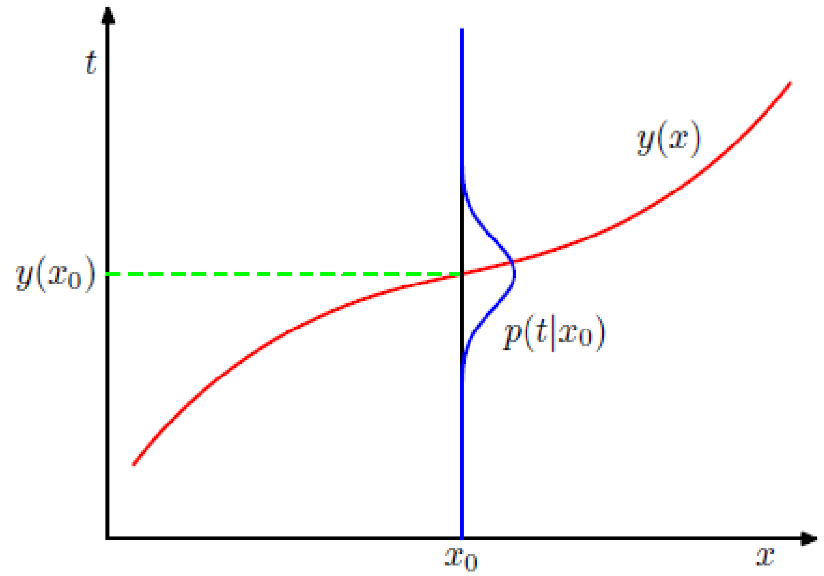
- The Generalization error depends on the interplay between model flexibility and training set size
- The learning curve is the relation between generalization and training set size: $E_{\text{test}}(N)$ vs. N.
- The generalization error is determined by the complexity of the model and the amount of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





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

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

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

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

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

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 $\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]$

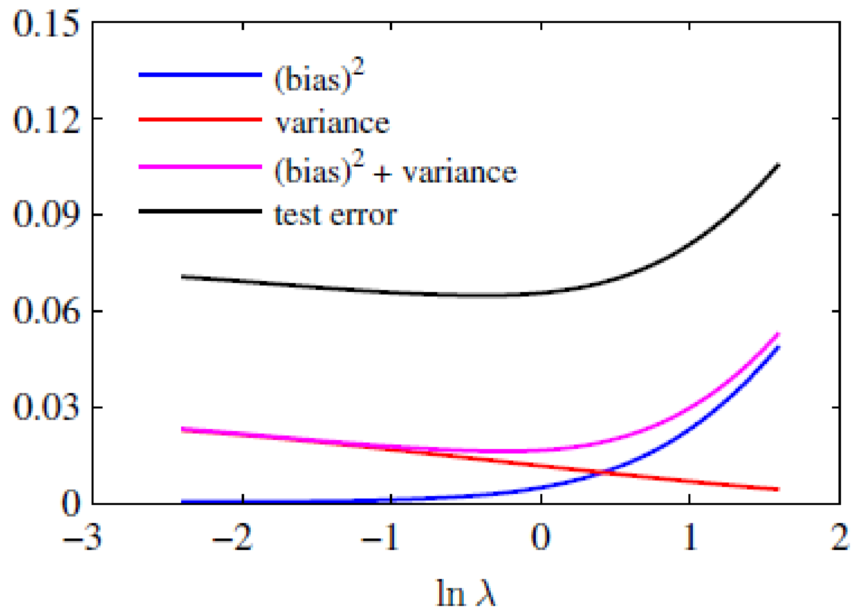
$$\begin{aligned}
&\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\
&= \{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};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

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

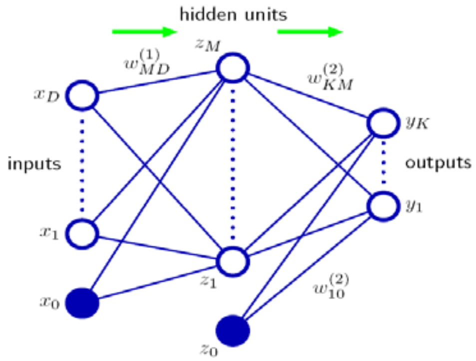
$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2\right] \\
&= \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};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

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.

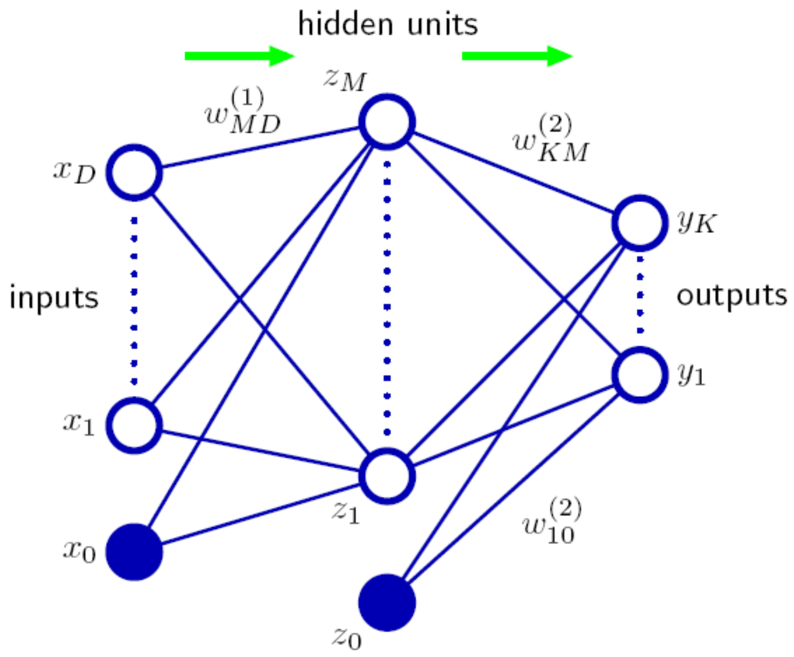


Neural network function approximation



We seek a conditional density model of the form

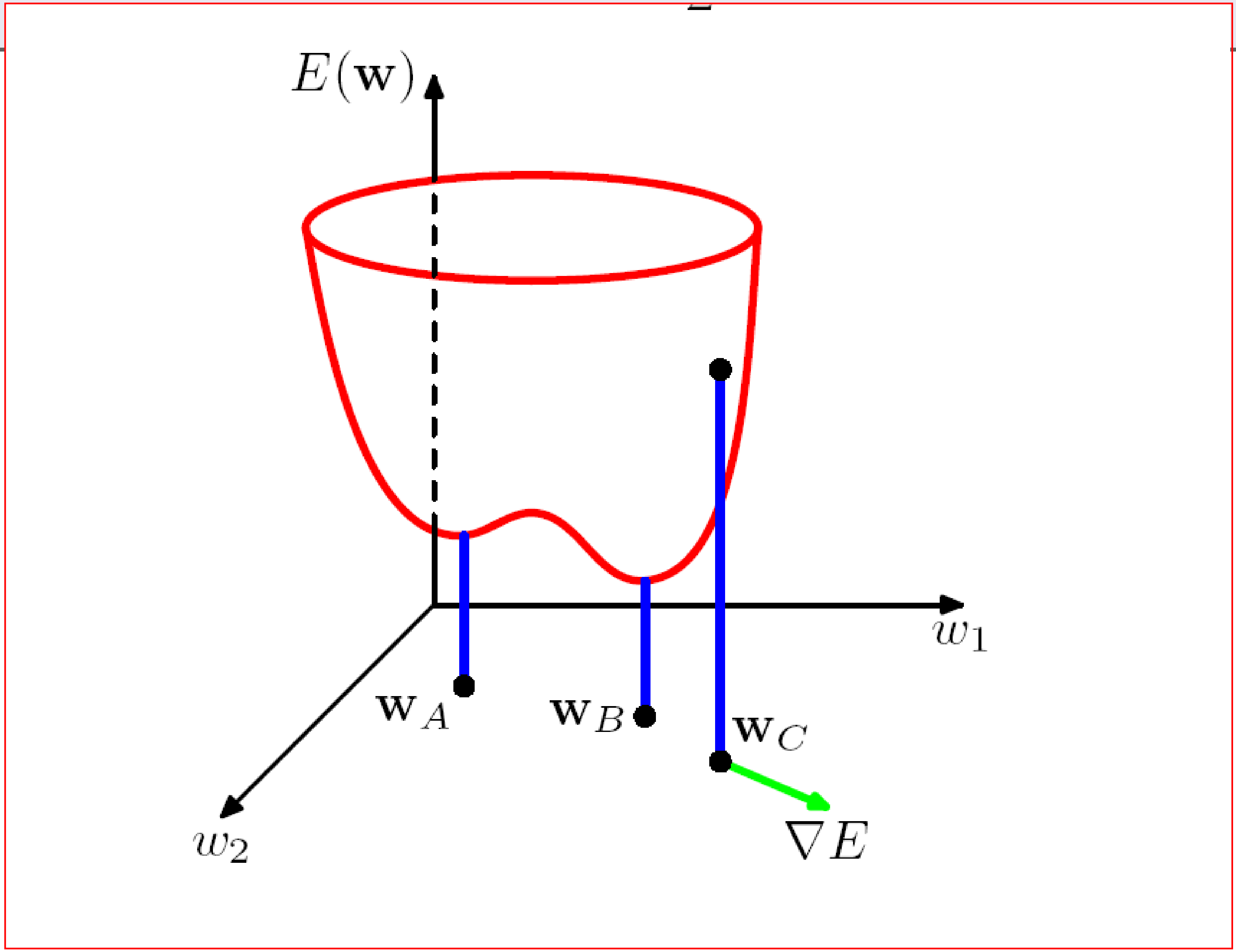
$$
\begin{aligned}
t &= y(\mathbf{x}, \mathbf{w}) + \nu \\
p(t|\mathbf{x}, \sigma^2, \mathbf{w}) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(t - y(\mathbf{x}, \mathbf{w}))^2\right) \\
p(\chi_t|\chi_{\mathbf{x}}, \sigma^2, \theta) &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(t^n - y(\mathbf{x}^n, \mathbf{w}))^2\right) \\
E(\mathbf{w}, \sigma^2) &= \frac{N}{2}\log 2\pi\sigma^2 + \frac{1}{2\sigma^2}\sum_{n=1}^{N}(t^n - y(\mathbf{x}^n, \mathbf{w}))^2
\end{aligned}
$$



$$
\begin{aligned}
y(\mathbf{x},\mathbf{w}) &= \tanh\left(\sum_{j=0}^{n_H} W_j z_j(\mathbf{x},\mathbf{w})\right) \\
z_j(\mathbf{x},\mathbf{w}) &= \tanh\left(\mathbf{w}_j^\top \mathbf{x} + w_0\right) \\
z_0 &= 1
\end{aligned}
$$

The set of linear output MLP's is dense in the continuous functions on a compact subset of a vector space: If given an ϵ and a continuous target function $f_{\mathrm{target}}(\mathbf{x})$ on the set Ω, we can find an MLP network for which

$$|y(\mathbf{x}, \mathbf{w}) - f_{\mathrm{target}}(\mathbf{x})| < \epsilon, \forall \mathbf{x} \in \Omega$$


$E(\mathbf{w})$
w_1
w_2
$\mathbf{w}_A$
$\mathbf{w}_B$
$\mathbf{w}_C$
∇E

- Objective: to solve the equation $\nabla E = 0$

$$\begin{aligned} \mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} + \Delta \mathbf{w}^{(\tau)} \\ \Delta \mathbf{w}^{(\tau)} &= -\eta \nabla E\big|_{\mathbf{w}^{(\tau)}} \end{aligned}$$

- η is the learning parameter
- η can be too small: convergence very slow
- η can be too large: oscillatory behavior

Linear output network y = w’ · z

$$
\begin{aligned}
E &= \sum_{n=1}^{N} (y(\mathbf{x}^n, \mathbf{w}) - t^n)^2 \\
\frac{\partial E}{\partial u} &= \sum_{n=1}^{N} \frac{\partial}{\partial u} (y(\mathbf{x}^n, \mathbf{w}) - t^n)^2 \\
&= 2 \sum_{n=1}^{N} (y(\mathbf{x}^n, \mathbf{w}) - t^n) \frac{\partial y(\mathbf{x}^n, \mathbf{w})}{\partial u}
\end{aligned}
$$

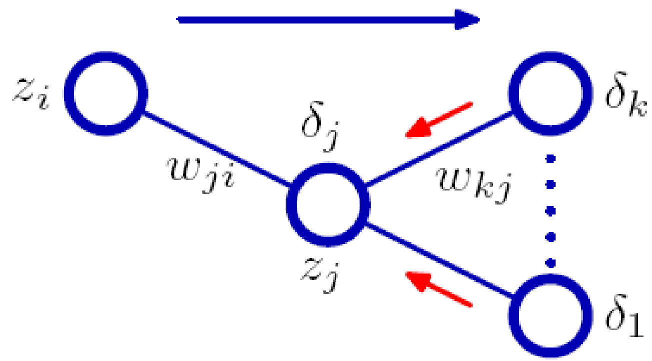
$$
\begin{aligned}
\frac{\partial y(\mathbf{x}^n, \mathbf{w})}{\partial w_{j'}} &= \frac{\partial}{\partial w_{j'}} \sum_{j=0}^{n_H} \mathbf{w}_j z_j(\mathbf{x}) \\
&= \sum_{j=0}^{n_H} \frac{\partial}{\partial w_{j'}} w_j z_j(\mathbf{x}) \\
&= z_{j'}(\mathbf{x})
\end{aligned}
$$

$$
\begin{aligned}
E &= \sum_{n=1}^{N} (y(\mathbf{x}^n, \mathbf{w}) - t^n)^2 \\
\frac{\partial E}{\partial u} &= \sum_{n=1}^{N} \frac{\partial}{\partial u} (y(\mathbf{x}^n, \mathbf{w}) - t^n)^2 \\
&= 2 \sum_{n=1}^{N} (y(\mathbf{x}^n, \mathbf{w}) - t^n) \frac{\partial y(\mathbf{x}^n, \mathbf{w})}{\partial u}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial y(\mathbf{x}^n, \mathbf{w})}{\partial w_{j',k'}} &= \frac{\partial}{\partial w_{j',k'}} \sum_{j=0}^{n_H} w_j z_j(\mathbf{x}) \\
&= \sum_{j=0}^{n_H} w_j \frac{\partial}{\partial w_{j',k'}} z_j(\mathbf{x}) \\
&= w_{j'} \frac{\partial}{\partial w_{j',k'}} \tanh \left(\sum_{k=0}^{n_I} w_{j,k} x_k^n \right) \\
&= w_{j'} \left(1 - \tanh^2 \left(\sum_{k=0}^{n_I} w_{j,k} x_k^n \right) \right) \frac{\partial}{\partial w_{j',k'}} \sum_{k=0}^{n_I} w_{j,k} x_k^n \\
&= w_{j'} \left(1 - z_{j'}^2 \right) x_{k'}^n
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial E}{\partial w_j} &= 2\sum_{n=1}^{N}(y(\mathbf{x}^n) - t^n) z_j(\mathbf{x}^n) \\
&\equiv 2\sum_{n=1}^{N} \delta^n z_j(\mathbf{x}^n)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial E}{\partial w_{j,k}} &= 2\sum_{n=1}^{N}(y(\mathbf{x}^n) - t^n) w_j \left(1 - z_j^2(\mathbf{x}^n)\right) x_k^n \\
&\equiv 2\sum_{n=1}^{N} \delta_j^n x_k^n
\end{aligned}
$$



$$\frac{\partial E}{\partial w_{ji}} = \sum_n \delta_j^n z_i(\mathbf{x}^n)$$

$$\delta_j^n \equiv \frac{\partial E^n}{\partial a_j} = \sum_k \frac{\partial E^n}{\partial a_k}\frac{\partial a_k}{\partial a_j}$$

$$\delta_j^n = g'(a_j^n)\sum_k w_{kj}\delta_k^n$$

Find η by line search along the search direction $\mathbf{d}^{(\tau)} = -\nabla E|_{\mathbf{w}^{(\tau)}}$:

$$\begin{aligned} \mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} + \eta^{(\tau)}\mathbf{d}^{(\tau)} \\ E(\eta) &= E(\mathbf{w}^{(\tau)} + \eta\mathbf{d}^{(\tau)}) \end{aligned}$$

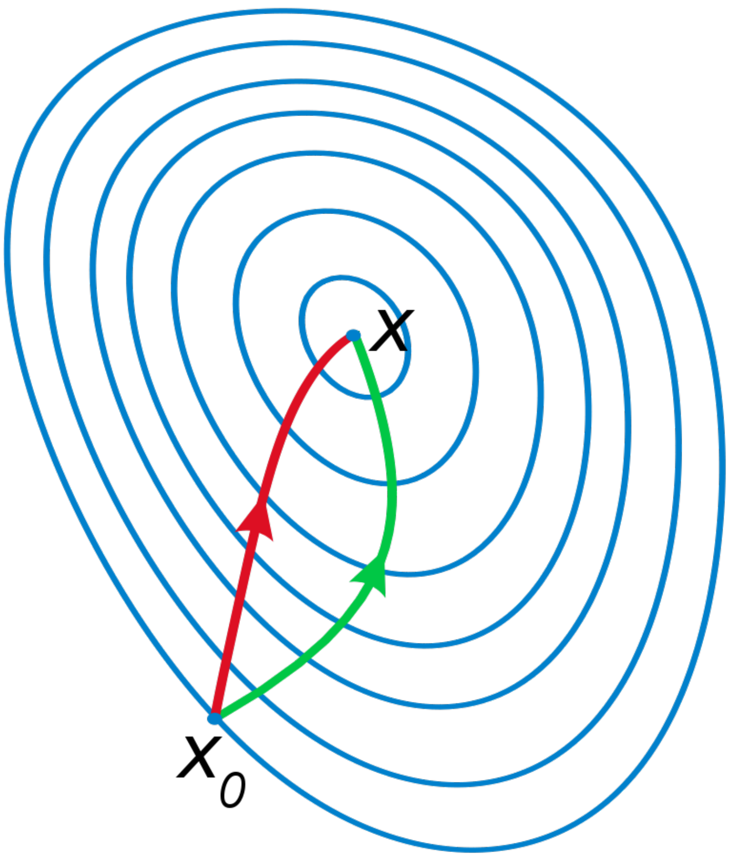
Newton method

$$E(w) = E(w^*) + \frac{1}{2}H(w - w^*)^2$$

$$\frac{\partial E}{\partial w}(w) = \frac{\partial E}{\partial w}(w^*) + H(w - w^*)$$

$$\frac{\partial E}{\partial w}(w) = H(w - w^*)$$

$$\boxed{w^* = w - H^{-1}\frac{\partial E}{\partial w}(w)}$$



Gradient orthogonal to contour

Hessian calculation

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

$$\frac{\partial E}{\partial \mathbf{w}} = \sum_{n=1}^{N}(y^n - d^n)\frac{\partial y^n}{\partial \mathbf{w}}$$

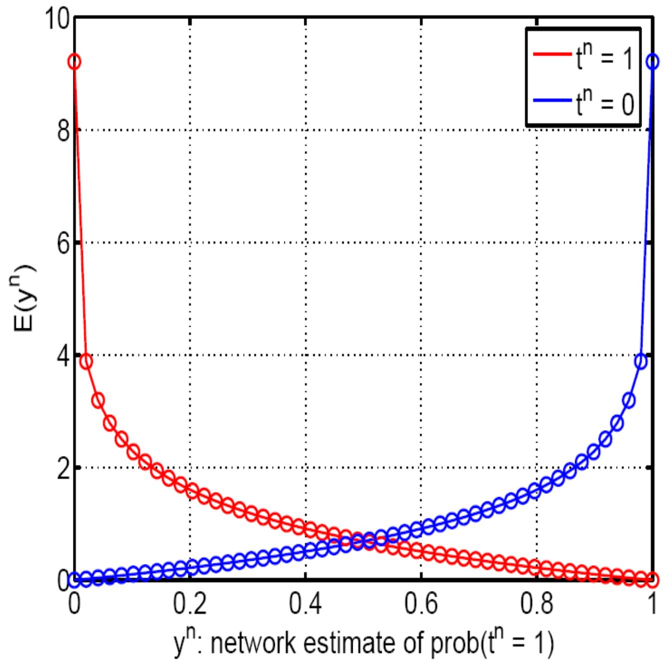
$$\frac{\partial^2 E}{\partial \mathbf{w}\partial \mathbf{w}^\top} = \sum_{n=1}^{N}\frac{\partial y^n}{\partial \mathbf{w}}\frac{\partial y^n}{\partial \mathbf{w}}^\top + \sum_{n=1}^{N}(y^n - d^n)\frac{\partial^2 y^n}{\partial \mathbf{w}\partial \mathbf{w}^\top}$$

$$\boxed{\frac{\partial^2 E}{\partial \mathbf{w}\partial \mathbf{w}^\top} \approx \sum_{n=1}^{N}\frac{\partial y^n}{\partial \mathbf{w}}\frac{\partial y^n}{\partial \mathbf{w}}^\top}$$

Likelihood for classification network

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

$$\begin{aligned} p(\chi_t|\chi_x, \mathbf{w}) &= \prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w}) \\ &= \prod_{n=1}^{N} y(\mathbf{x}_n|\mathbf{w})^{t_n}[1 - y(\mathbf{x}_n|\mathbf{w})]^{(1-t_n)} \end{aligned}$$



$$\boxed{E(\mathbf{w}) = -\sum_{n=1} t_n \log y(\mathbf{x}_n|\mathbf{w}) + (1-t_n)\log[1 - y(\mathbf{x}_n|\mathbf{w})]}$$

Softmax for multi-label problems

- We use $0 \leq y \leq 1$ coding for C classes and we want the outputs to be the posterior probabilities $P(C|\mathbf{x})$, hence they "should sum to one"

$$y_k(\mathbf{x}) = \frac{\exp a_k(\mathbf{x})}{\sum_{k'} \exp a_{k'}(\mathbf{x})}$$

- Targets are represented by 0-1 vectors:

$$\mathbf{t}_k = [0, 0, 0, ..., 1, 0, 0]$$

- The likelihood function is given by

$$p(\mathbf{t}|\mathbf{x}) = \prod_{k=1}^{C} y_k(\mathbf{x})^{t_k}$$

Pruning networks for improved generalizability

$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{\partial E}{\partial \mathbf{w}}(\mathbf{w}-\mathbf{w}^*) + \frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^T \mathbf{H} (\mathbf{w}-\mathbf{w}^*)$$

$$\mathbf{w} - \mathbf{w}^* = w_j \mathbf{e}_j$$

$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{\partial E}{\partial \mathbf{w}} w_j \mathbf{e}_j + \frac{1}{2} w_j \mathbf{e}_j^T \mathbf{H} w_j \mathbf{e}_j$$

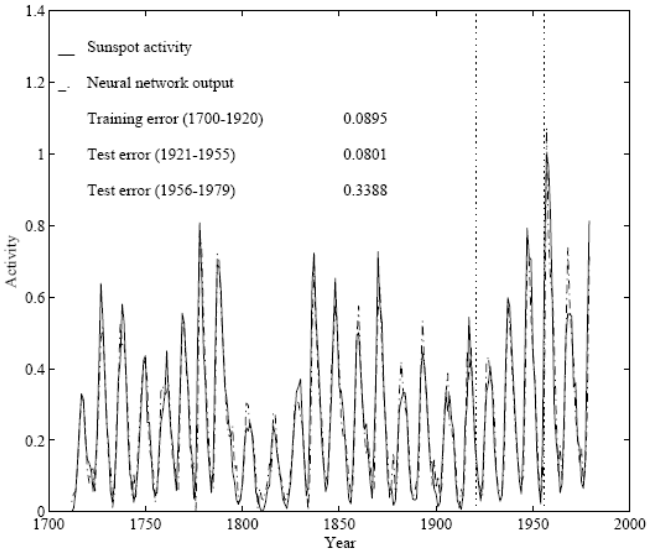
$$\Delta E(\mathbf{w})_{\mathrm{obd}} \approx \frac{1}{2}\mathbf{H}_{jj} w_j^2$$

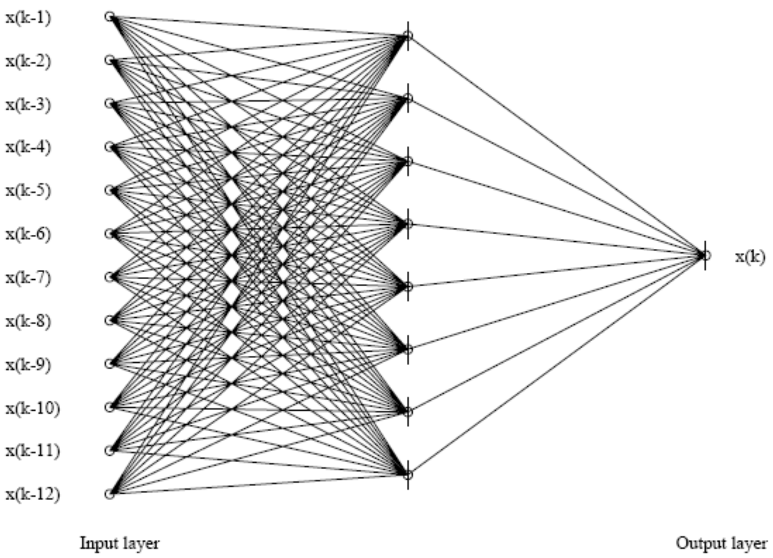
$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{\partial E}{\partial w_j} w_j + \frac{1}{2}\mathbf{H}_{j,j} w_j^2$$

$$\Delta E(\mathbf{w})_{\mathrm{obs}} \approx \frac{1}{2}\frac{w_j^2}{(\mathbf{H}^{-1})_{jj}}$$

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





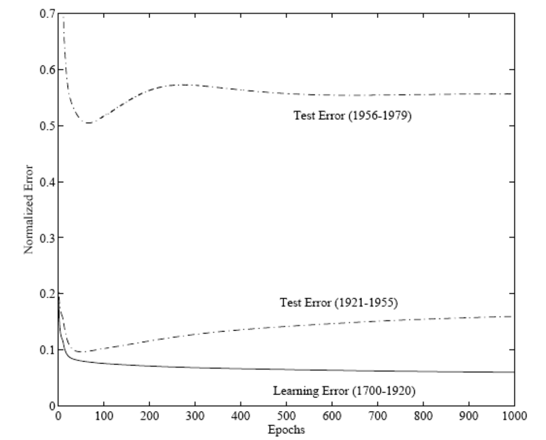
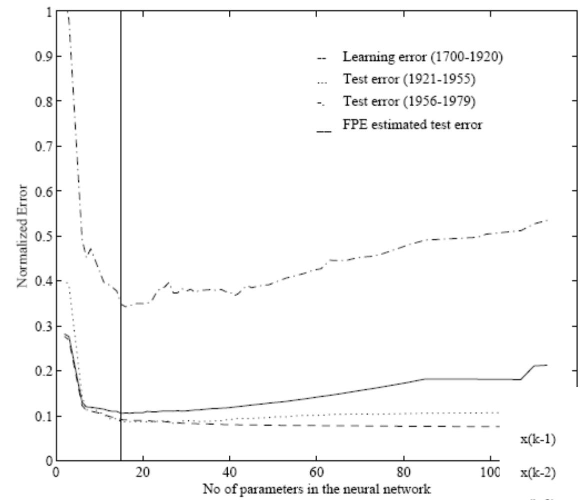
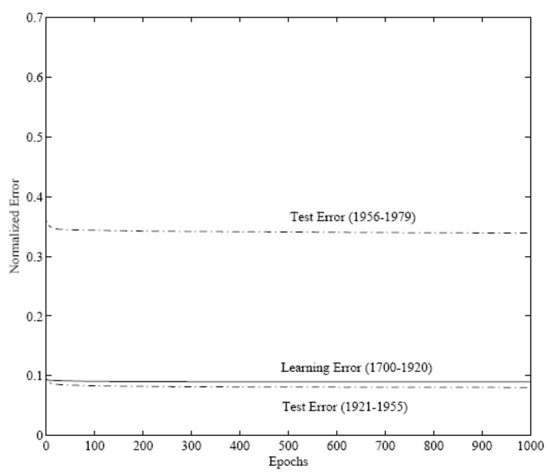
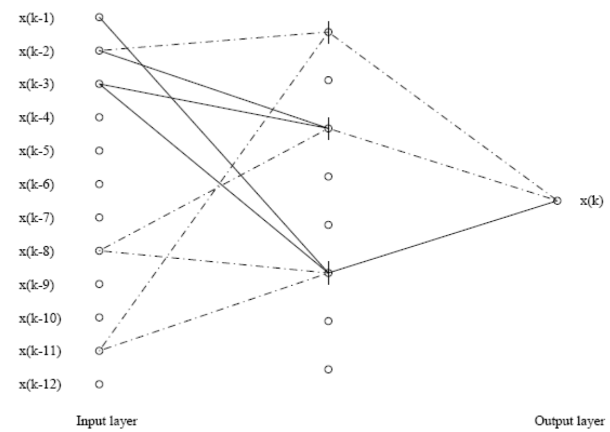


Fig. 2 Training and test error when training the fully connected network. An 'Epoch' is a full sweep through the training set.



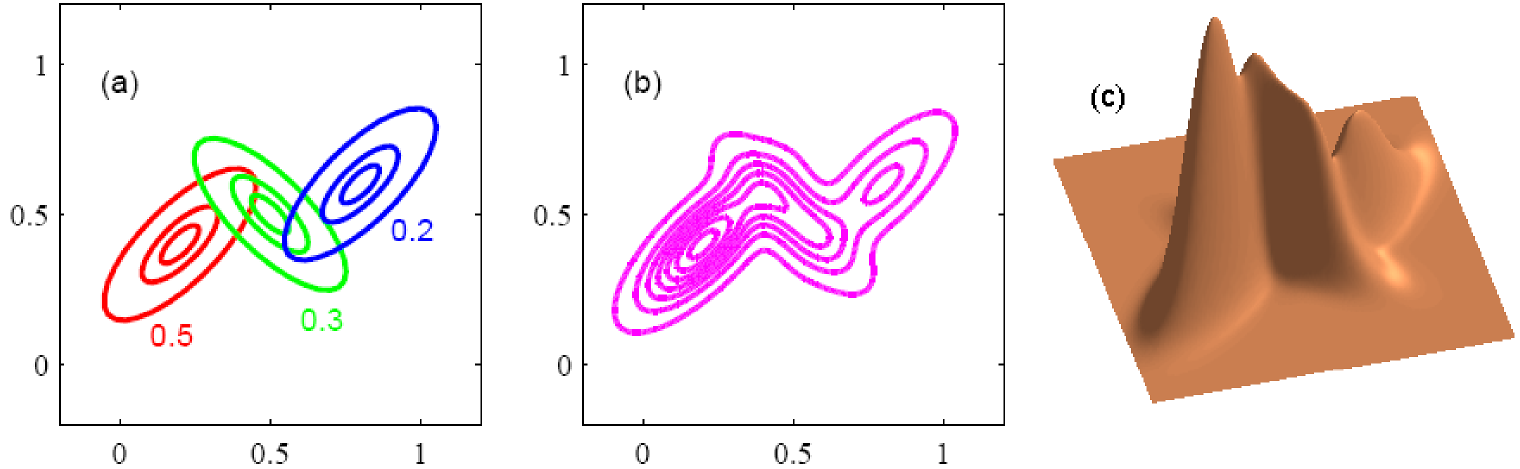




Model	Train (1700-1920)	Test (1921-55)	Test (1956-79)	Number of parameters
Tong and Lim [10]	0.097	0.097	0.28	16
Weigend *et al.* [11]	0.082	0.086	0.35	43
Linear model[1]	0.132	0.130	0.37	13
Fully connected network[2]	0.078 ± 0.002	0.104 ± 0.005	0.46 ± 0.07	113
Pruned network[3]	0.090 ± 0.001	0.082 ±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

Mixture of Gaussians

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \ , \qquad \sum_k \pi_k = 1$$



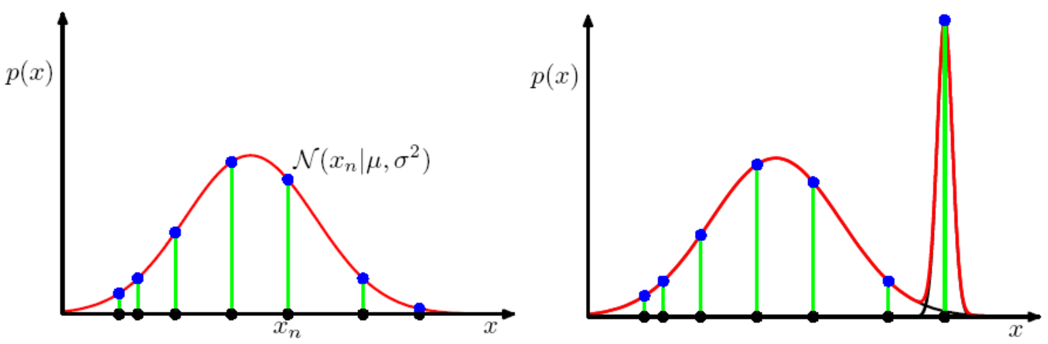
- How to obtain samples from the mixture?
  First you draw a “number” k in the interval [1,…,K] with probability π_k
  Next draw a vector from the k’th Gaussian distribution

Likelihood function for a mixture of Gaussians

- The cost function is (notice sum inside log)

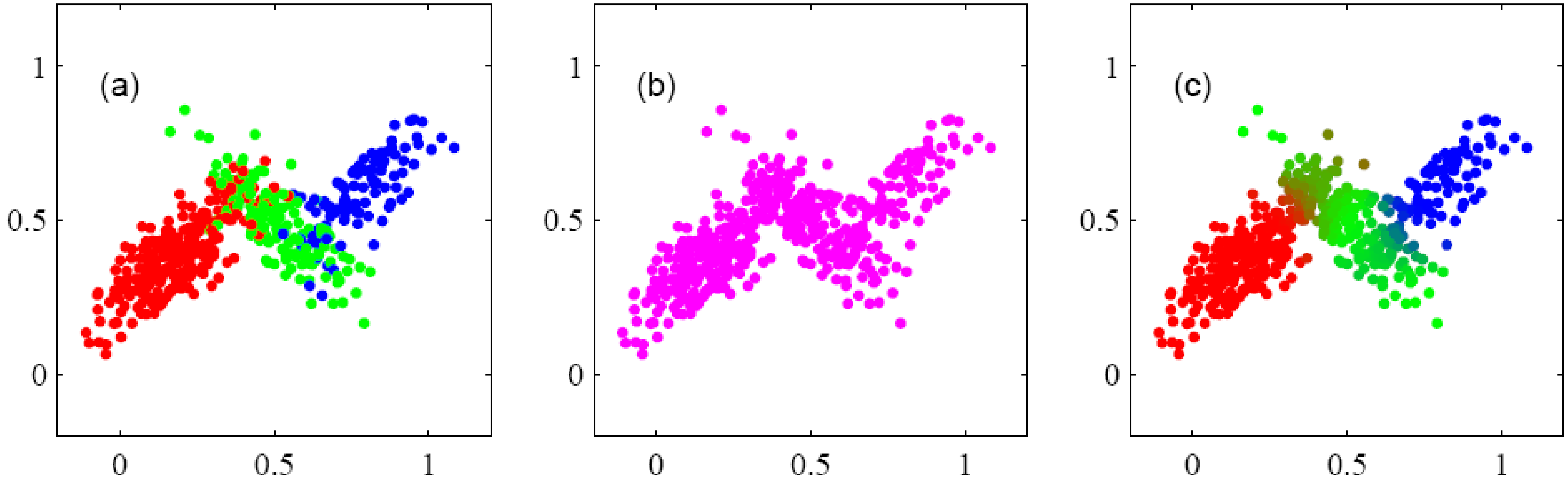
$$E(\mathbf{w}) = -\sum_{n=1}^{N} \log p(\mathbf{x}_n|\mathbf{w}) = -\sum_{n=1}^{N} \log \sum_{k=1}^{K} p(\mathbf{x}_n|\mathbf{w}_k)\pi_k$$

$$= -\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



Key idea: Introduce the posterior assignment probabilities: The "responsibilities"

$$
\begin{aligned}
\gamma_{nk} &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{k'}, \Sigma_{k'})} \\
&= \frac{p(k)p(\mathbf{x}_n|k)}{\sum_{k'} p(k')p(\mathbf{x}_n|k')} = p(k|\mathbf{x}_n) \in [0, 1] \ .
\end{aligned}
$$



Simplification with Jensen's inequality

- We bound the change in cost function:

$$\begin{aligned} E^{\text{new}}(\mathbf{w}) &= -\sum_{n=1}^{N} \log p^{\text{new}}(\mathbf{x}_n|\mathbf{w}) \\ &= -\sum_{n=1}^{N} \log \sum_{k=1}^{K} p^{\text{new}}(\mathbf{x}_n|k)\pi_k^{\text{new}} \frac{\gamma_{nk}^{\text{old}}}{\gamma_{nk}^{\text{old}}} \\ &\leq -\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk}^{\text{old}} \log \frac{p^{\text{new}}(\mathbf{x}_n|k)\pi_k^{\text{new}}}{\gamma_{nk}^{\text{old}}} \end{aligned}$$

Jensen's inequality

$$\log\left(\sum_j \lambda_j x_j\right) \geq \sum_j \lambda_j \log(x_j) \qquad \sum_j \lambda_j = 1$$

Expectation maximization

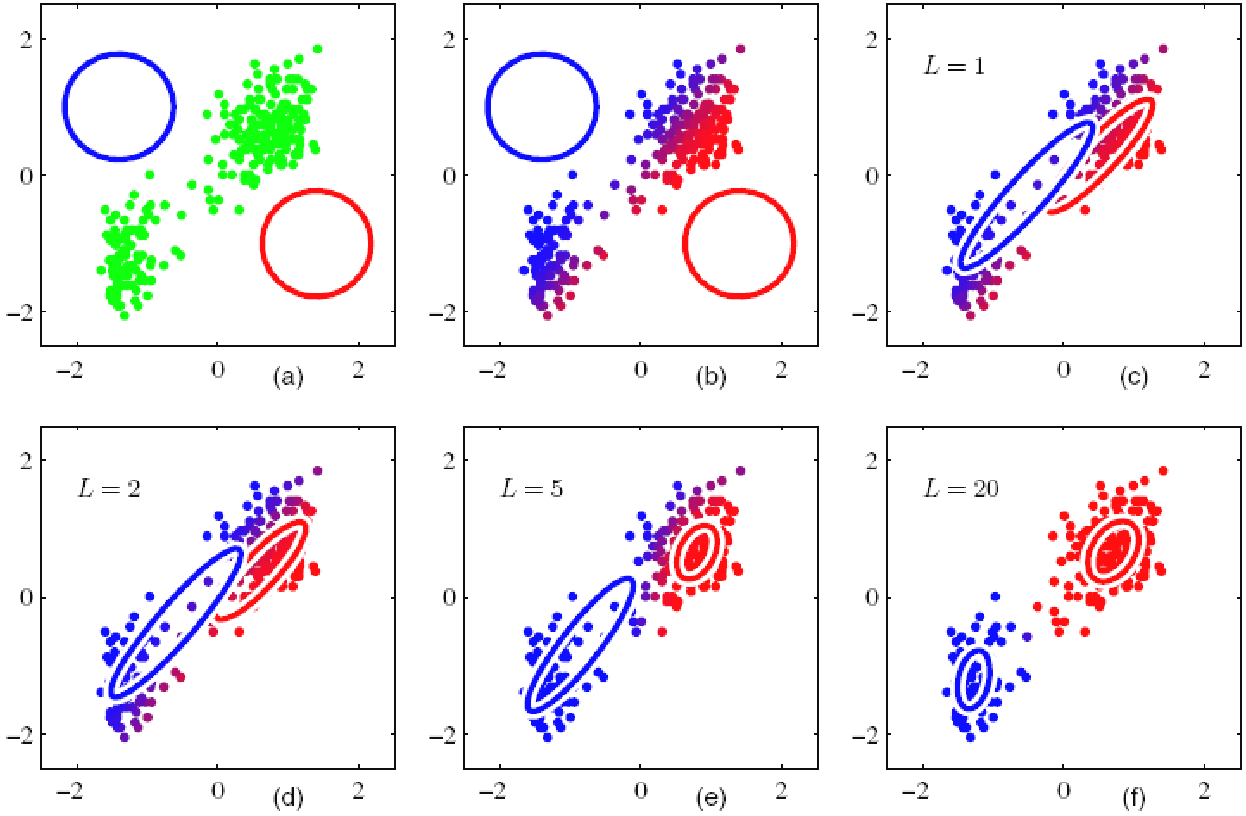
M-step – minimizing the bound gives:

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma_{nk}^{\text{old}} \mathbf{x}_n}{\sum_{n=1}^{N} \gamma_{nk}^{\text{old}}}$$

and

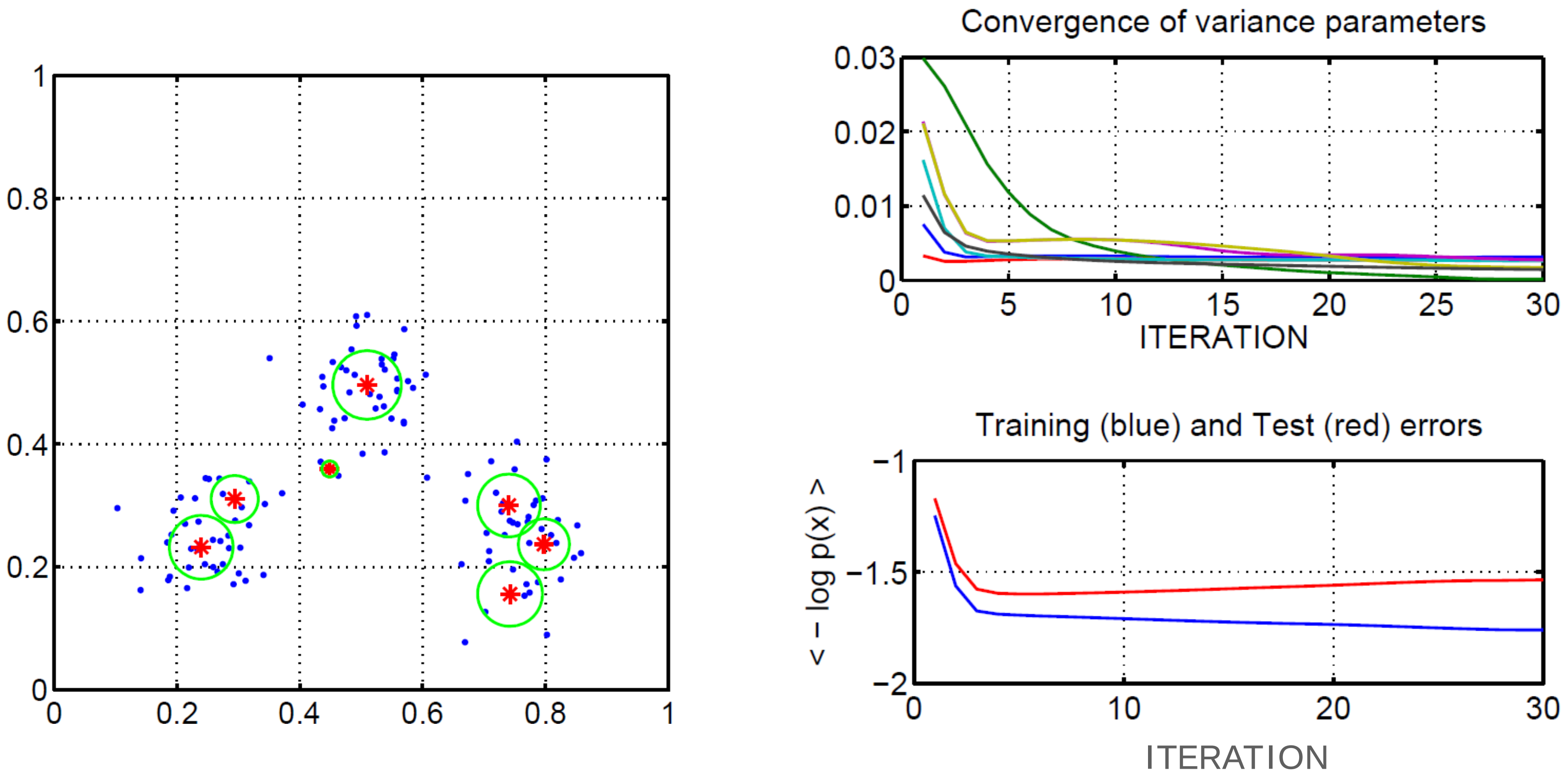
$$(\sigma_j^{\text{new}})^2 = \frac{1}{d} \frac{\sum_{n=1}^{N} \gamma_{nk}^{\text{old}} ||\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}||^2}{\sum_{n=1}^{N} \gamma_{nk}^{\text{old}}}$$

$$\pi_k^{\text{new}} = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}^{\text{old}}$$



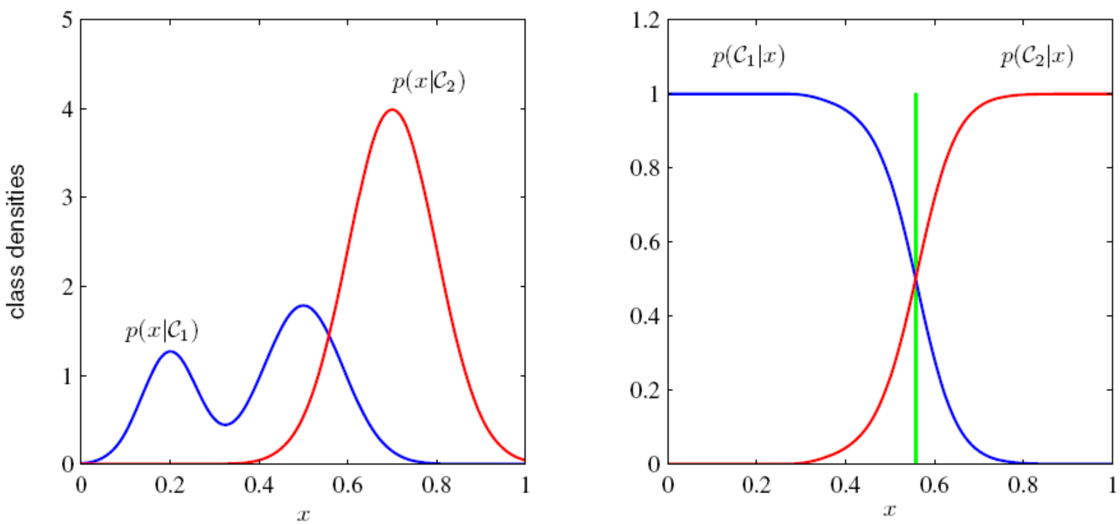
(a)
(b)
(c)
(d)
(e)
(f)
L = 1
L = 2
L = 5
L = 20

Two dimensional example from exercise 7



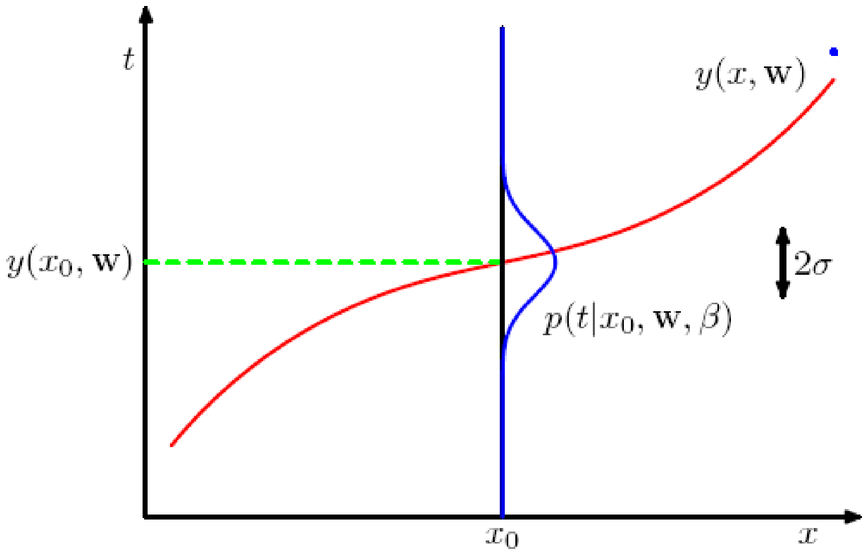
Supervised learning with mixtures of Gaussians

- Signal detection is straightforward
  - Optimal classifier is obtained by the posterior probabilities.
  - Estimate the class conditional densities p(x|C) with MoG's
  - Estimate the prior probabilities p(C) from relative rates
  - Compute posterior probs: p(C|x) = p(x|C)*p(C)/p(x)



Regression with mixture of Gaussians

Let $p(t, \mathbf{x})$ be a joint input-output density



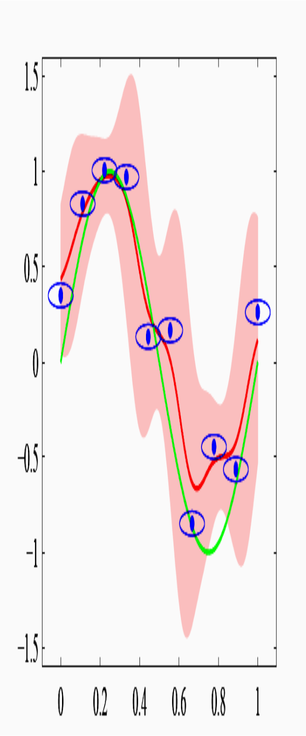
$$
\begin{aligned}
y(\mathbf{x}) &= < t|x > \\
&= \int t p(t|\mathbf{x}) dt \\
&= \frac{\int t p(t, \mathbf{x}) dt}{\int p(t, \mathbf{x}) dt}
\end{aligned}
$$

$$
p(t, \mathbf{x}) = \sum_{j=1}^{M} P(j) \frac{1}{(2\pi\sigma_j^2)^{\frac{d+c}{2}}} \exp\left(-\frac{(\mathbf{x} - \boldsymbol{\mu}_j)^2}{2\sigma_j^2} - \frac{(t - \nu_j)^2}{2\sigma_j^2} \right)
$$

Regression with Gaussian mixture

$$p(t,\mathbf{x}) = \sum_{j=1}^{M} P(j) \frac{1}{(2\pi\sigma_j^2)^{\frac{d+c}{2}}} \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_j)^2}{2\sigma_j^2} - \frac{(t-\nu_j)^2}{2\sigma_j^2}\right)$$

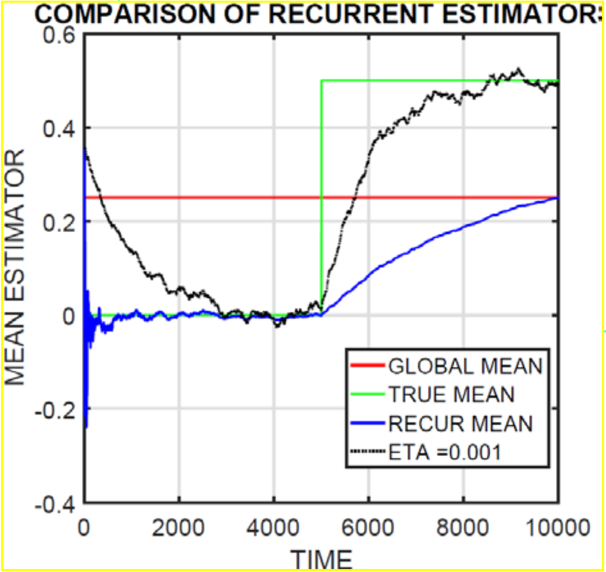
$$y(\mathbf{x}) = \frac{\sum_{j=1}^{M} \frac{P(j)\nu_j}{(2\pi\sigma_j^2)^{\frac{d}{2}}} \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_j)^2}{2\sigma_j^2}\right)}{\sum_{j'=1}^{M} \frac{P(j')}{(2\pi\sigma_{j'}^2)^{\frac{d}{2}}} \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_{j'})^2}{2\sigma_{j'}^2}\right)}$$



Dynamic estimator of the mean

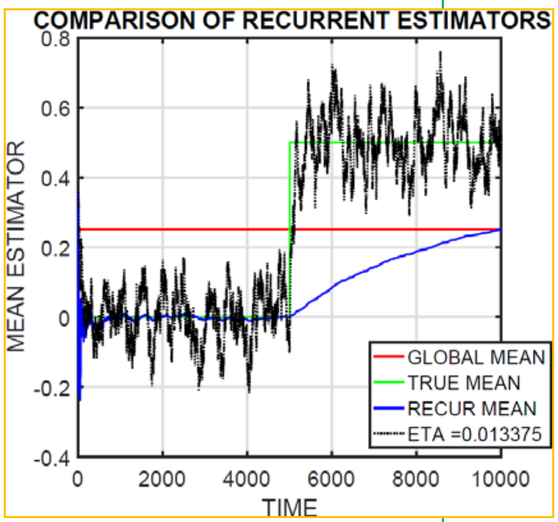
Dynamic updates for stream of data $\{x_1, x_2, ..., x_N\}$, $\mu = \frac{1}{N}\sum_{n=1}^{N} x_n$

$$
\begin{aligned}
\mu_N &= \frac{1}{N}x_N + \frac{1}{N}\sum_{n=1}^{N-1} x_n \\
&= \frac{1}{N}x_N + \frac{N-1}{N}\frac{1}{N-1}\sum_{n=1}^{N-1} x_n \\
&= \frac{1}{N}x_N + \frac{N-1}{N}\mu_{N-1} \\
&= \mu_{N-1} + \frac{1}{N}(x_N - \mu_{N-1})
\end{aligned}
$$



Non-stationarity, stochastic gradient

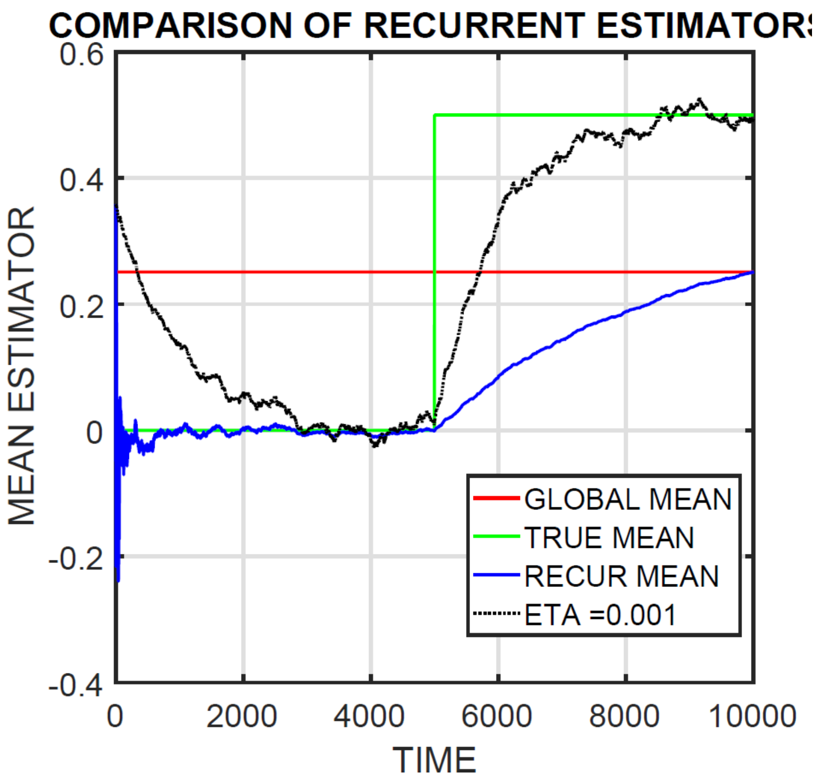
$$
\begin{aligned}
\mu_N &= \mu_{N-1} - \eta \frac{\partial}{\partial \mu}\left[\frac{1}{2}(x_N - \mu)^2\right]_{N-1} \\
&= \mu_{N-1} + \eta(x_N - \mu_{N-1})
\end{aligned}
$$

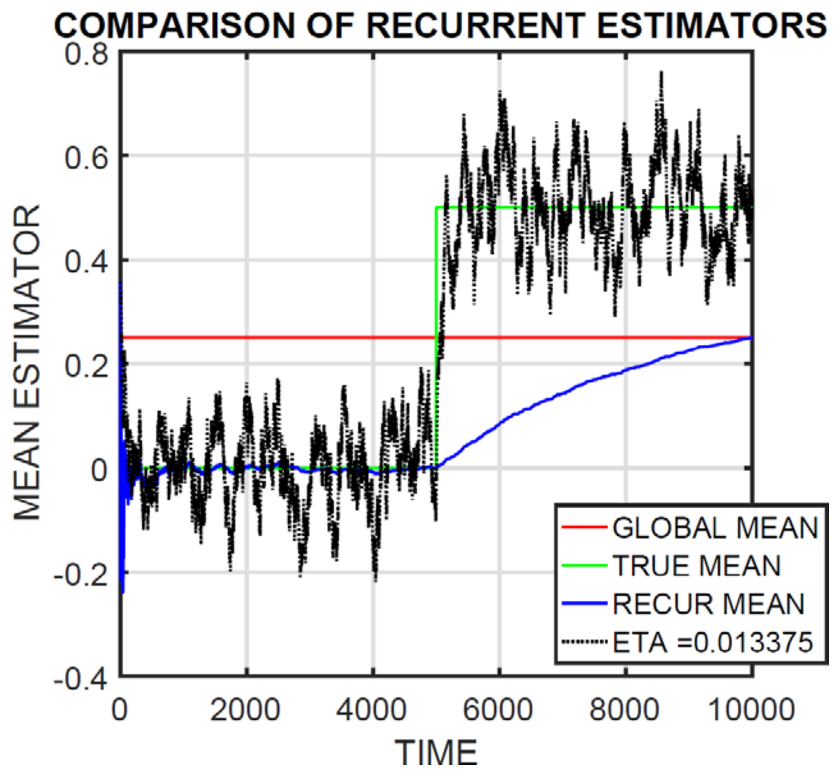


Difference equation has explicit solution

$$
\begin{aligned}
\mu_N &= \sum_{n=1}^{N} \eta(1-\eta)^{N-n} x_n \\
&= \eta \sum_{n=1}^{N} \exp\left((N-n)\log(1-\eta)\right) x_n \\
&\approx \eta \sum_{n=1}^{N} \exp\left(-(N-n)\eta\right) x_n \\
&\approx \eta \sum_{q=1}^{N} \exp(-q\eta) x_{N-q} \\
&\approx \frac{1}{W} \sum_{q=1}^{W} x_{N-q}
\end{aligned}
$$

Compare dynamic estimators in non-stationary data





Bayesian linear learning

Let $\mathcal{D} = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), ..., (\mathbf{x}_N, t_N)\}$ be a data set of N samples with $\mathbf{x} \in \mathbb{R}^d$.

$$\boxed{t = \mathbf{w}^\top \mathbf{x} + \epsilon}$$

$$p(\mathcal{D}|\mathbf{w}, \beta) = \left(\sqrt{\frac{\beta}{2\pi}}\right)^N \exp\left(-\frac{\beta}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^\top \mathbf{x}_n)^2\right)$$

assign a standard Gaussian prior to the weights $\mathbf{w} \sim \mathcal{N}(0, \alpha^{-1}\mathbf{I})$

$$\begin{aligned} p(\mathbf{w}|\alpha, \beta, \mathcal{D}) &= \frac{p(\mathcal{D}|\mathbf{w}, \beta)p(\mathbf{w}|\alpha)}{p(\mathcal{D}|\alpha, \beta)} \\ &\propto \left(\sqrt{\frac{\beta}{2\pi}}\right)^N \exp\left(-\frac{\beta}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^\top \mathbf{x}_n)^2\right)\left(\sqrt{\frac{\alpha}{2\pi}}\right)^d \exp\left(-\frac{\alpha}{2}||\mathbf{w}||^2\right) \end{aligned}$$

The product between $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, is proportional to $\mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ with mean vector and covariance matrix given by,

$$\begin{aligned} \boldsymbol{\mu}_p &= \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right) \\ \boldsymbol{\Sigma}_p &= \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}. \end{aligned}$$

Bayesian linear learning

$$p(\mathbf{w}|\alpha,\beta,\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w},\beta)p(\mathbf{w}|\alpha)}{p(\mathcal{D}|\alpha,\beta)}$$
$$\propto \left(\sqrt{\frac{\beta}{2\pi}}\right)^N \exp\left(-\frac{\beta}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^\top\mathbf{x}_n)^2\right)\left(\sqrt{\frac{\alpha}{2\pi}}\right)^d \exp\left(-\frac{\alpha}{2}||\mathbf{w}||^2\right)$$

The product between $\mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2)$, is proportional to $\mathcal{N}(\boldsymbol{\mu}_p,\boldsymbol{\Sigma}_p)$ with mean vector and covariance matrix given by,

$$\boldsymbol{\mu}_p = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right)$$
$$\boldsymbol{\Sigma}_p = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}.$$

In this case the prior is given by $\boldsymbol{\mu}_2 \equiv \boldsymbol{\mu}_{\text{prior}} = \mathbf{0}$ and $\boldsymbol{\Sigma}_2 \equiv \boldsymbol{\Sigma}_{\text{prior}} = \alpha^{-1}\mathbf{I}$. For the likelihood a bit of algebra leads to,

$$\boldsymbol{\mu}_1 \equiv \left(\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top\right)^{-1}\sum_{n=1}^{N}\mathbf{x}_n t_n$$
$$\boldsymbol{\Sigma}_1 \equiv \left(\beta\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top\right)^{-1}. \tag{3}$$

Bayesian linear learning

Hence the posterior mean vector and covariance matrix are found as

$$
\begin{aligned}
\boldsymbol{\mu}_p &\equiv \left(\alpha \mathbf{I} + \sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top\right)^{-1} \sum_{n=1}^{N} \mathbf{x}_n t_n \\
\boldsymbol{\Sigma}_p &\equiv \left(\alpha \mathbf{I} + \beta \sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top\right)^{-1} .
\end{aligned}
$$

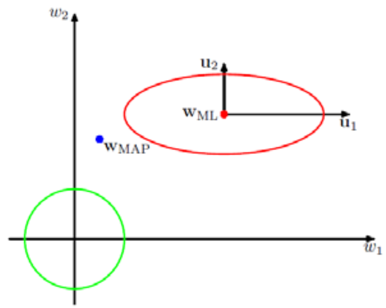


Figure 3.15 Contours of the likelihood function (red) and the prior (green) in which the axes in parameter space have been rotated to align with the eigenvectors $\mathbf{u}_i$ of the Hessian. For $\alpha = 0$, the mode of the posterior is given by the maximum likelihood solution $\mathbf{w}_{\mathrm{ML}}$, whereas for nonzero α the mode is at $\mathbf{w}_{\mathrm{MAP}} = \mathbf{m}_N$. In the direction w_1 the eigenvalue λ_1, defined by (3.87), is small compared with α and so the quantity $\lambda_1/(\lambda_1 + \alpha)$ is close to zero, and the corresponding MAP value of w_1 is also close to zero. By contrast, in the direction w_2 the eigenvalue λ_2 is large compared with α and so the quantity $\lambda_2/(\lambda_2+\alpha)$ is close to unity, and the MAP value of w_2 is close to its maximum likelihood value.

The predictive density is computed as

$$
p(t_{N+1}|\mathbf{x}_{N+1}, \mathcal{D}) \;=\; \int p(t_{N+1}|\mathbf{x}_{N+1}, \mathbf{w}) p(\mathbf{w}|\mathcal{D}) d\mathbf{w}
$$

This is again a normal distribution. We note

$$
t_{N+1} = \mathbf{w}_N^\top \mathbf{x}_{N+1} + \epsilon_{N+1}, \quad \text{and} \quad \mathbf{w}_N \sim \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p),
$$

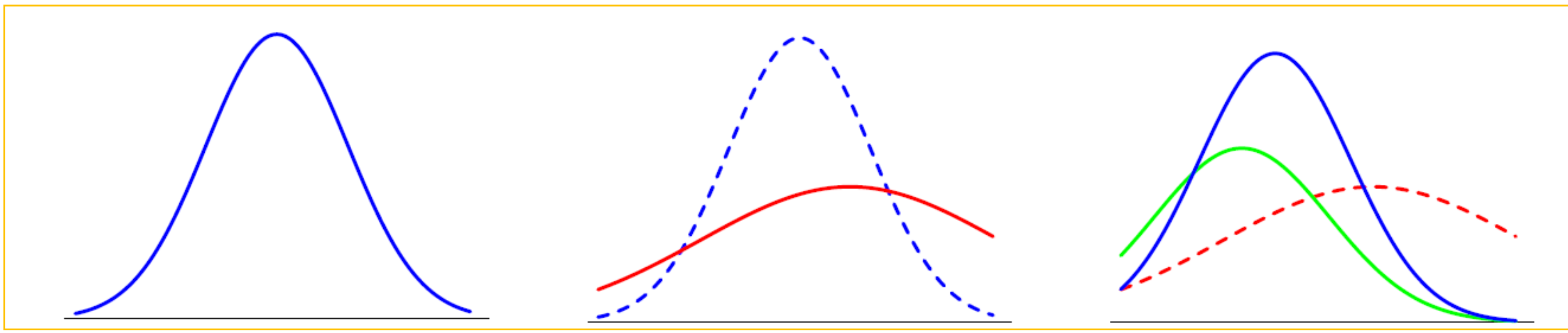
which leads to the predictive mean and variance,

$$
\begin{aligned}
\boldsymbol{\mu}_{t_{N+1}} &= \boldsymbol{\mu}_p^\top \mathbf{x}_{N+1}, \\
\sigma^2_{t_{N+1}} &= \beta^{-1} + \mathbf{x}_{N+1}^\top \boldsymbol{\Sigma}_p \mathbf{x}_{N+1}.
\end{aligned}
$$

Dynamic linear learning: Markov prior on $\mathbf{w}_n$

A natural prior is then the Markovian random walk $\mathbf{w}_n = \mathbf{w}_{n-1} + \boldsymbol{\nu}_n$ with $\boldsymbol{\nu}_n \sim \mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$,

$$p(\mathbf{w}_n|\mathbf{w}_{n-1}, \alpha) = \left(\sqrt{\frac{\alpha}{2\pi}}\right)^d \exp\left(-\frac{\alpha}{2}||\mathbf{w}_n - \mathbf{w}_{n-1}||^2\right),$$



$p(\mathbf{w}_{n-1}, \mathbf{z}_{1:(n-1)})$

$p(\mathbf{w}_n|\mathbf{w}_{n-1}, \alpha)$

$p(\mathbf{z}_n|\mathbf{w}_n, \beta)$

$p(\mathbf{w}_n, \mathbf{z}_{1:n})$

Forward recursion, to compute $\boxed{p(\mathbf{w}_n, \mathbf{z}_{1:n})}$

$$\boxed{\begin{aligned}
p(\mathbf{w}_n, \mathbf{z}_{1:n}) &= \int p(\mathbf{w}_n, \mathbf{w}_{n-1}, \mathbf{z}_{1:n}) d\mathbf{w}_{n-1} \\
&= \int p(\mathbf{w}_n, \mathbf{w}_{n-1}, \mathbf{z}_n, \mathbf{z}_{1:(n-1)}) d\mathbf{w}_{n-1} \\
&= \int p(\mathbf{z}_n | \mathbf{w}_n, \mathbf{w}_{n-1}, \mathbf{z}_{1:(n-1)}) p(\mathbf{w}_n, \mathbf{w}_{n-1}, \mathbf{z}_{1:(n-1)}) d\mathbf{w}_{n-1} \\
&= p(\mathbf{z}_n | \mathbf{w}_n) \int p(\mathbf{w}_n | \mathbf{w}_{n-1}, \mathbf{z}_{1:(n-1)}) p(\mathbf{w}_{n-1}, \mathbf{z}_{1:(n-1)}) d\mathbf{w}_{n-1} \\
&= p(\mathbf{z}_n | \mathbf{w}_n) \int p(\mathbf{w}_n | \mathbf{w}_{n-1}) p(\mathbf{w}_{n-1}, \mathbf{z}_{1:(n-1)}) d\mathbf{w}_{n-1}.
\end{aligned}}$$

The posterior distribution of $\mathbf{w}_n$, in turn, can be obtained by normalization,

$$p(\mathbf{w}_n | \mathbf{z}_{1:n}) = \frac{p(\mathbf{w}_n, \mathbf{z}_{1:n})}{\int p(\mathbf{w}_n, \mathbf{z}_{1:n}) d\mathbf{w}_n}.$$

Linear model

For the linear model we analyzed above, we get specifically,

$$
\begin{aligned}
p(\mathbf{z}_n|\mathbf{w}_n,\beta) &= p(t_n|\mathbf{w}_n,\mathbf{x}_n,\beta)p(\mathbf{x}_n) = \sqrt{\frac{\beta}{2\pi}}\exp\left(-\frac{\beta}{2}(t_n-\mathbf{w}^\top\mathbf{x}_n)^2\right)p(\mathbf{x}_n) \\
p(\mathbf{w}_n|\mathbf{w}_{n-1},\alpha) &= \left(\sqrt{\frac{\alpha}{2\pi}}\right)^d \exp\left(-\frac{\alpha}{2}||\mathbf{w}_n-\mathbf{w}_{n-1}||^2\right).
\end{aligned}
$$

Initialization

$$
p(\mathbf{w}_1,\mathbf{z}_1) \propto p(\mathbf{z}_1|\mathbf{w}_1)p(t_1|\mathbf{x}_1,\mathbf{w}_1)p(\mathbf{x}_1)
$$

$$
p(\mathbf{w}_2,\mathbf{z}_{1:2}) = p(\mathbf{z}_2|\mathbf{w}_2,\beta)\int p(\mathbf{w}_2|\mathbf{w}_1,\alpha)p(\mathbf{w}_1,\mathbf{z}_1)d\mathbf{w}_1.
$$

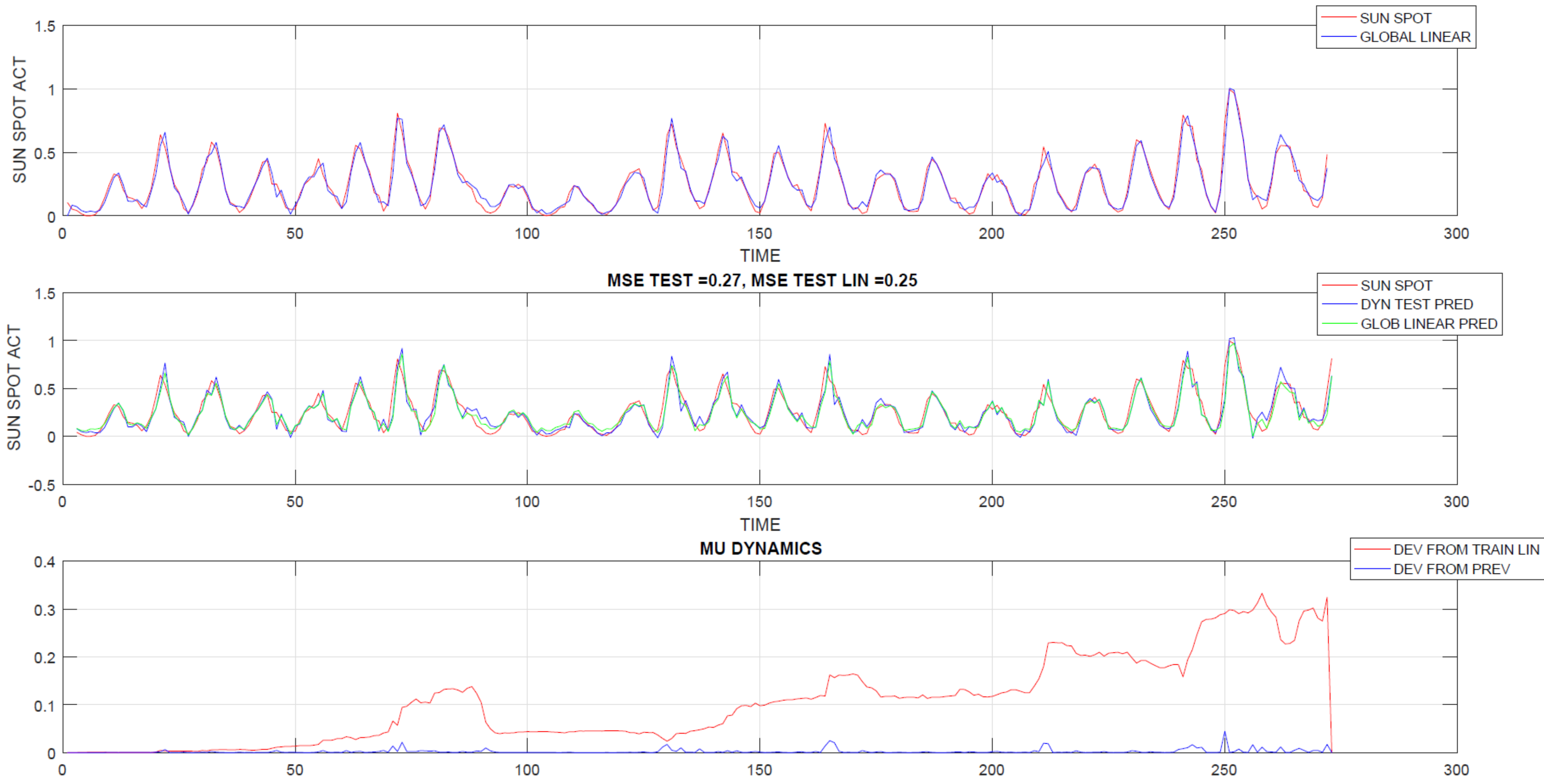
Dynamic linear model – message passing

$$p(\mathbf{w}_2, \mathbf{z}_{1:2}) \;=\; p(\mathbf{z}_2|\mathbf{w}_2, \beta) \int p(\mathbf{w}_2|\mathbf{w}_1, \alpha) p(\mathbf{w}_1, \mathbf{z}_1) d\mathbf{w}_1.$$

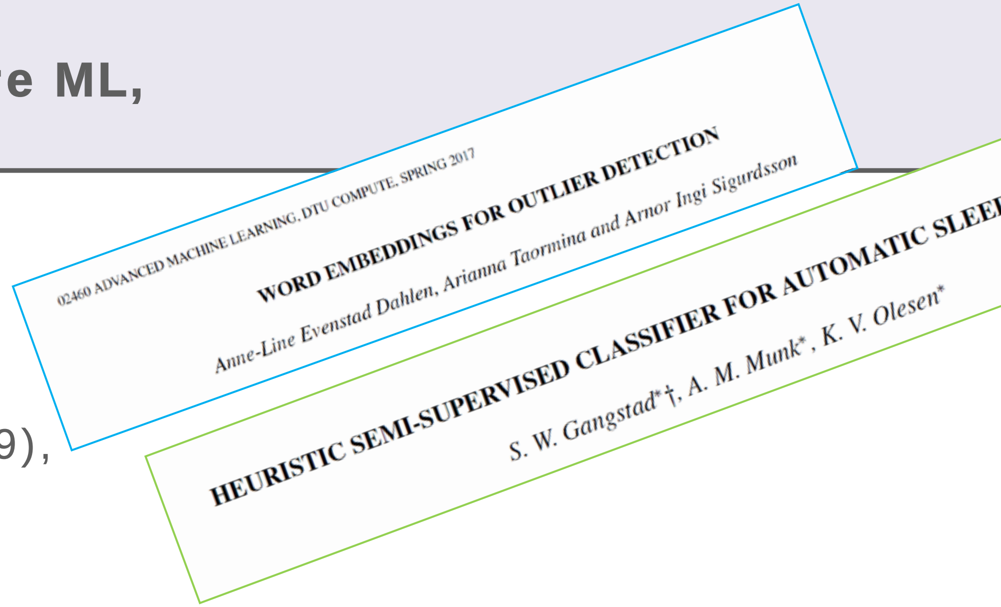
$$\begin{aligned}
\boldsymbol{\mu}_{\mathbf{w},n} &= \left(\left(\boldsymbol{\Sigma}_{\mathbf{w},n-1} + \alpha^{-1}\mathbf{I} \right)^{-1} + \beta \mathbf{x}_n \mathbf{x}_n^\top \right)^{-1} \left(\left(\boldsymbol{\Sigma}_{\mathbf{w},n-1} + \alpha^{-1}\mathbf{I} \right)^{-1} \boldsymbol{\mu}_{\mathbf{w},n-1} + \beta t_n \mathbf{x}_n \right) \\
\boldsymbol{\Sigma}_{\mathbf{w},n} &= \left(\left(\boldsymbol{\Sigma}_{\mathbf{w},n-1} + \alpha^{-1}\mathbf{I} \right)^{-1} + \beta \mathbf{x}_n \mathbf{x}_n^\top \right)^{-1}
\end{aligned}$$

α determines the effective window

Dynamic linear model: sun spots



See you later - learn more ML,


02460 ADVANCED MACHINE LEARNING, DTU COMPUTE, SPRING 2017

WORD EMBEDDINGS FOR OUTLIER DETECTION

Anne-Line Evenstad Dahlen, Arianna Taormina and Arnor Ingi Sigurdsson

HEURISTIC SEMI-SUPERVISED CLASSIFIER FOR AUTOMATIC SLEEP

S. W. Gangstad† , A. M. Munk*, K. V. Olesen**

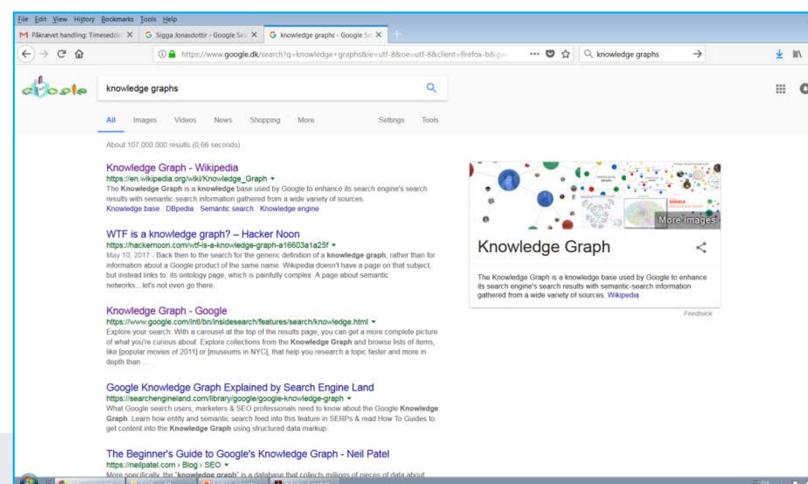
Courses:

02460 Advanced ML (spring 2019),
02456 Deep ML(fall 2018)

02XXX Machine learning systems (fall 2018)

<u>Special courses on research questions related to</u>

AI systems (knowledge graphs)
Large scale, distributed learning
Interpretability
Brain state decoding



Exam:

Jonas;Holfelt;s175331;
Torkil Sølvi;Petersen;s113059;
Sebastian;Balle;s144243;