Chris Bishop's PRML
Ch. 3: Linear Models of Regression

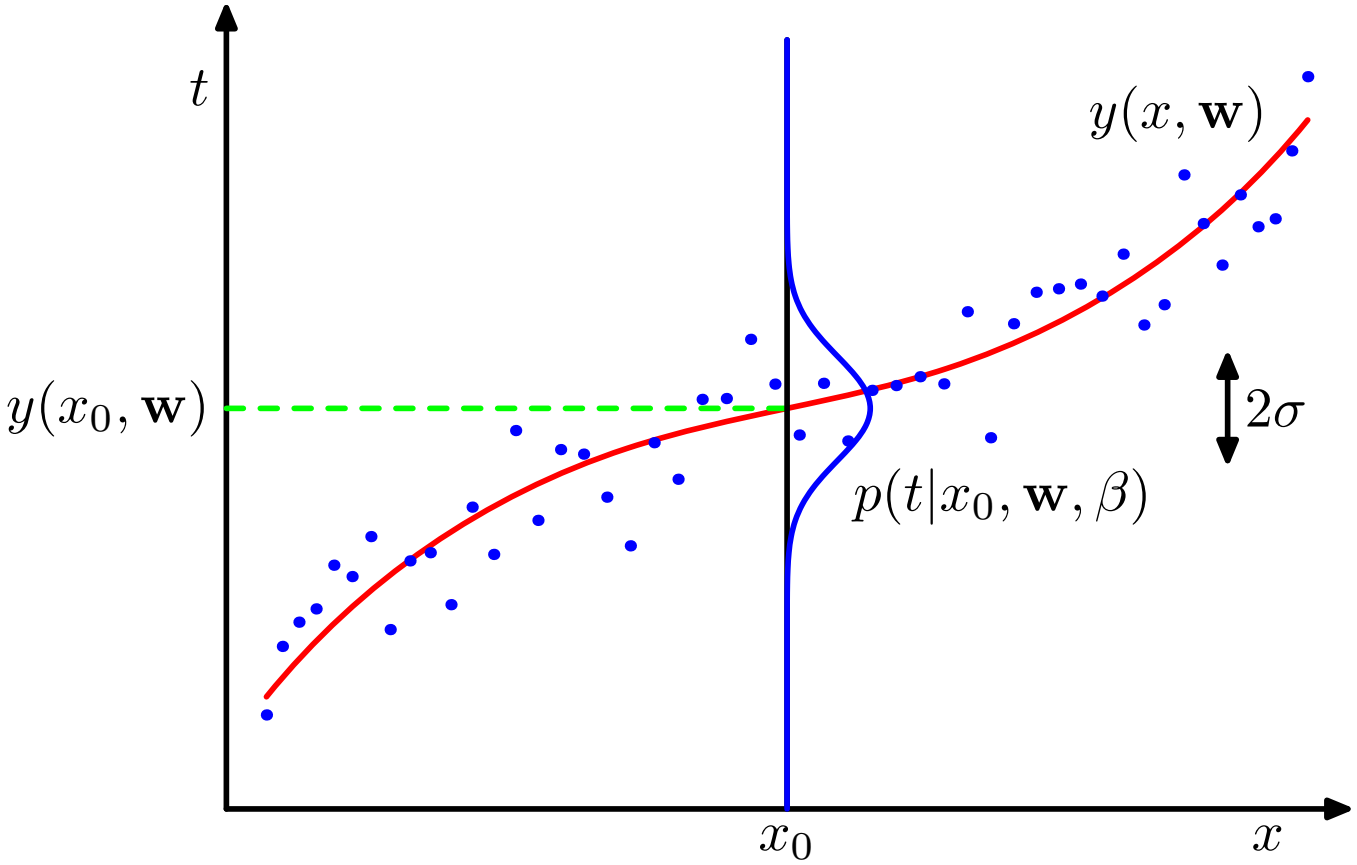
Mathieu Guillaumin & Radu Horaud

October 25, 2007

Chapter content

- An example – polynomial curve fitting – was considered in Ch. 1
- A linear combination – *regression* – of a fixed set of nonlinear functions – *basis functions*
- Supervised learning: N observations $\{\boldsymbol{x}_n\}$ with corresponding target values $\{t_n\}$ are provided. **The goal is to predict t of a new value $\boldsymbol{x}$.**
- Construct a function such that $y(\boldsymbol{x})$ is a prediction of t.
- Probabilistic perspective: model the predictive distribution $p(t|\boldsymbol{x})$.

Figure 1.16, page 29



The chapter section by section

Linear Basis Function Models

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

where:

- $\boldsymbol{w} = (w_0, \ldots, w_{M-1})^\top$ and $\boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^\top$ with $\phi_0(\boldsymbol{x}) = 1$ and $w_0 =$ bias parameter.
- In general $\boldsymbol{x} \in \mathcal{R}^D$ but it will be convenient to treat the case $x \in \mathcal{R}$
- We observe the set $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \ldots, \boldsymbol{x}_N\}$ with corresponding target variables $\boldsymbol{t} = \{t_n\}$.

Basis function choices

- **Polynomial**

$$\phi_j(x) = x^j$$

- **Gaussian**

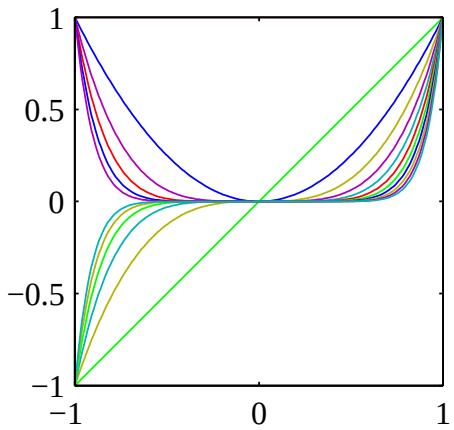
$$\phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right)$$

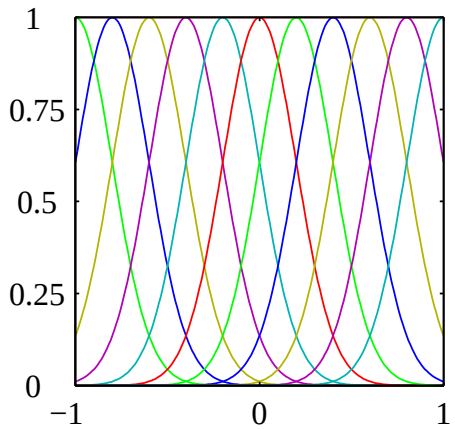
- **Sigmoidal**

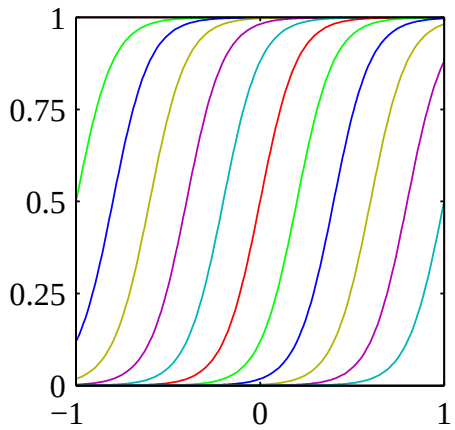
$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \text{ with } \sigma(a) = \frac{1}{1+e^{-a}}$$

- **splines**, **Fourier**, **wavelets**, etc.

Examples of basis functions







Maximum likelihood and least squares

$$t = \underbrace{y(\boldsymbol{x}, \boldsymbol{w})}_{\text{deterministic}} + \underbrace{\epsilon}_{\text{Gaussian noise}}$$

For a i.i.d. data set we have the likelihood function:

$$p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n | \underbrace{\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_n)}_{\text{mean}}, \underbrace{\beta^{-1}}_{\text{var}})$$

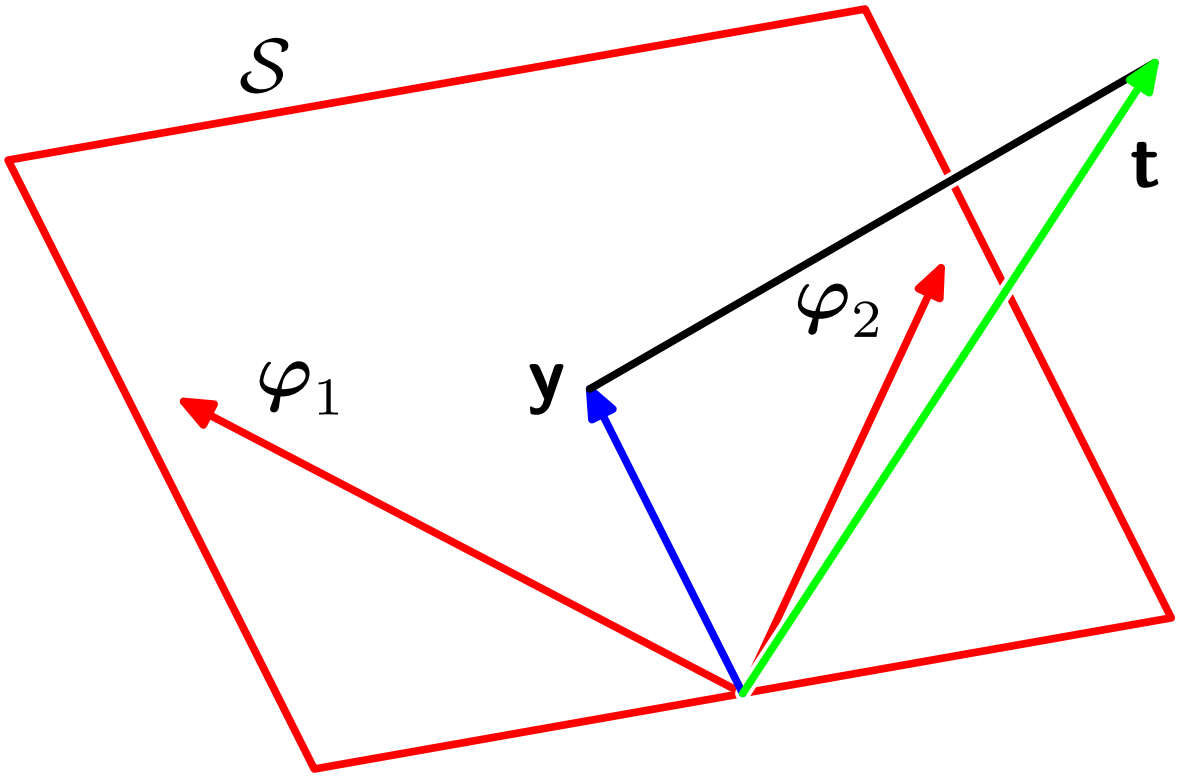
We can use the machinery of MLE to estimate the parameters $\boldsymbol{w}$ and the precision β:

$$\boldsymbol{w}_{ML} = (\Phi^\top \Phi)^{-1} \Phi^\top \boldsymbol{t} \text{ with } \Phi_{M \times N} = [\phi_{mn}(\boldsymbol{x}_n)]$$

and:

$$\beta_{ML}^{-1} = \frac{1}{N} \sum_{n=1}^{N} \left(t_n - \boldsymbol{w}_{ML}^\top \boldsymbol{\phi}(\boldsymbol{x}_n)\right)^2$$

Geometry of least squares



Sequential learning

Apply a technique known as stochastic gradient descent or sequential gradient descent, i.e.,
replace:

$$E_D(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_n)\right)^2$$

with (η is a learning rate parameter):

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} + \eta \underbrace{(t_n - \boldsymbol{w}^{(\tau)^\top}\boldsymbol{\phi}(\boldsymbol{x}_n))\boldsymbol{\phi}(\boldsymbol{x}_n)}_{\nabla E_n} \tag{3.23}$$

Regularized least squares

The total error function:

$$\frac{1}{2}\sum_{n=1}^{N}\left(t_n - \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_n)\right)^2 + \frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w}$$

$$\boldsymbol{w} = (\lambda \mathbf{I} + \Phi^\top \Phi)^{-1}\Phi^\top \boldsymbol{t}$$

Regularization has the advantage of limiting the model complexity (the appropriate number of basis functions). This is replaced with the problem of finding a suitable value of the regularization coefficient λ.

The Bias-Variance Decomposition

- Over-fitting occurs whenever the number of basis functions is large and with training data sets of limited size.
- Limiting the number of basis functions limits the flexibility of the model.
- Regularization can control over-fitting but raises the question of how to determine λ.
- The **bias-variance tradeoff** is a frequentist viewpoint of model complexity.

Back to section 1.5.5

- The regression loss-function: $L(t, y(\boldsymbol{x})) = (y(\boldsymbol{x}) - t)^2$
- The decision problem = minimize the expected loss:

$$E[L] = \int \int (y(\boldsymbol{x}) - t)^2 p(\boldsymbol{x}, t) d\boldsymbol{x} dt$$

- Solution: $y(\boldsymbol{x}) = \int t p(t|\boldsymbol{x}) dt = E_t[t|\boldsymbol{x}]$
  - this is known as the regression function
  - conditional average of t conditioned on $\boldsymbol{x}$, e.g., figure 1.28, page 47
- Another expression for the expectation of the loss function:

$$E[L] = \int (y(\boldsymbol{x}) - E[t|\boldsymbol{x}])^2 p(\boldsymbol{x}) d\boldsymbol{x} + \int (E[t|\boldsymbol{x}] - t)^2 p(\boldsymbol{x}) d\boldsymbol{x}. \tag{1.90}$$

- The optimal prediction is obtained by minimization of the expected squared loss function:

$$h(\boldsymbol{x}) = E[t|\boldsymbol{x}] = \int tp(t|\boldsymbol{x})dt \tag{3.36}$$

- The expected squared loss can be decomposed into two terms:

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

- The theoretical minimum of the first term is zero for an appropriate choice of the function $y(\boldsymbol{x})$ (for unlimited data and unlimited computing power).
- The second term arises from noise in the data and it represents the minimum achievable value of the expected squared loss.

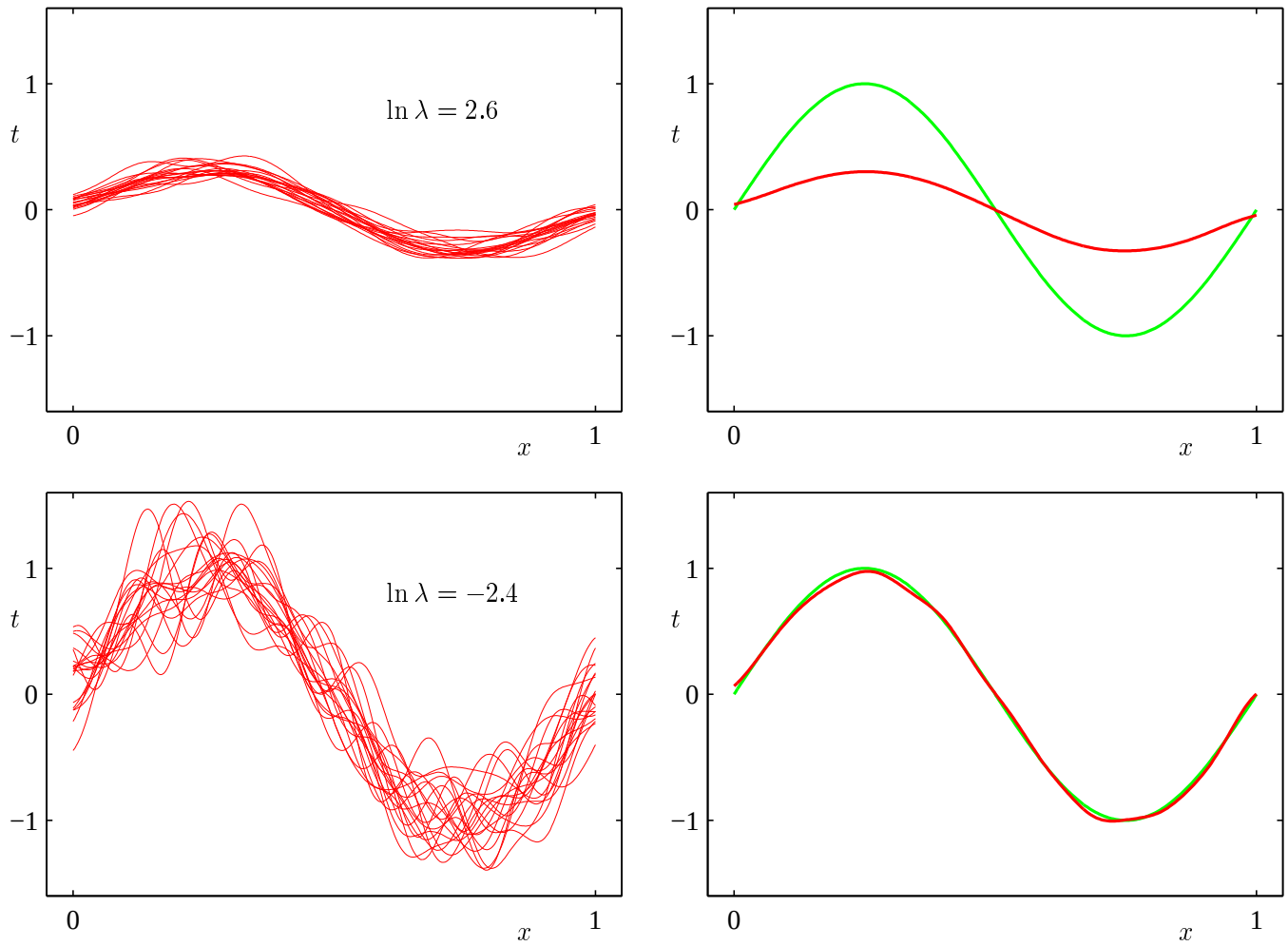
An ensemble of data sets

- For any given data set $\mathcal{D}$ we obtain a prediction function $y(\boldsymbol{x}, \mathcal{D})$.
- The performance of a particular algorithm is assessed by taking the average over all these data sets, namely $E_{\mathcal{D}}[L]$. This expands into the following terms:

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

- There is a tradeoff between bias and variance:
  - flexible models have low bias and high variance
  - rigid models have high bias and low variance
- The bias-variance decomposition provides interesting insights in model complexity, **it is of limited practical value** because several data sets are needed.

Example: L=100, N=25, M=25, Gaussian basis



Bayesian Linear Regression (1/5)

Assume additive gaussian noise with known precision β.

The likelihood function $p(\mathbf{t}|\mathbf{w})$ is the exponential of a quadratic function of $\mathbf{w}$, its conjugate prior is Gaussian:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0) \tag{3.48}$$

Its posterior is also Gaussian (2.116):

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \propto p(\mathbf{t}|\mathbf{w})p(\mathbf{w}) \tag{3.49}$$

$$\text{where} \left| \begin{array}{rcl} \mathbf{m}_N & = & \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\Phi^{\mathrm{T}}\mathbf{t}) \\ \mathbf{S}_N^{-1} & = & \mathbf{S}_0^{-1} + \beta\Phi^{\mathrm{T}}\Phi \end{array} \right. \tag{3.50/3.51}$$

- Note how this fits a sequential learning framework
- The max of a Gaussian is at its mean: $\mathbf{w}_{\text{MAP}} = \mathbf{m}_N$

Bayesian Linear Regression (2/5)

Assume $p(\mathbf{w})$ is governed by a hyperparameter α following a Gaussian law of scalar covariance (i.e. $\mathbf{m}_0 = 0$ and $\mathbf{S}_0 = \alpha^{-1}\mathbf{I}$):

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|0, \alpha^{-1}\mathbf{I}) \tag{3.52}$$

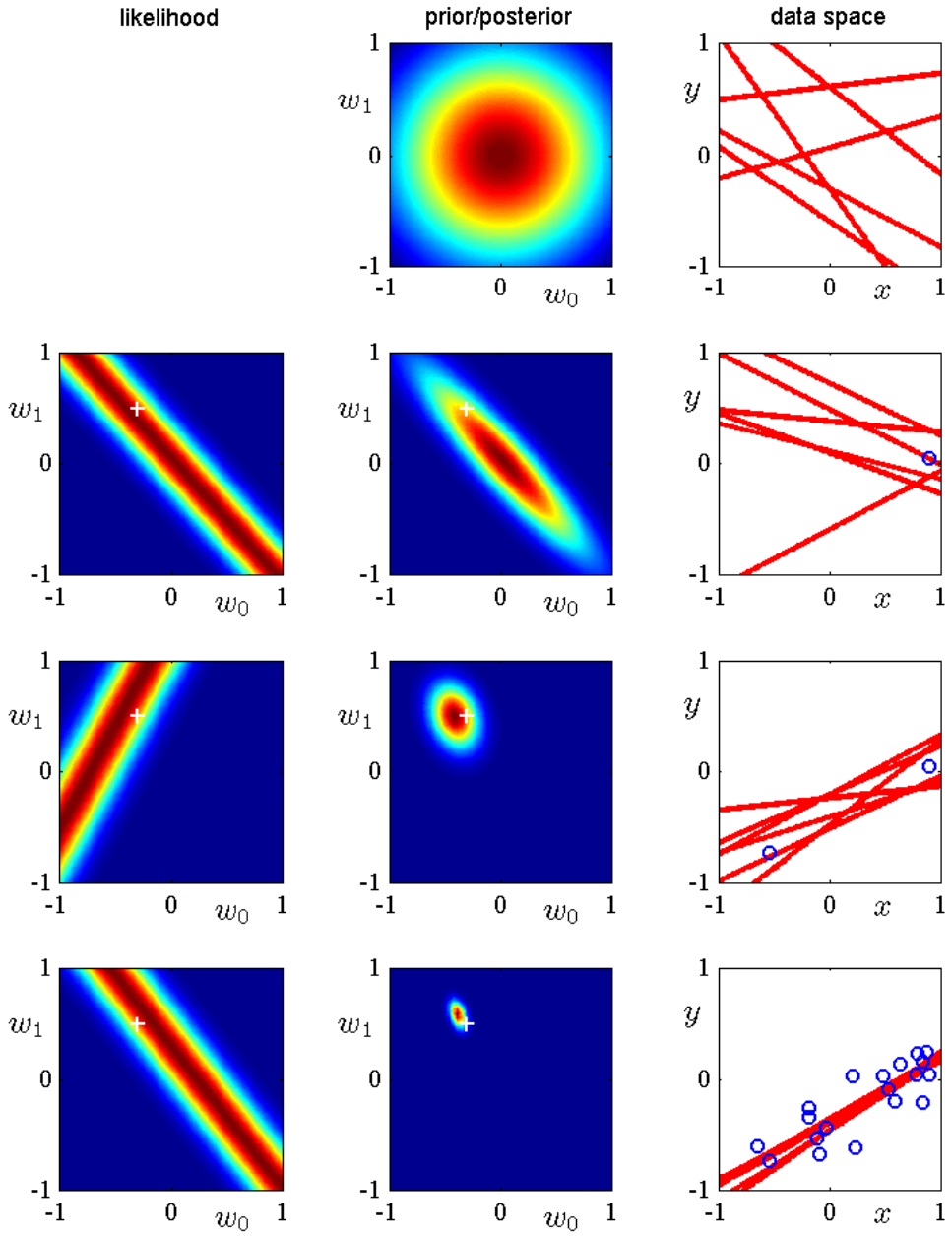
$$\text{then} \left| \begin{array}{rcl} \mathbf{m}_N & = & \beta \mathbf{S}_N \Phi^{\mathrm{T}} \mathbf{t} \\ \mathbf{S}_N^{-1} & = & \alpha \mathbf{I} + \beta \Phi^{\mathrm{T}} \Phi \end{array} \right. \tag{3.53/3.54}$$

- Note $\alpha \to 0$ implies $\mathbf{m}_N \to \mathbf{w}_{\mathrm{ML}} = (\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}\mathbf{t}$ (3.35)

Log of posterior is sum of log of likelihood and log of prior:

$$\ln p(\mathbf{w}|\mathbf{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\right)^2 - \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \text{const} \tag{3.55}$$

which is equivalent to a quadratic regularizer with coeff. α/β


likelihood
prior/posterior
data space
w_0
w_1
x
y

Bayesian Linear Regression (3/5)

In practice, we want to make predictions of t for new values of $\mathbf{x}$:

$$p(t|\mathbf{t}, \alpha, \beta) = \int p(t|\mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \alpha, \beta) d\mathbf{w} \tag{3.57}$$

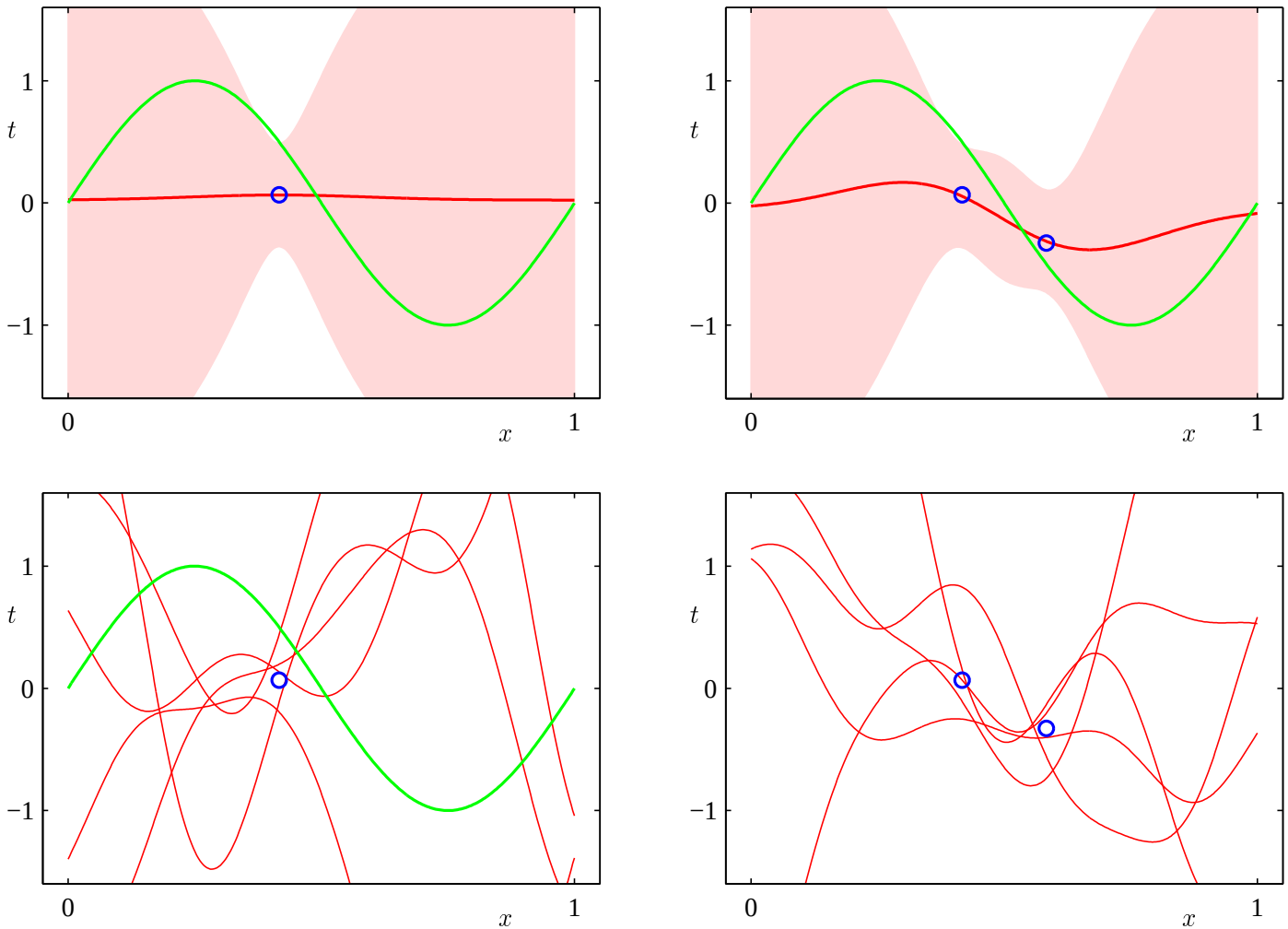
- Conditional distribution: $p(t|\mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1})$ (3.8)
- Posterior: $p(\mathbf{w}|\mathbf{t}, \alpha, \beta) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$ (3.49)

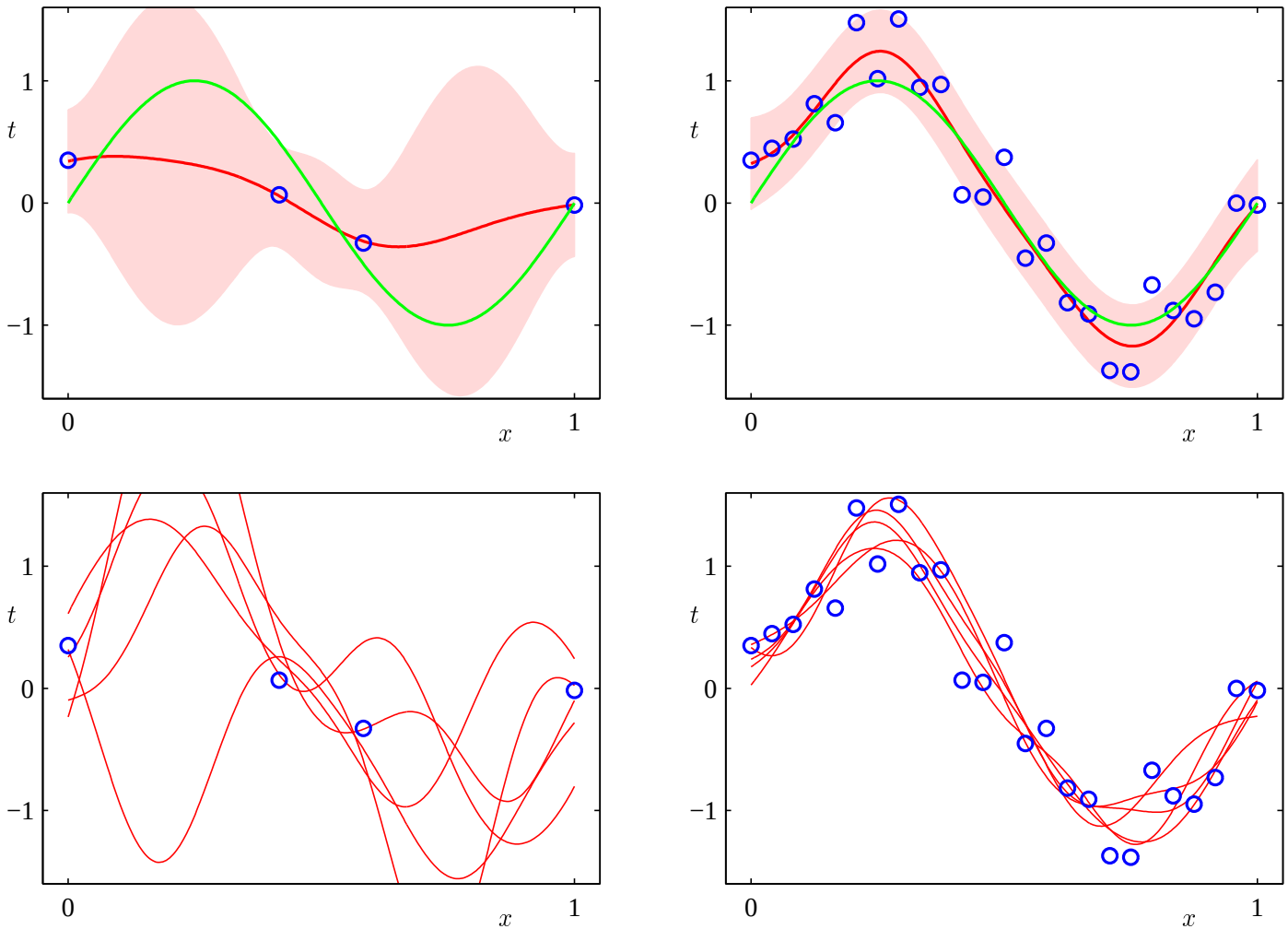
The convolution is a Gaussian (2.115):

$$p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(t|\mathbf{m}_N^{\mathrm{T}} \Phi(\mathbf{x}), \sigma_N^2(\mathbf{x})) \tag{3.58}$$

where

$$\sigma_N^2(\mathbf{x}) = \underbrace{\beta^{-1}}_{\text{noise in data}} + \underbrace{\Phi(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \Phi(\mathbf{x})}_{\text{uncertainty in } \mathbf{w}} \tag{3.59}$$


1
t
0
−1
0
x
1


1
t
0
−1
0
x
1

Bayesian Linear Regression (4/5)

$y(\mathbf{x}, \mathbf{m}_N)$ rewrites as $\sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n)\mathbf{t}_n$ where

$$k(\mathbf{x}, \mathbf{x}') = \beta \Phi(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \Phi(\mathbf{x}') \qquad (3.61\text{-}3.62)$$

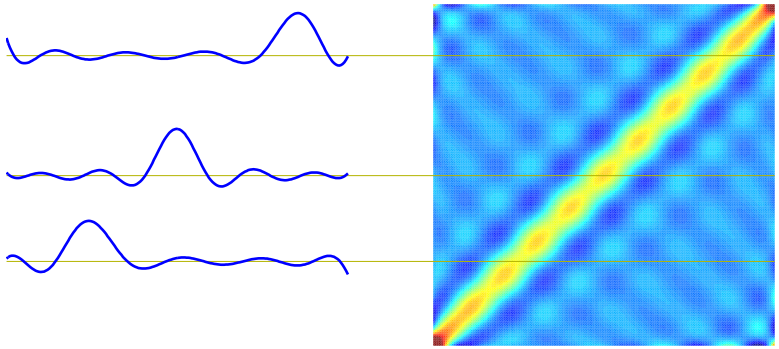
Smoother matrix, *equivalent kernel*, *linear smoother*

The kernel works as a similarity or closeness measure, giving more weight to evidence that is close to the point where we want to make the prediction

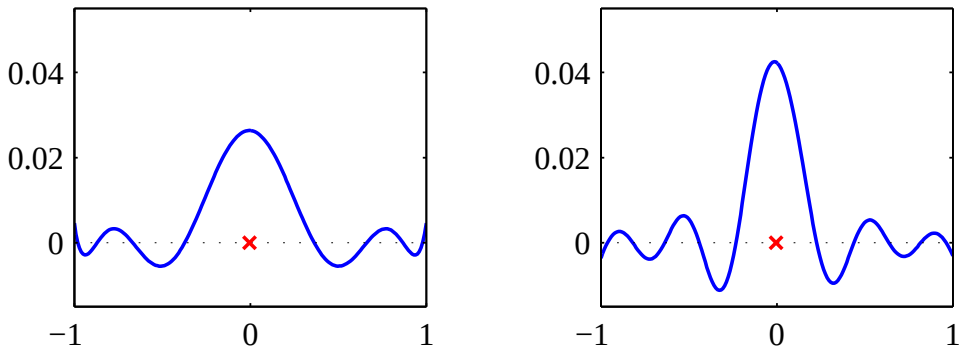
- Basis functions $\leftrightsquigarrow$ kernel duality
- With $\Psi(\mathbf{x}) = \beta^{-1/2} \mathbf{S}_N^{1/2} \Phi(\mathbf{x})$, $k(\mathbf{x}, \mathbf{x}') = \Psi(\mathbf{x})^{\mathrm{T}} \Psi(\mathbf{x}')$ (3.65)
- The kernel sums to one (over the training set)
- $\mathrm{cov}(y(\mathbf{x}), y(\mathbf{x}')) = \beta^{-1} k(\mathbf{x}, \mathbf{x}')$ (3.63)

Bayesian Linear Regression (5/5)

Kernel from Gaussian basis functions



Kernels at $\mathbf{x} = 0$ for kernels corresponding (left) to the polynomial basis functions and (right) to the sigmoidal basis functions.



Bayesian Model Comparison (1/2)

The overfitting that appears in ML can be avoided by marginalizing over the model parameters.

- Cross-validation is no more useful
- We can use all the data for better training the model
- We can compare models based on training data alone

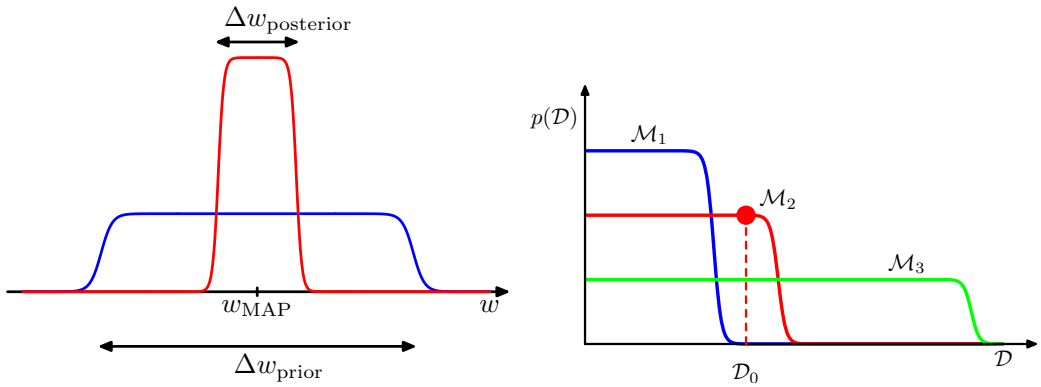
$$p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_i) \tag{3.66}$$

$p(\mathcal{D}|\mathcal{M}_i)$: *model evidence* or *marginal likelihood*.

Using *model selection* and assuming the posterior $p(w|\mathcal{D}, \mathcal{M}_i)$ is sharply peaked at w_{MAP} (single parameter case):

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w)dw \simeq p(\mathcal{D}|w_{\text{MAP}})\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}} \tag{3.70}$$

Bayesian Model Comparison (2/2)



Back to multiple parameters, assuming they share the same Δw ratio, the complexity penalty is linear in M:

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\mathbf{w}_{\text{MAP}}) + M \ln \left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}} \right) \qquad (3.72)$$

About $p(\mathcal{D}|\mathcal{M}_i)$:

- if $\mathcal{M}_i$ is too simple, bad fitting of the data
- if $\mathcal{M}_i$ is too complex/powerful, the probability of generating the observed data is washed out

The evidence approximation (1/2)

Fully bayesian treatment would imply marginalizing over hyperparameters and parameters, but this is intractable:

$$p(t|\mathbf{t}) = \iiint p(t|\mathbf{w},\beta)p(\mathbf{w}|\mathbf{t},\alpha,\beta)p(\alpha,\beta|\mathbf{t})d\mathbf{w}d\alpha d\beta \qquad (3.74)$$

An approximation is found by maximizing the marginal likelihood function $p(\alpha,\beta|\mathbf{t}) \propto p(\mathbf{t}|\alpha,\beta)p(\alpha,\beta)$ to get $(\hat{\alpha},\hat{\beta})$ (*empirical Bayes*).

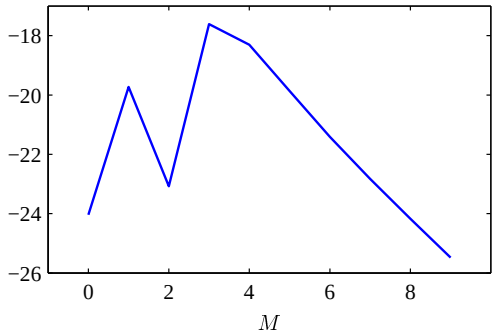
$$\ln p(\mathbf{t}|\alpha,\beta) = \frac{M}{2}\ln\alpha + \frac{N}{2}\ln\beta - E(\mathbf{m}_N) - \frac{1}{2}\ln|\mathbf{S}_N^{-1}| - \frac{N}{2}\ln(2\pi)$$
$$(3.77 \rightarrow 3.86)$$

Assuming $p(\alpha,\beta|\mathbf{t})$ is highly peaked at $(\hat{\alpha},\hat{\beta})$:

$$p(t|\mathbf{t}) \simeq p(t|\mathbf{t},\hat{\alpha},\hat{\beta}) = \int p(t|\mathbf{w},\hat{\beta})p(\mathbf{w},\hat{\alpha},\hat{\beta})d\mathbf{w} \qquad (3.75)$$

The evidence approximation (2/2)

Plot of the model evidence $\ln p(\mathbf{t}|\alpha, \beta)$ versus M, the model complexity, for the polynomial regression of the synthetic sinusoidal example (with fixed α).



The computation for $(\hat{\alpha}, \hat{\beta})$ give rise to $\gamma = \alpha \mathbf{m}_N^{\mathrm{T}} \mathbf{m}_N$ (3.90)

γ has the nice interpretation of being the *effective number of parameters*