LUND UNIVERSITY

FMAN-45: Machine Learning

Lecture 3: Linear Models for Regression

Cristian Sminchisescu

Machine Learning: Frequentist vs. Bayesian

- In the **frequentist setting**, we seek a fixed parameter (vector), with value(s) determined by an estimator (ML, MAP, etc.)
- Analysis (e.g. the bias-variance decomposition) is realized by computing expectations over all possible datasets
- Empirically, uncertainty (error bars) for the estimate is obtained by approximating the distribution of possible datasets, e.g. by bootstrap - sampling the original, observed data, with replacement
- In a **Bayesian perspective**, there is a single dataset, the one actually observed, and uncertainty is expressed through a probability distribution (either assumed or computed) over all parameters and variables of interest
- In this lecture we will analyze linear models both from a frequentist and from a Bayesian perspective

Linear Models

- It is mathematically easy to fit linear models to data and we can get insight by analyzing this relatively simple case. We already studied the polynomial curve fitting problem in the previous lectures
- There are many ways to make linear models more powerful while retaining their attractive mathematical properties
- By using non-linear, non-adaptive basis functions, we obtain generalized linear models. These offer non-linear mappings from inputs to outputs but are linear in their parameters. Typically, only the linear part of the model is learnt
- By using kernel methods we can handle expansions of the raw data that use a huge number of non-linear, non-adaptive basis functions. By using large margin loss functions, we can avoid overfitting even when we use huge numbers of basis functions

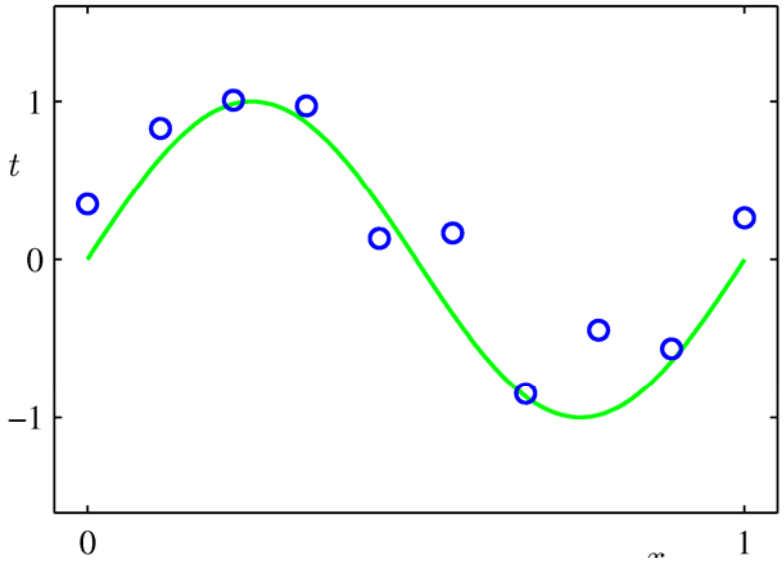
adapted from MLG@Toronto

The Loss Function

- Once a model class is chosen, fitting the model to data is typically done by finding the parameter values that minimize a loss function
- There are many possible loss functions. What criterion should we use in selecting one?
  - Loss that makes analytic calculations easy (squared error)
  - Loss that makes the fitting correspond to maximizing the likelihood of the training data, given some noise model for the observed outputs
  - Loss that makes it easy to interpret the learned coefficients (easy if mostly zeros)
  - Loss that corresponds to the real loss on a practical application (training/testing losses are often asymmetric)

Linear Basis Function Models (1)

Consider the polynomial curve fitting. The model is linear in the parameters $\mathbf{w}$ and non-linear in inputs $\mathbf{x}$



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

Linear Basis Function Models (2)

Generally

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

where $\phi_j(x)$are known as *basis functions*.

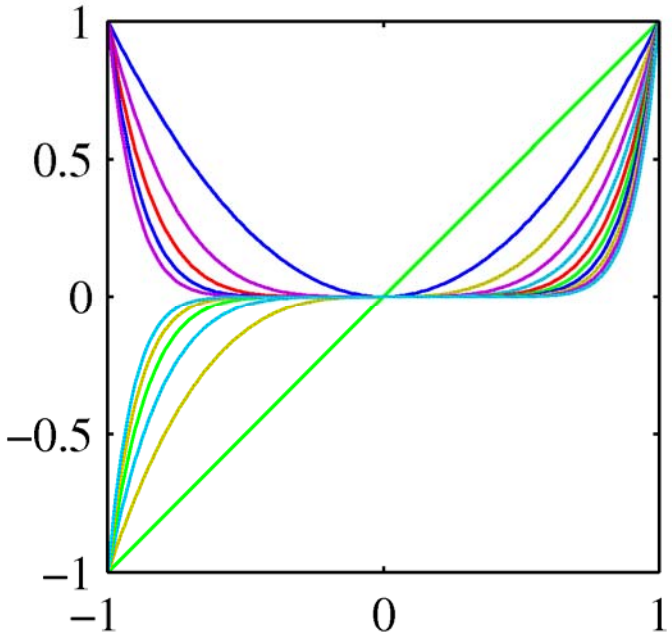
- Typically, $\phi_0(x) = 1$ (dummy basis function) so that w acts as a bias (allows for any fixed offset in the data – do not confuse with statistical bias)
- In the simplest case, we use linear basis functions: $\phi_d(x) = x_d$

Linear Basis Function Models (3)

Polynomial basis functions:

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

These are global; a small change in x affects all basis functions.



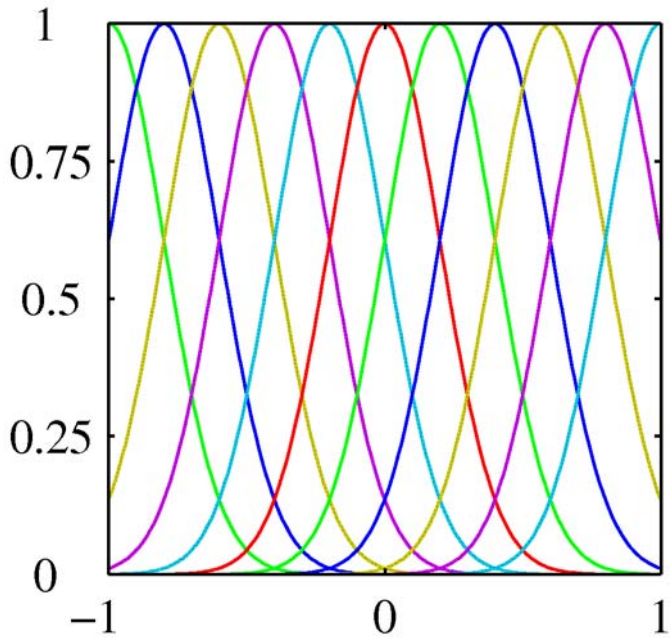
The issue can be resolved by dividing input space into regions and fitting a different polynomial in each region. This leads to *spline functions*.

Linear Basis Function Models (4)

Gaussian basis functions:

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

These are local; a small change in x only affect nearby basis functions. μ_j and s control location and scale (width).



Linear Basis Function Models (5)

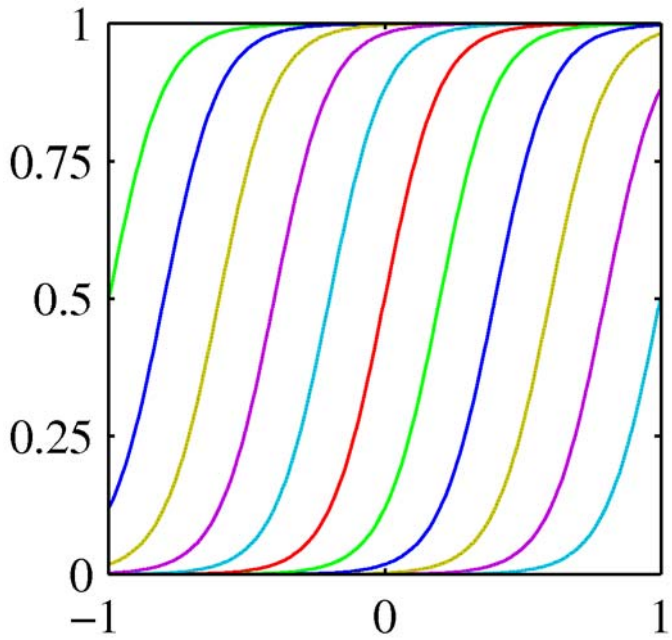
Sigmoidal basis functions:

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right)$$

where

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

Also these are local; a small change in x only affect nearby basis functions. μ_j and s control location and scale (slope).



Other Basis Function Models (6)

- In a Fourier representation, each basis function represents a given frequency and has infinite spatial extent

- Wavelets are localized in both space and frequency, and by definition are linearly orthogonal

Maximum Likelihood and Least Squares (1)

Assume observations from a deterministic function with added Gaussian noise:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \qquad \text{where} \qquad p(\epsilon|\beta) = \mathcal{N}(\epsilon|0, \beta^{-1})$$

which is the same as saying,

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}) = (\beta/2\pi)^{\frac{1}{2}} \exp\{-\frac{\beta}{2}\big(t - y(x, w)\big)^2\}$$

Given observed inputs, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and targets, $\mathbf{t} = [t_1, \ldots, t_N]^{\mathrm{T}}$, we obtain the likelihood function

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}).$$

Maximum Likelihood and Least Squares (2)

Taking the logarithm, we get

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{w},\beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n),\beta^{-1}) \\ &= \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(\mathbf{w})\end{aligned}$$

where

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

is the sum-of-squares error

Maximum Likelihood and Least Squares (3)

Computing the gradient and setting it to zero yields

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t}|\mathbf{w}, \beta) = \beta \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\} \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} = \mathbf{0}.$$

Solving for $\mathbf{w}$, we get

$$\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t}$$

normal equations

The Moore-Penrose pseudo-inverse, $\boldsymbol{\Phi}^{\dagger}$.

where

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

design matrix (NxM)

Maximum Likelihood and Least Squares (4)

$$E_D(w) = \frac{1}{2}\sum_{\{n=1\}}^{N}\{t_n - w_0 - \sum_{j=1}^{M-1} w_j\phi_j(x_n)\}^2$$

Maximizing with respect to the bias, w_0 , we get

$$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j\overline{\phi_j}$$

$$= \frac{1}{N}\sum_{n=1}^{N} t_n - \sum_{j=1}^{M-1} w_j \frac{1}{N}\sum_{n=1}^{N}\phi_j(\mathbf{x}_n).$$

We can also maximize with respect to β, giving

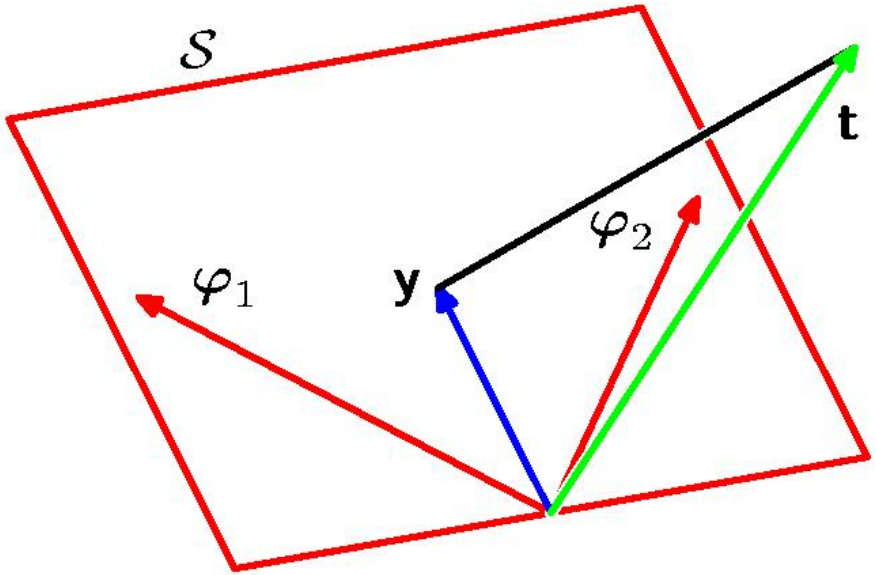
$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{t_n - \mathbf{w}_{\mathrm{ML}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

Bias compensates for the difference between averages of the target values and weighted sum of averages of basis function values

Inverse noise precision is given by residual variance of targets around regression function

Geometry of Least Squares

- The space has one axis for each training case. The vector of target values, $\mathbf{t} = (t_1, t_2 \ldots t_N)^T$ is a point in space
- Each vector of the values of one component of the input φ_j is also a point in this space.
- The input component vectors span a subspace, S. A weighted sum of the input component vectors must lie in S.
- The optimal solution is the **orthogonal projection** of the vector of target values onto S.



$$\mathbf{y} = \mathbf{\Phi}\mathbf{w}_{\mathrm{ML}} = [\varphi_1, \ldots, \varphi_M]\,\mathbf{w}_{\mathrm{ML}}.$$

$$\mathbf{w}_{\mathrm{ML}} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}$$

$$\mathbf{\Phi} = \begin{matrix} \\ \phi(\mathbf{x}_2) \\ \\ \\ \end{matrix} \overset{\varphi_1(\mathbf{x})}{\begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}}$$

Sequential Learning

Data items considered one at a time (a.k.a. online learning); use stochastic (sequential) gradient descent:

$$\begin{aligned} \mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} - \eta \nabla E_n \\ &= \mathbf{w}^{(\tau)} + \eta (t_n - \mathbf{w}^{(\tau)\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n)) \boldsymbol{\phi}(\mathbf{x}_n). \end{aligned}$$

This is known as the *least-mean-squares (LMS) algorithm*. Issue: how to choose η?

Regularized Least Squares (1)

Consider the error function:

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

Data term + Regularization term

With the sum-of-squares error function and a quadratic regularizer, we get

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

which is minimized by

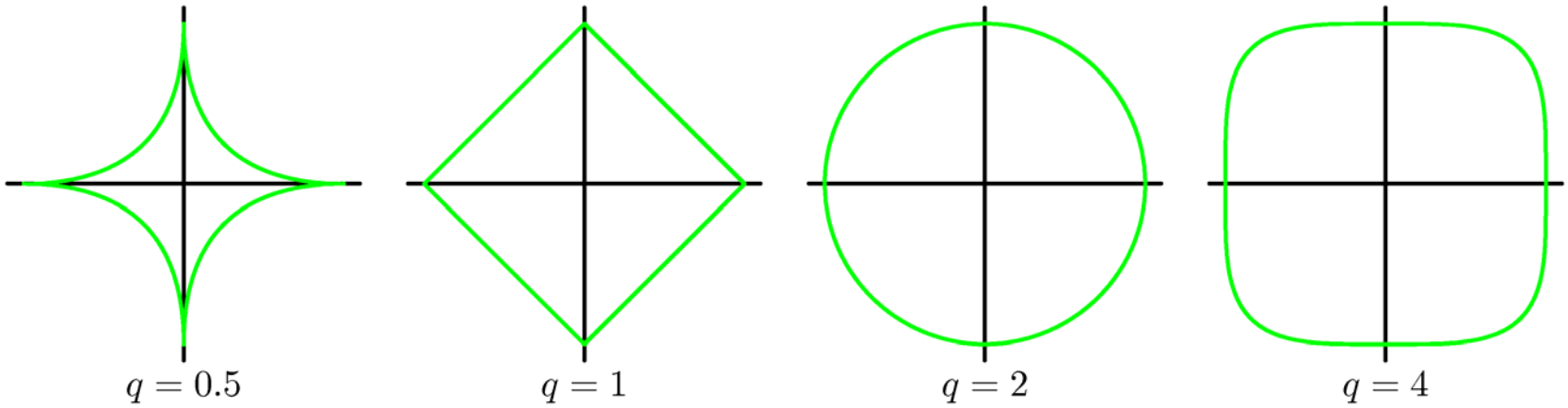
$$\mathbf{w} = \left(\lambda\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}.$$

λ is called the regularization coefficient.

Regularized Least Squares (2)

With a more general regularizer, we have

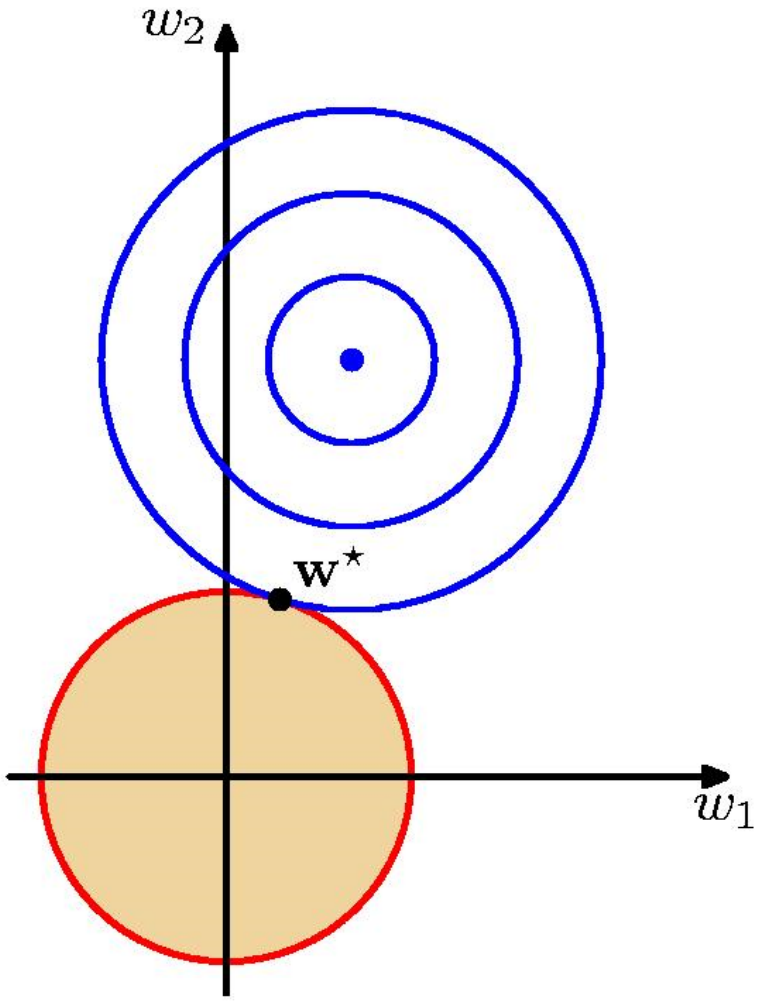
$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$



Lasso

Quadratic

Effect of a Quadratic Regularizer



- The overall cost function is the sum of two quadratic functions
- The sum is also a quadratic
- The combined minimum lies on the line between the minimum of the squared error and the origin
- The regularizer shrinks the weights, but does not make them exactly zero

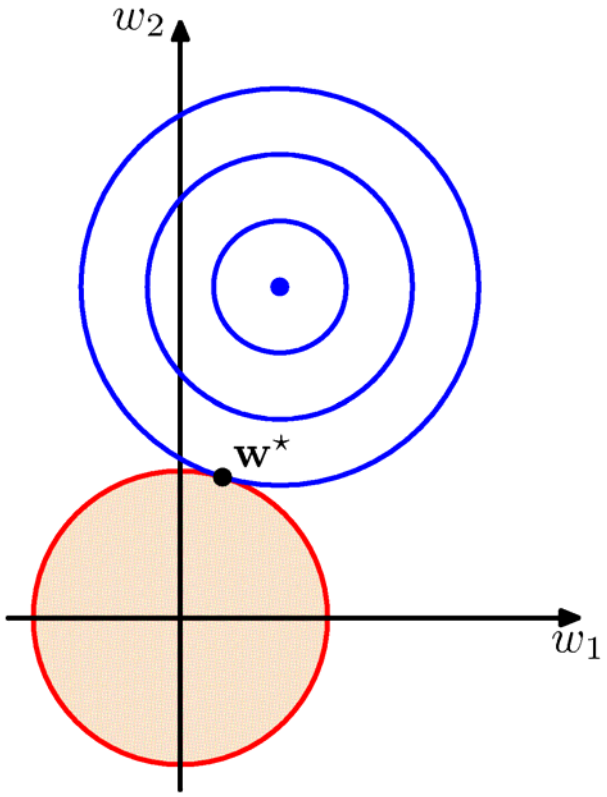
Lasso: penalizing the absolute values of weights

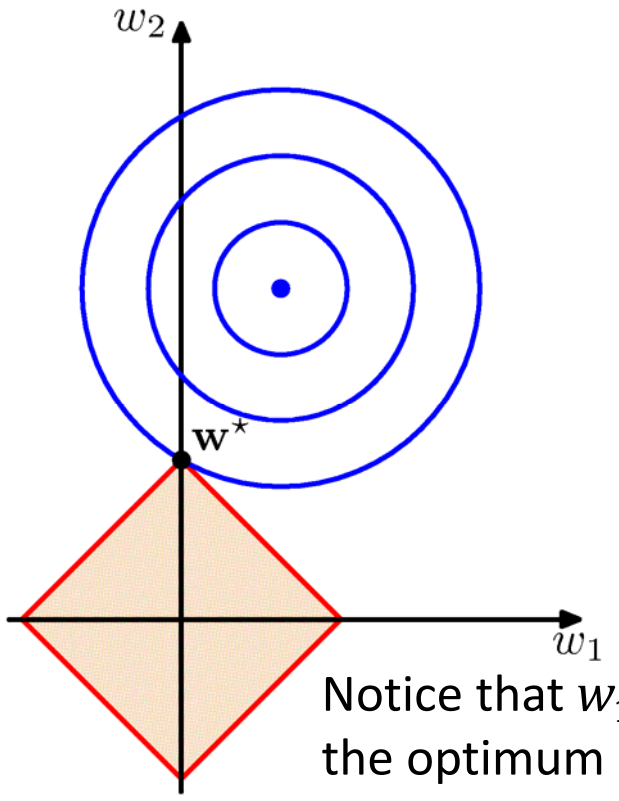
$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|$$

- Finding the minimum requires quadratic programming but it is still unique because the cost function is convex
  - Sum of two convex functions: a bowl plus an inverted pyramid
- As λ is increased, many of the weights go exactly to zero
  - This is good for interpretation: identify factors causing a given disease, given input vector $\mathbf{x}$ containing many measurement components (temperature, blood pressure, sugar level, etc.)
  - It is also good in preventing overfitting

Penalty over square weights vs. Lasso

Lasso tends to generate sparser solutions than a quadratic regularizer

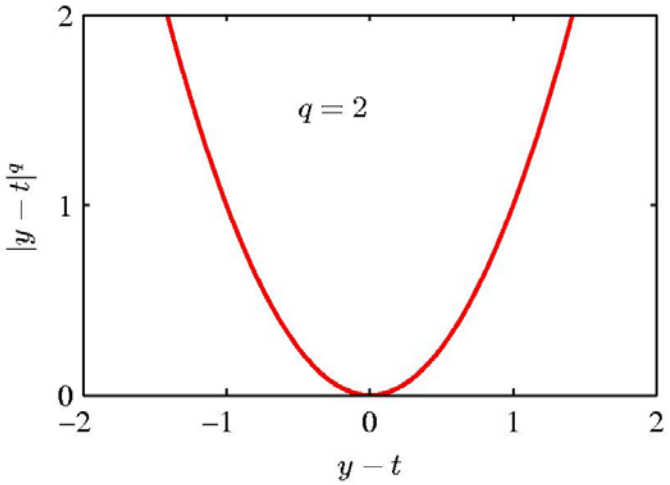




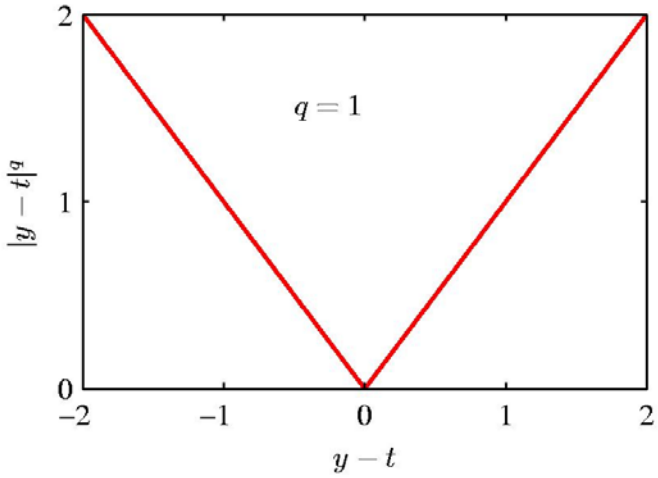
Notice that $w_1 = 0$ at the optimum

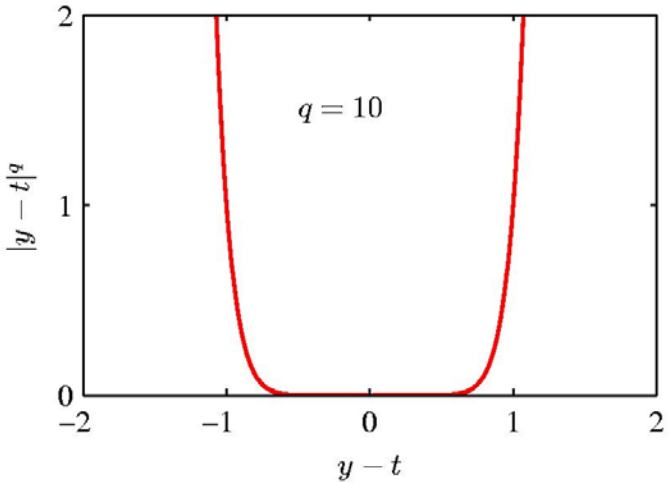
1-d loss functions with different powers

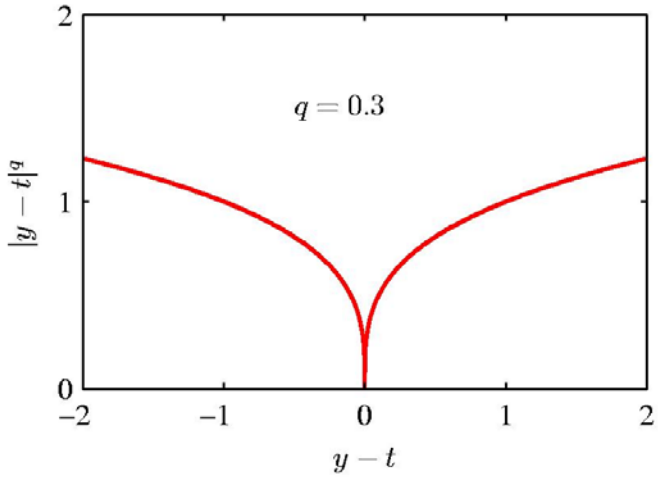
Negative log of Gaussian



Negative log of Laplacian







Multiple Outputs

- If there are multiple outputs we can often treat the learning problem as a set of independent problems, one per output

- This approach is suboptimal if the output noise is correlated and changes across datapoints
  - There are ways to handle correlations, e.g. de-correlate the outputs using linear/non-linear dimensionality reduction, then perform independent predictions

- Even though they are independent problems we can save work by only multiplying the input vectors by the inverse covariance of the input components once

Multiple Outputs (1)

Analogously to the single output case we have:

$$\begin{aligned} p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) &= \mathcal{N}(\mathbf{t}|\mathbf{y}(\mathbf{W}, \mathbf{x}), \beta^{-1}\mathbf{I}) \\ &= \mathcal{N}(\mathbf{t}|\mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}), \beta^{-1}\mathbf{I}). \end{aligned}$$

Given observed inputs, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and targets, $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^{\mathrm{T}}$, we obtain the log likelihood function

$$\begin{aligned} \ln p(\mathbf{T}|\mathbf{X}, \mathbf{W}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(\mathbf{t}_n|\mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}\mathbf{I}) \\ &= \frac{NK}{2} \ln\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2} \sum_{n=1}^{N} \left\|\mathbf{t}_n - \mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\|^2. \end{aligned}$$

Note the same set of basis functions to model all of the components of the target vectors

Multiple Outputs (2)

Maximizing with respect to $\boldsymbol{W}$, we obtain

$$\mathbf{W}_{\mathrm{ML}} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{T}.$$

If we consider a single target variable, $\mathbf{t_k}$, we see that

$$\mathbf{w}_k = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}_k = \mathbf{\Phi}^{\dagger}\mathbf{t}_k$$

where $\mathbf{t}_k = [t_{1k}, \ldots, t_{Nk}]^{\mathrm{T}}$, which is identical with the single output case.

Decision Theory for Regression (recall L2)

Inference step

Determine $p(\mathbf{x}, t)$

Decision step

For given $\mathbf{x}$, make optimal prediction $y(\mathbf{x})$ for t

Loss function:

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

The Squared Loss Function (L2)

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

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

substitute and integrate over t, w. r. t $p(t|\mathbf{x})$

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

$$y(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] \quad \text{where } E[t|\boldsymbol{x}] \equiv E_t[t|\boldsymbol{x}] = \int tp(t|\boldsymbol{x})dt$$

regression function

- Optimal least squares predictor given by the conditional average
- First term in $E[L]$ gives prediction error according to conditional mean estimates
- Second term is independent of $y(\mathbf{x})$. It measures the intrinsic variability of the target data, hence it represents the irreducible minimum value of loss, independent of $y(\mathbf{x})$

The Bias-Variance Decomposition (1)

Expected squared loss,

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

where

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

The second term of $E[L]$ corresponds to the noise inherent in the random variable t.

What about the first term?

Frequentist's Approach

- Making a point estimate of $\boldsymbol{w}$ based on the data set D
- Interpret the uncertainty of this estimate through the following thought experiment:

1. Suppose we had a large number of datasets each of size N
2. Each dataset is drawn independently from the distribution $p(t, \boldsymbol{x})$
3. For any given dataset D, we can run our learning algorithm and obtain a prediction function $y(\boldsymbol{x}; D)$
4. Different data sets from the ensemble will give different functions and consequently different values of the squared loss
5. The performance of a particular learning algorithm is then assessed by averaging over this ensemble of data sets

The Bias-Variance Decomposition (2)

Suppose we were given multiple data sets, each of size N. Any particular data set, D, will give a particular function $y(\mathbf{x}; D)$. We then have

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

The Bias-Variance Decomposition (3)

Taking the expectation over D yields

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

The Bias-Variance Decomposition (4)

Thus we can write

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

where

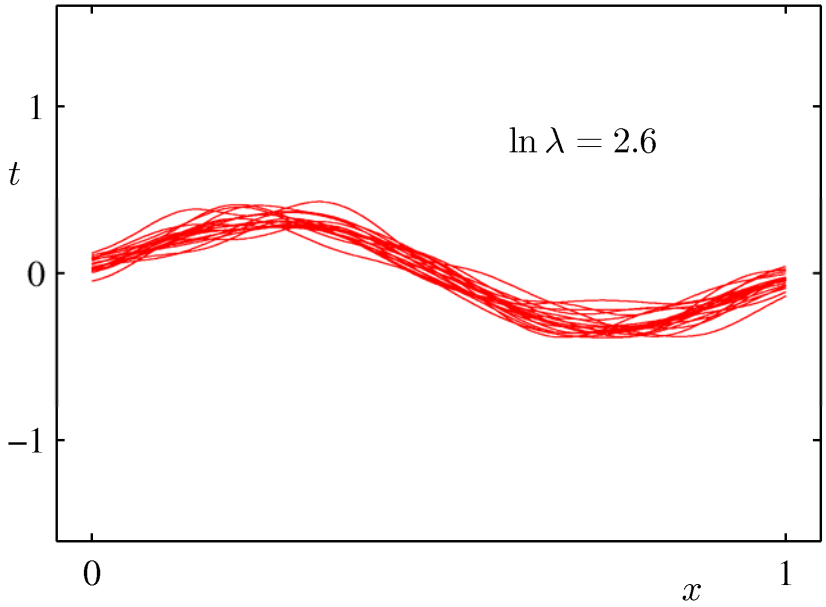
$$(\text{bias})^2 = \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

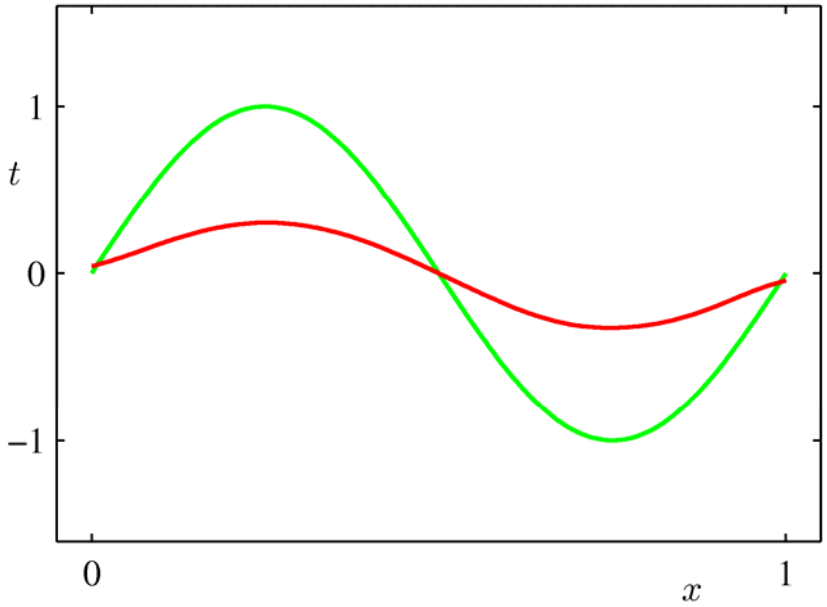
$$\text{variance} = \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

$$\text{noise} = \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x},t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

The Bias-Variance Decomposition (5)

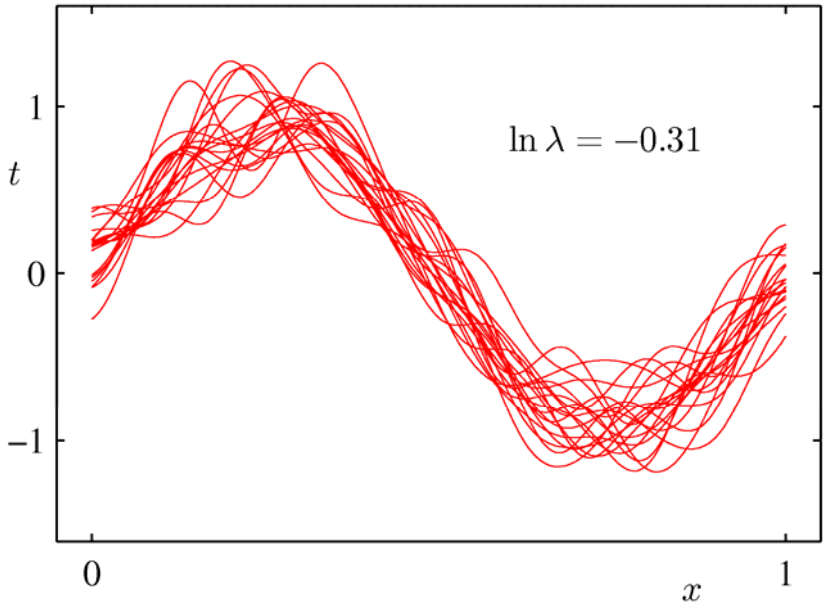
Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ

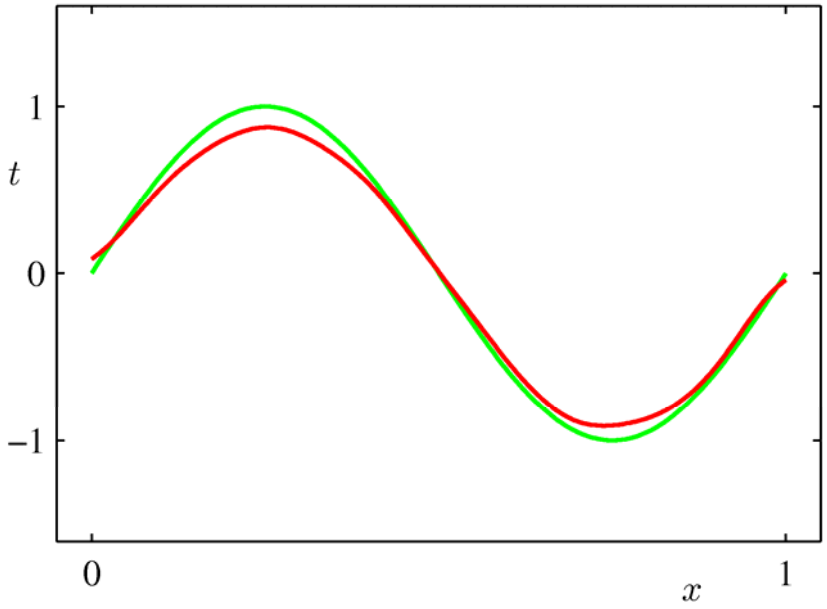




The Bias-Variance Decomposition (6)

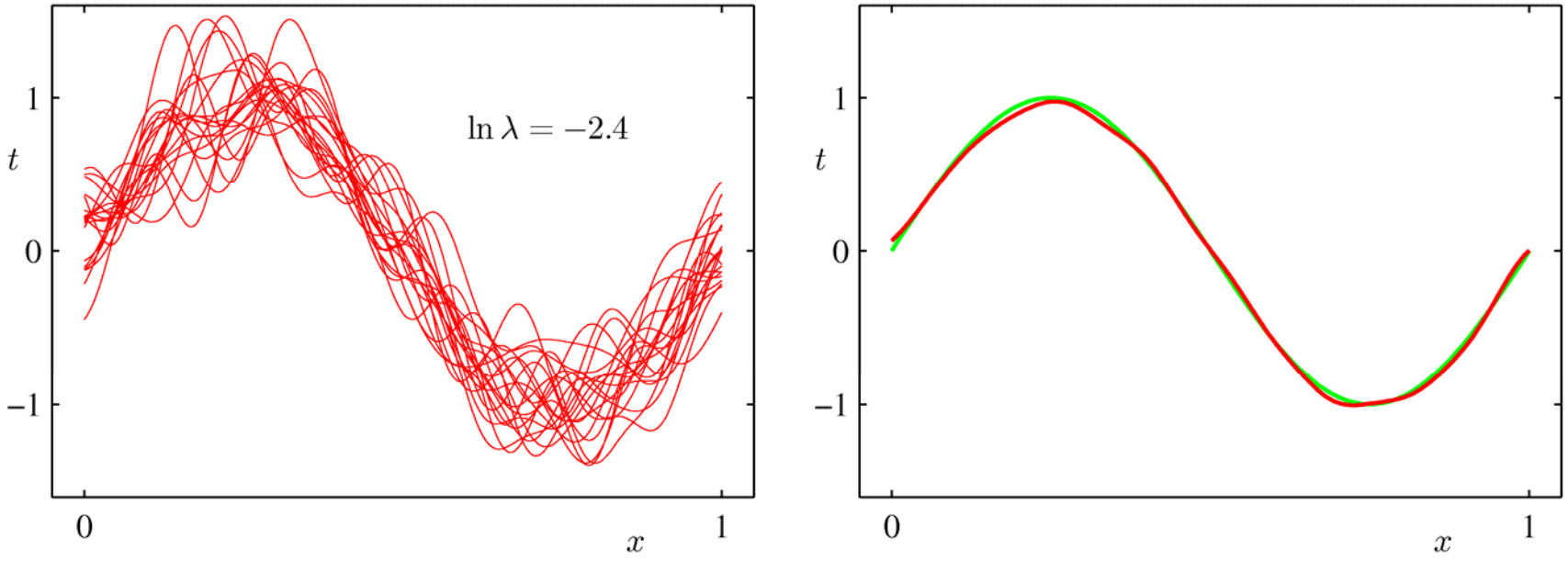
Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ



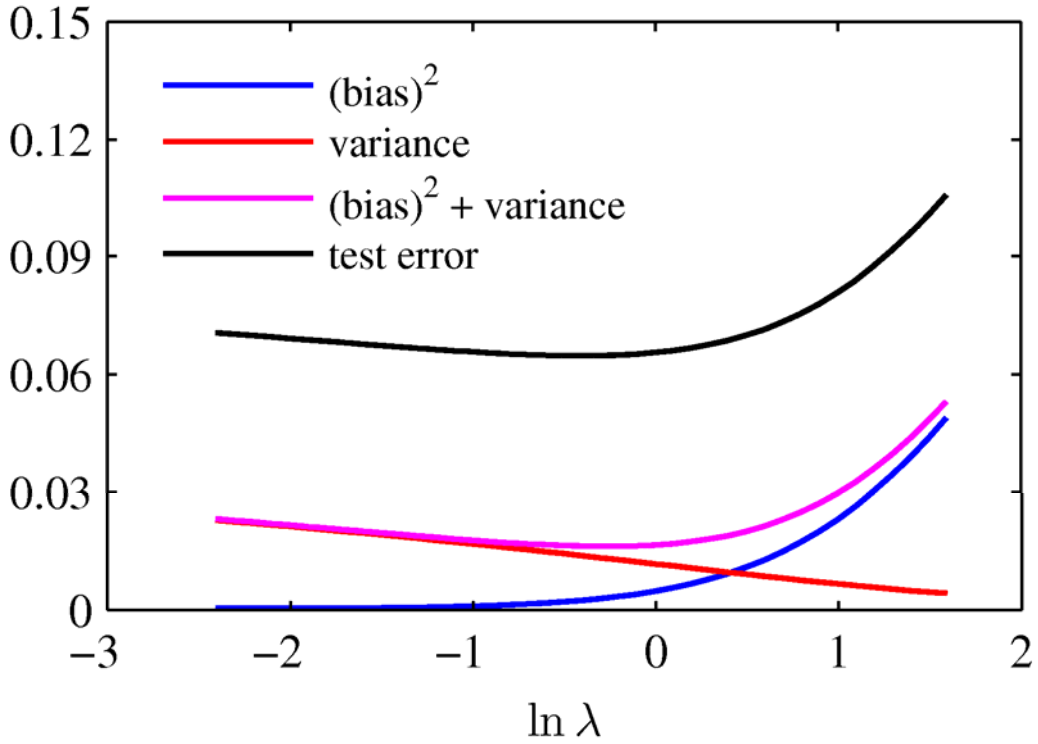


The Bias-Variance Decomposition (7)

Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ

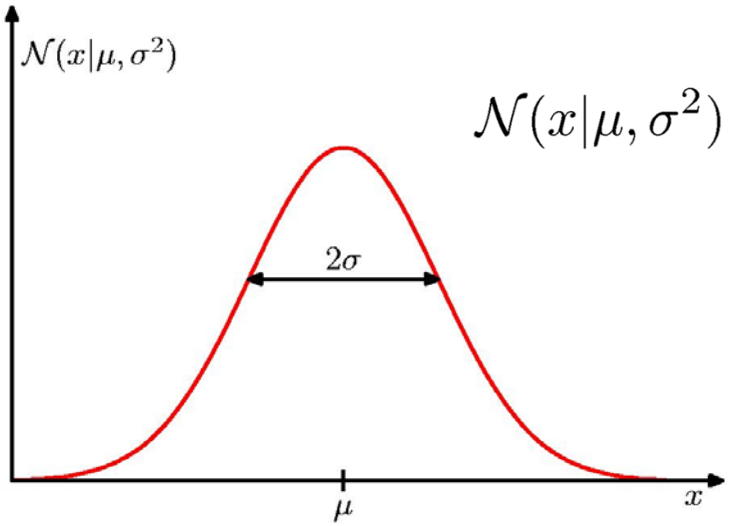


The Bias-Variance Trade-off



From these plots, we note that an over-regularized model (largeλ) will have a high bias, while an under-regularized model (small λ) will have a high variance.

Recall: The Gaussian Distribution (L2)

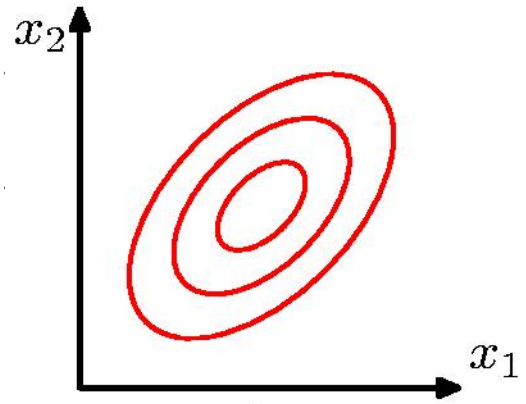


$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right)\,\mathrm{d}x = 1$$

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



Can also use precision $\beta = \frac{1}{\sigma^2}$, $\Lambda = \Sigma^{-1}$

Recall: Linear Gaussian Models (L2)

Given

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right) \\ p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}\left(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b}, \mathbf{L}^{-1}\right) \end{aligned}$$

we have

$$\begin{aligned} p(\mathbf{y}) &= \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}+\mathbf{b}, \mathbf{L}^{-1}+\mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}) \\ p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y}-\mathbf{b})+\boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) \end{aligned}$$

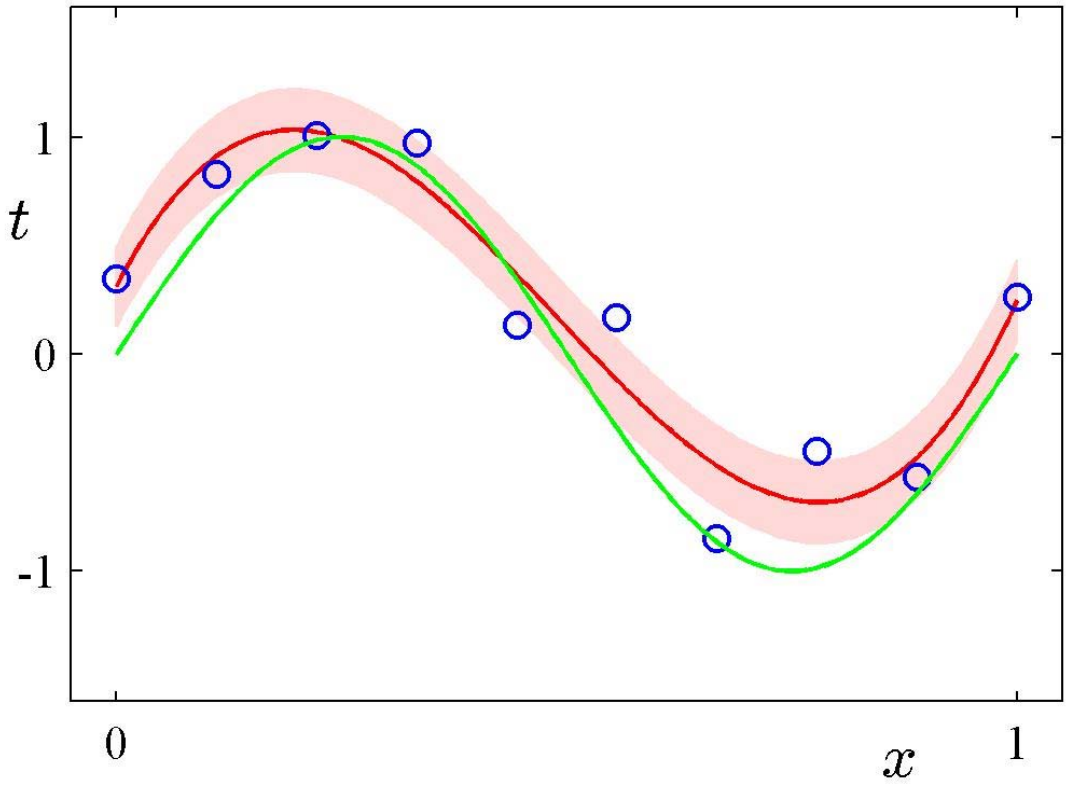
where

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}$$

Many applications for linear Gaussian models,
time series models - linear dynamical systems (Kalman filtering)

Predictive Distribution

$$p(t|x, \mathbf{w}_{\mathrm{ML}}, \beta_{\mathrm{ML}}) = \mathcal{N}\left(t|y(x, \mathbf{w}_{\mathrm{ML}}), \beta_{\mathrm{ML}}^{-1}\right)$$



Bayesian Regression (1)

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right\}$$

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}).$$

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha)$$

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

Determine $\mathbf{w}_{\mathrm{MAP}}$ by minimizing regularized sum-of-squares error, $\widetilde{E}(\mathbf{w})$ with regularizer corresponding to $\lambda = \alpha/\beta$

Bayesian Regression (2)

$$p(t|x,\mathbf{x},\mathbf{t}) = \int p(t|x,\mathbf{w})p(\mathbf{w}|\mathbf{x},\mathbf{t})\,\mathrm{d}\mathbf{w} = \mathcal{N}\left(t|m(x),s^2(x)\right)$$

$$m(x) = \beta\boldsymbol{\phi}(x)^{\mathrm{T}}\mathbf{S}\sum_{n=1}^{N}\boldsymbol{\phi}(x_n)t_n \qquad s^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^{\mathrm{T}}\mathbf{S}\boldsymbol{\phi}(x)$$

$$\mathbf{S}^{-1} = \alpha\mathbf{I} + \beta\sum_{n=1}^{N}\boldsymbol{\phi}(x_n)\boldsymbol{\phi}(x_n)^{\mathrm{T}} \qquad \boldsymbol{\phi}(x_n) = \left(x_n^0,\ldots,x_n^M\right)^{\mathrm{T}}$$

In compact form,
for $\mathbf{t} = (t_1, t_2 \ldots t_N)^T$

$$\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}$$

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

$\longrightarrow$

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

$$\mathbf{m}_N = \beta\mathbf{S}_N\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}$$

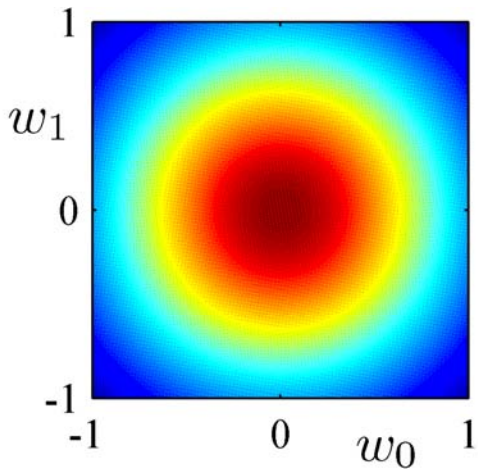
$$\mathbf{S}_N^{-1} = \alpha\mathbf{I} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}.$$

Bayesian Linear Regression (3)

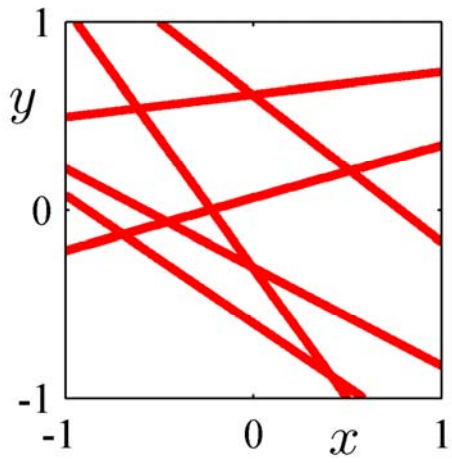
0 data points observed

Red lines: samples from the model prior

Prior



Data Space



Fit $y(x, w) = w_0 + w_1 x$, assume known $\beta = \left(\frac{1}{0.2}\right)^2 = 25$

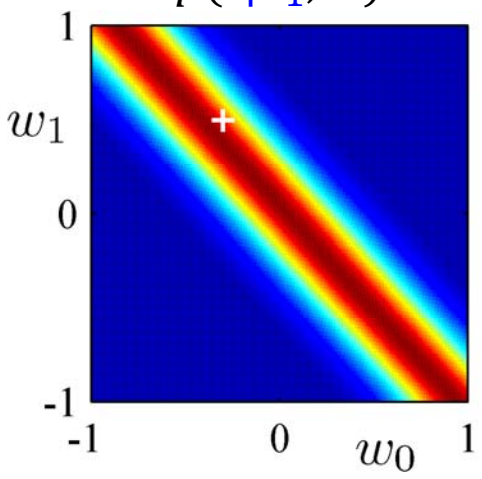
Ground truth $f(x, a) = a_0 + a_1\, x + n;\ a_0 = -0.3, a_1 = 0.5, U(x|-1,1),\ n \sim N(0,0.2)$

Bayesian Linear Regression (4)

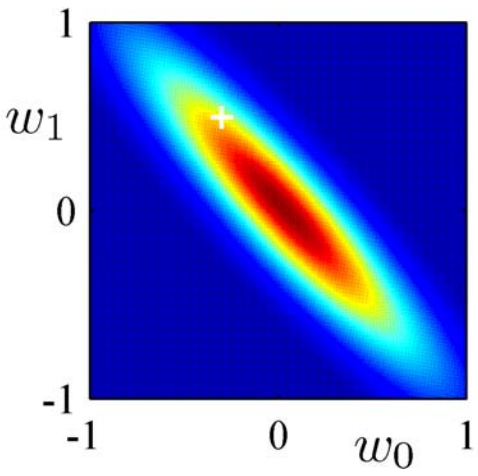
1 data point observed $\mathbf{x}_1$

Red lines: samples from the model posterior

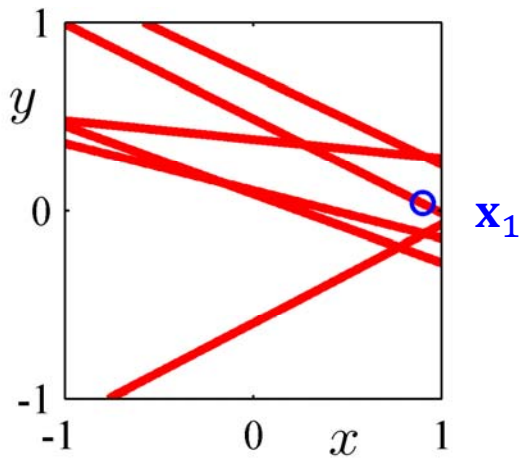
Likelihood $p(t|\mathbf{x}_1, \mathbf{w})$



Posterior



Data Space



Fit $y(x, w) = w_0 + w_1 x$, assume known $\beta = \left(\frac{1}{0.2}\right)^2 = 25$

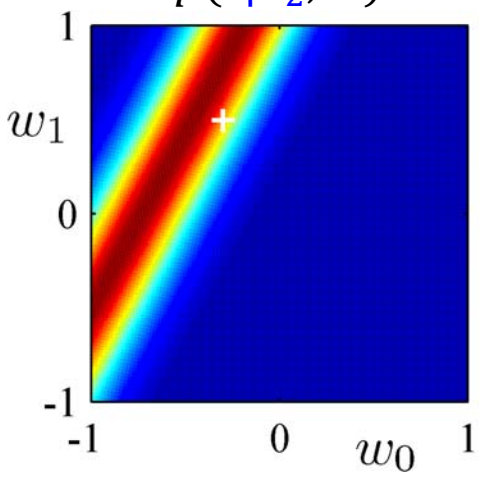
Ground truth $f(x, a) = a_0 + a_1\, x + n;\ a_0 = -0.3, a_1 = 0.5, U(x|-1,1),\ n \sim N(0,0.2)$

Bayesian Linear Regression (5)

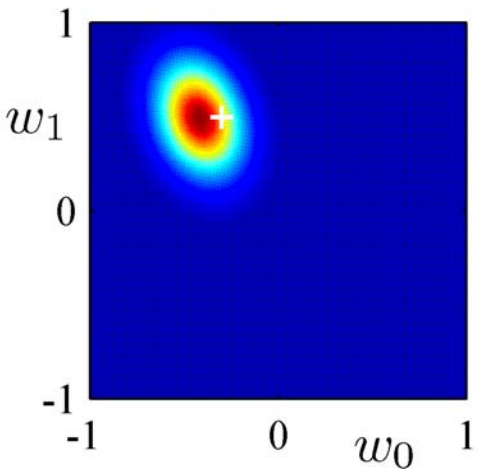
2 data points observed, second $\mathbf{x}_2$

Red lines: samples from the model posterior

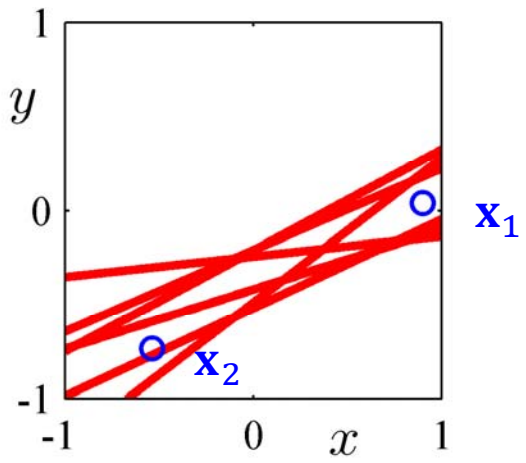
Likelihood $p(t|\mathbf{x}_2, \mathbf{w})$



Posterior



Data Space



Fit $y(x, w) = w_0 + w_1 x$, assume known $\beta = \left(\frac{1}{0.2}\right)^2 = 25$

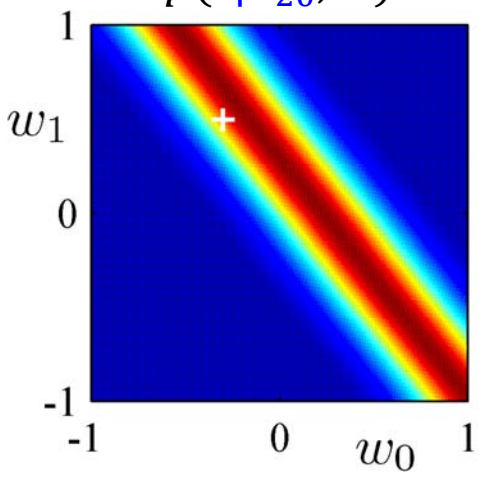
Ground truth $f(x, a) = a_0 + a_1\, x + n;\ a_0 = -0.3, a_1 = 0.5, U(x|-1,1),\ n \sim N(0,0.2)$

Bayesian Linear Regression (6)

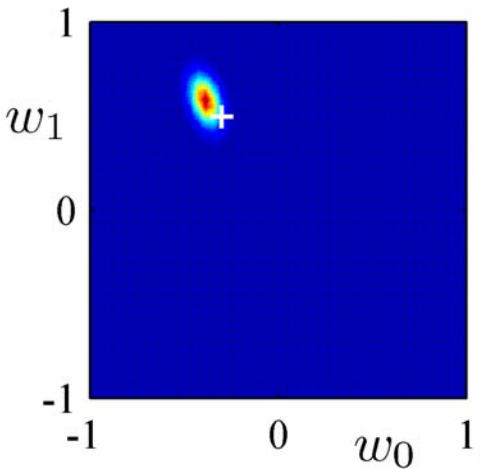
20 data points observed, last being $\mathbf{x}_{20}$

Red lines: samples from the model posterior

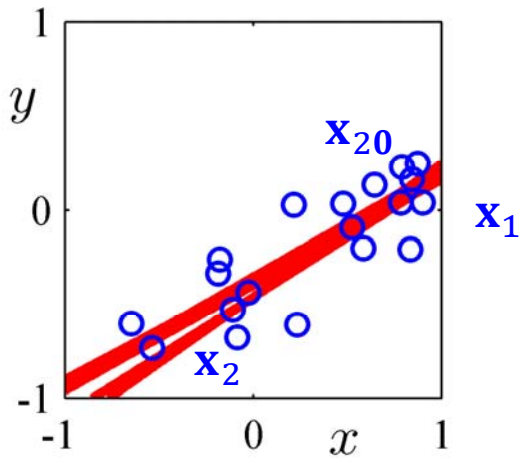
Likelihood $p(t|\mathbf{x}_{20}, \mathbf{w})$



Posterior



Data Space



Fit $y(x, w) = w_0 + w_1 x$, assume known $\beta = \left(\frac{1}{0.2}\right)^2 = 25$

Ground truth $f(x, a) = a_0 + a_1\, x + n;\ a_0 = -0.3, a_1 = 0.5, U(x|-1,1),\ n \sim N(0,0.2)$

Predictive Distribution (1)

Predict t for new values of $\mathbf{x}$ by integrating over $\mathbf{w}$ (input dependencies dropped):

$$\begin{aligned} p(t|\mathbf{t}, \alpha, \beta) &= \int p(t|\mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \alpha, \beta)\, \mathrm{d}\mathbf{w} \\ &= \mathcal{N}(t|\mathbf{m}_N^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}), \sigma_N^2(\mathbf{x})) \end{aligned}$$

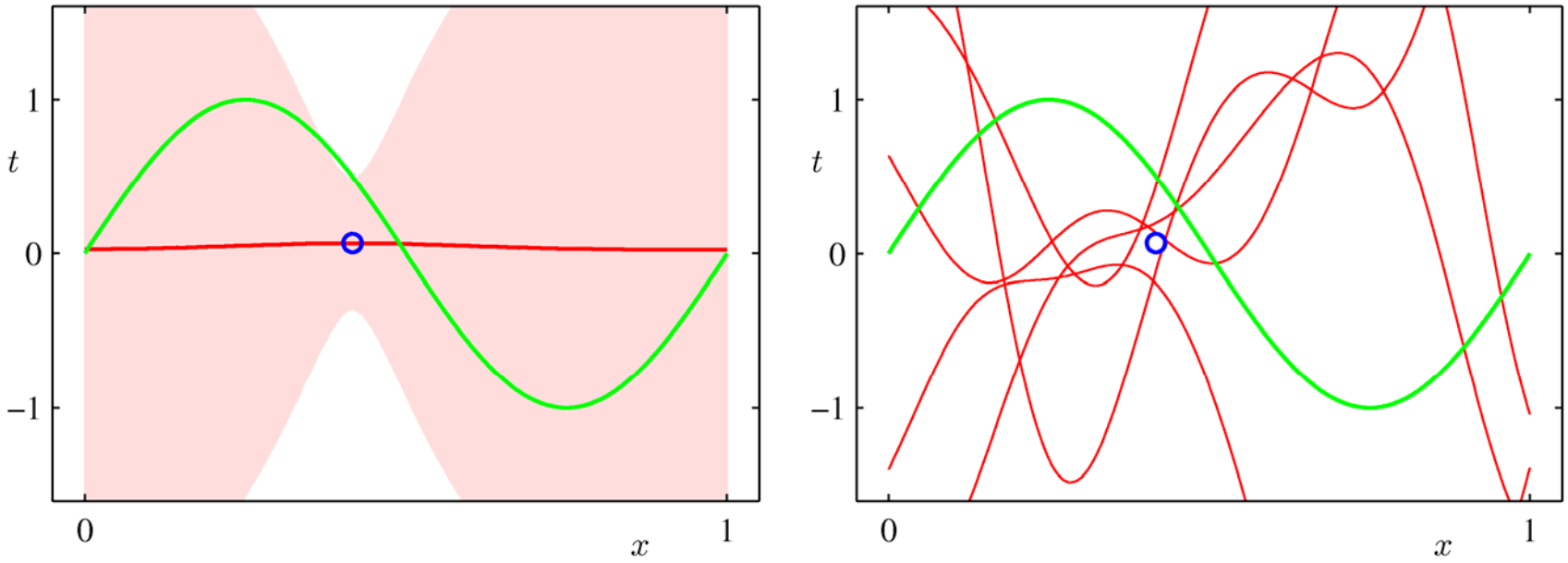
where

$$\sigma_N^2(\mathbf{x}) = \underbrace{\frac{1}{\beta}}_{\text{Noise in the data}} + \underbrace{\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x})}_{\text{Uncertainty for parameters } \mathbf{w}}.$$

Can prove:
$N \to \infty, \phi^T S_N \phi \to 0$

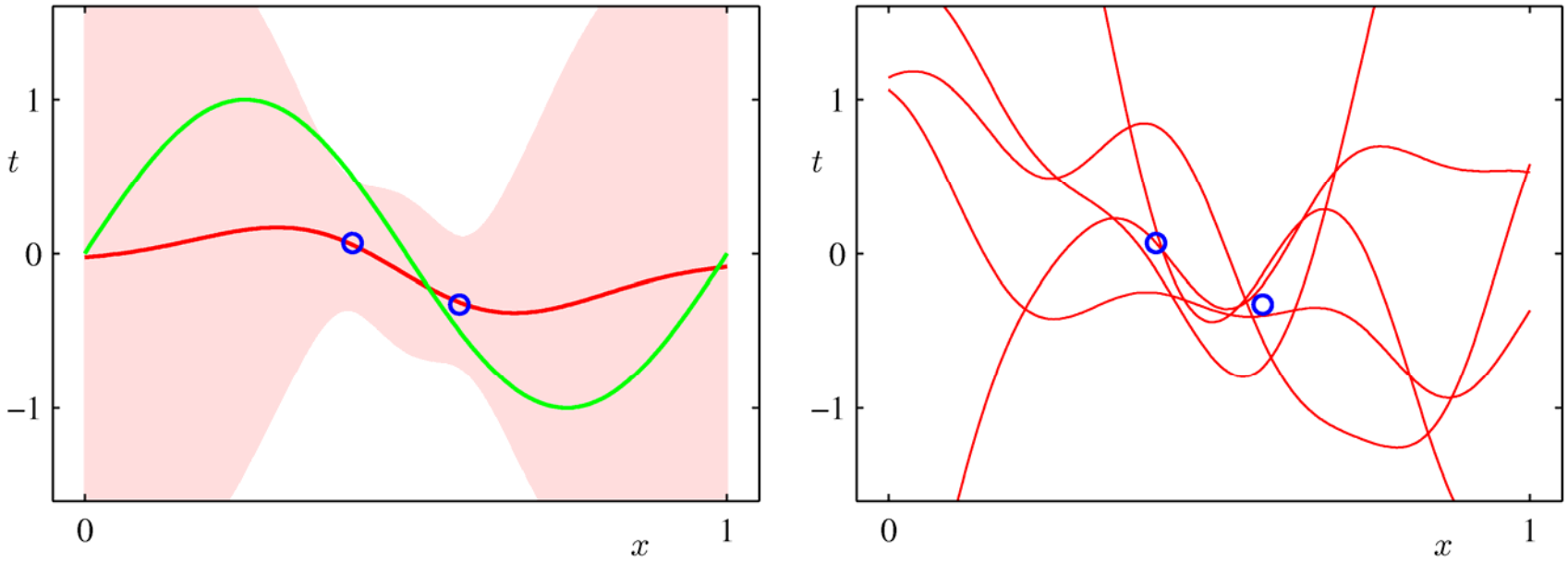
Predictive Distribution (2)

Example: Sinusoidal data, 9 Gaussian basis functions, 1 data point



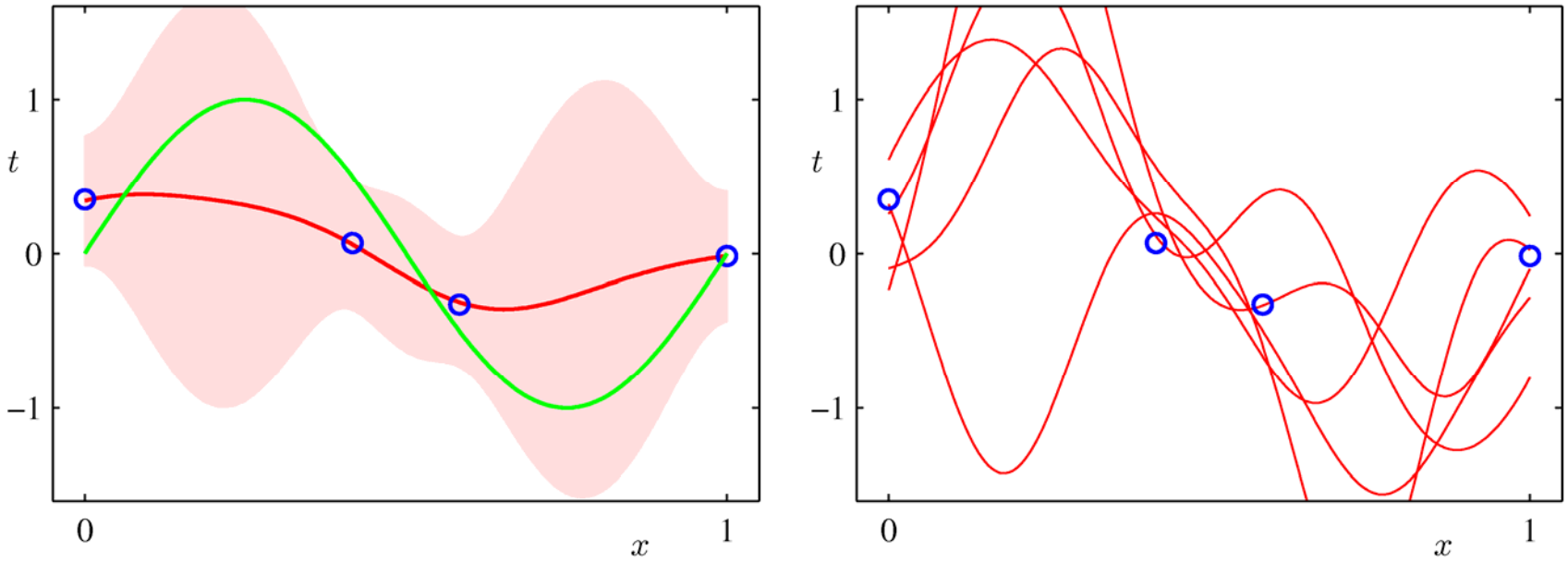
Predictive Distribution (3)

Example: Sinusoidal data, 9 Gaussian basis functions, 2 data points



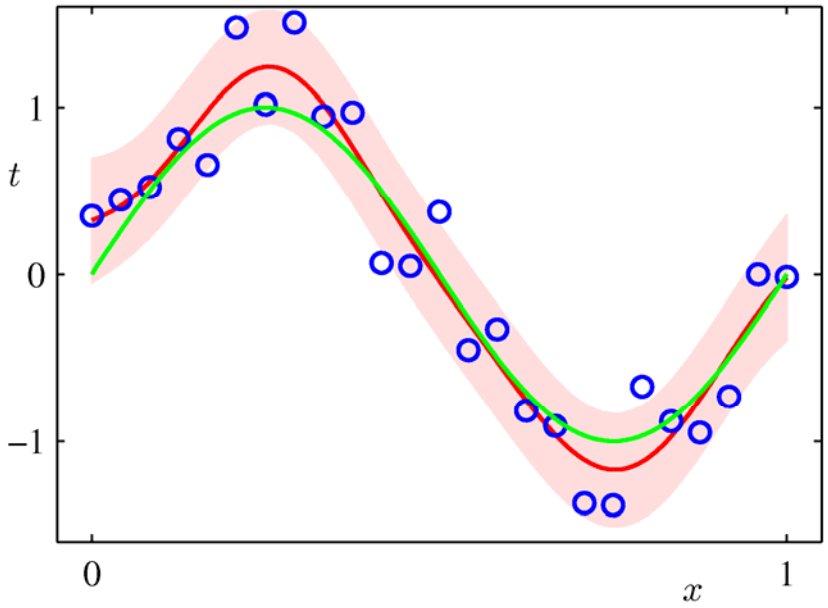
Predictive Distribution (4)

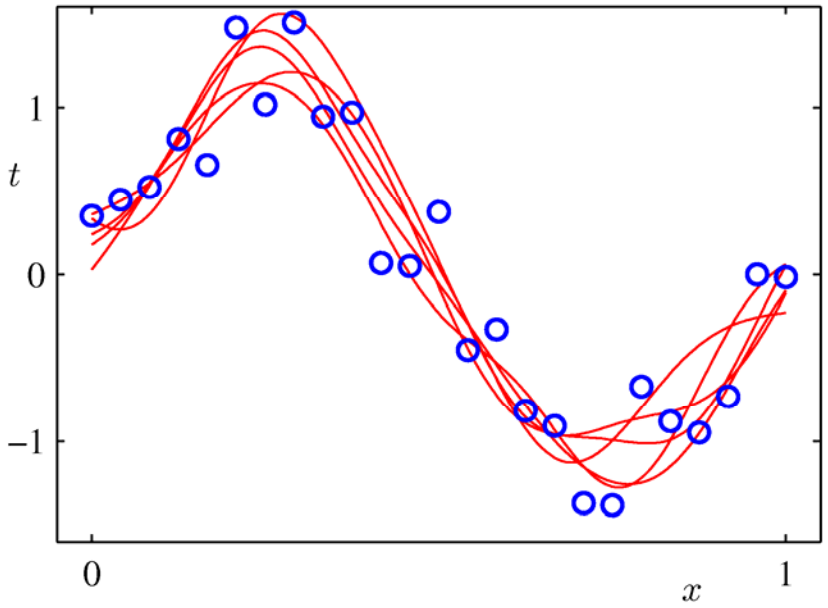
Example: Sinusoidal data, 9 Gaussian basis functions, 4 data points



Predictive Distribution (5)

Example: Sinusoidal data, 9 Gaussian basis functions, 25 data points





Equivalent Kernel (1)

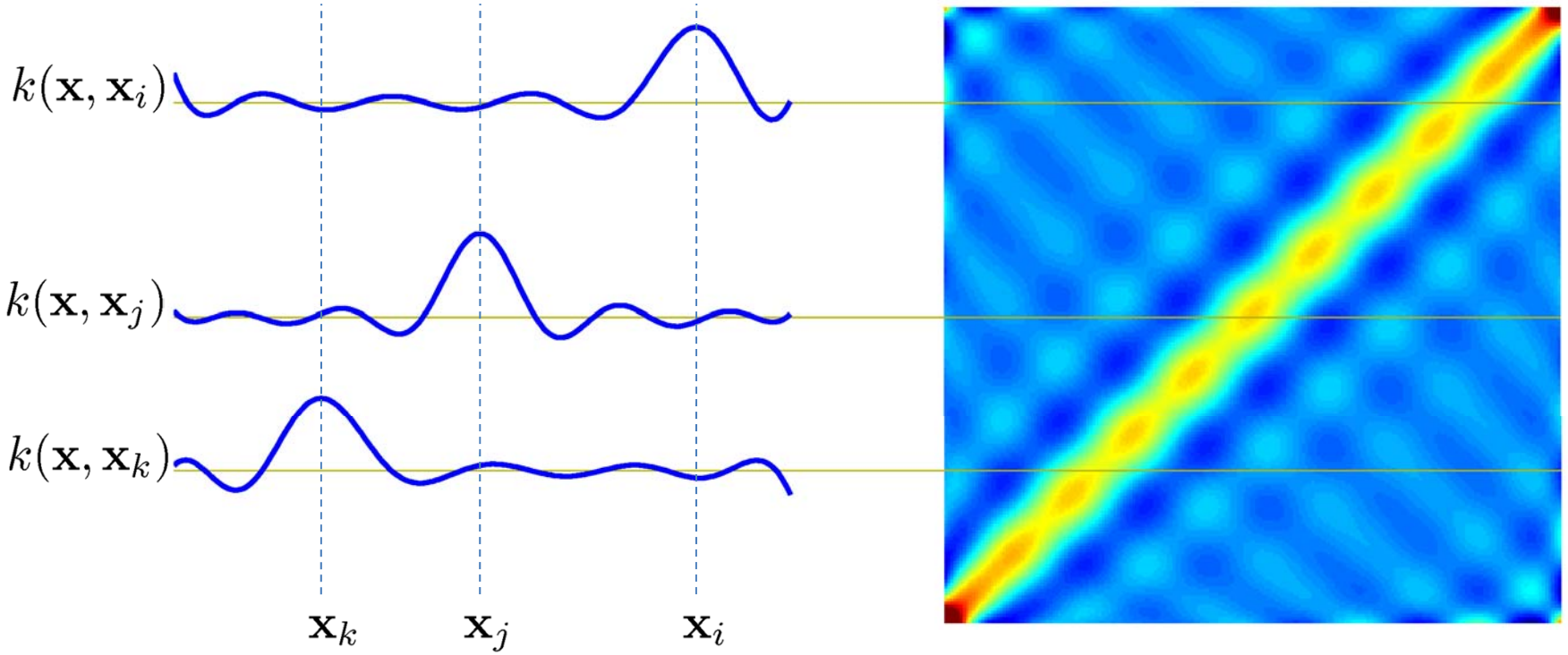
The predictive mean can be written

$$
\begin{aligned}
y(\mathbf{x}, \mathbf{m}_N) &= \mathbf{m}_N^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}) = \beta \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \\
&= \sum_{n=1}^{N} \beta \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}_n) t_n \\
&= \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) t_n .
\end{aligned}
$$

Equivalent kernel or *smoother matrix*.

This is a weighted sum of the training data target values, t_n.

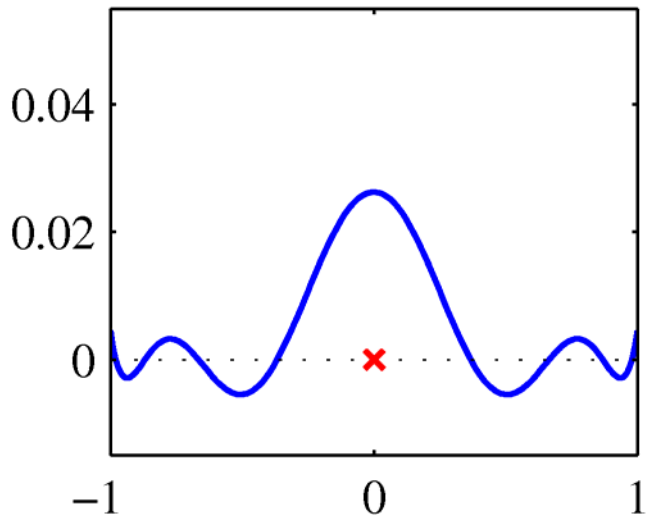
Equivalent Kernel (2)



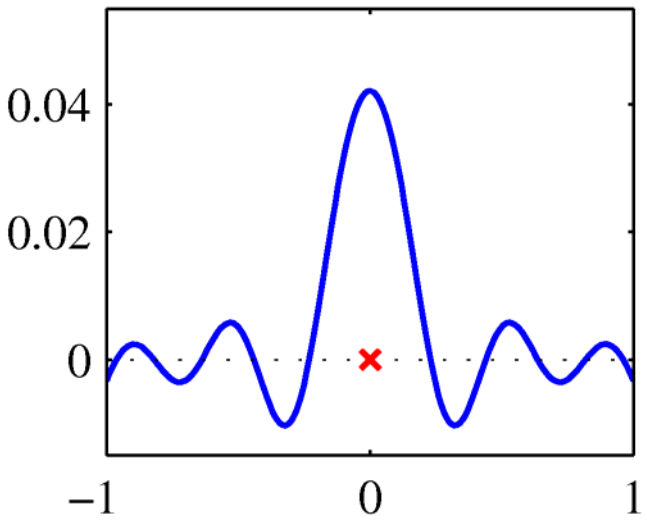
Weight of t_n depends on distance between $\mathbf{x}$ and $\mathbf{x_n}$; nearby $\mathbf{x_n}$ carry more weight.

Equivalent Kernel (3)

Non-local basis functions have local equivalent kernels:



Polynomial



Sigmoidal

Equivalent Kernel (4)

The kernel as a covariance function: consider

$$\begin{aligned}\mathrm{cov}[y(\mathbf{x}), y(\mathbf{x}')] &= \mathrm{cov}[\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{w}, \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}')] \\ &= \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}') = \beta^{-1}k(\mathbf{x}, \mathbf{x}').\end{aligned}$$

We can avoid the use of basis functions and define the kernel function directly, leading to *Gaussian Processes* (Chapter 6, textbook).

Equivalent Kernel (5)

$$\sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) = 1$$

for all values of $\mathbf{x}$; however, the equivalent kernel may be negative for some values of $\mathbf{x}$.

Like all kernel functions, the equivalent kernel can be expressed as an inner product:

$$k(\mathbf{x}, \mathbf{z}) = \boldsymbol{\psi}(\mathbf{x})^{\mathrm{T}} \boldsymbol{\psi}(\mathbf{z})$$

where $\boldsymbol{\psi}(\mathbf{x}) = \beta^{1/2} \mathbf{S}_N^{1/2} \boldsymbol{\phi}(\mathbf{x})$.

Bayesian Model Comparison

- We usually need to decide between many different models
  - Numbers of basis functions
  - Type of basis functions (polynomial, Gaussian)
  - Strengths and type of regularizers (quadratic, Lasso)
- The ***frequentist approach*** is to hold back a validation set and pick the model that does best on the validation data.
  - This gives less training data. We can use bootstrap to sample data with replacement, obtain different estimators and average them to obtain error bars
- The ***Bayesian alternative*** is to use all the data for training each model and to use **the evidence** to pick the best model, or to average over models
- The evidence is the **marginal likelihood** with the parameters integrated out

adapted from MLG@Toronto

Bayesian Model Comparison (1)

How do we choose the 'right' model?

Assume we want to compare models $M_i, i = 1, \ldots, L,$ using data D; this requires computing

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

Posterior — Prior — *Model evidence* or *marginal likelihood*

Bayes Factor: ratio of evidence for two models

$$\frac{p(\mathcal{D}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_j)}$$

Bayesian Model Comparison (2)

Having computed $p(M_i|D)$, we can compute the predictive (mixture) distribution

$$p(t|\mathbf{x}, \mathcal{D}) = \sum_{i=1}^{L} p(t|\mathbf{x}, \mathcal{M}_i, \mathcal{D}) p(\mathcal{M}_i|\mathcal{D}).$$

A simpler approximation, known as *model selection*, is to use the model with the highest evidence.

Bayesian Model Comparison (3)

For a model with parameters $\mathbf{w}$, we get the model evidence by marginalizing over $\mathbf{w}$

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w}, \mathcal{M}_i) p(\mathbf{w}|\mathcal{M}_i) \, \mathrm{d}\mathbf{w}.$$

Note that

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

From a sampling perspective, the marginal likelihood can be viewed as probability of generating the dataset *D* from a model whose parameters are sampled at random from the prior.

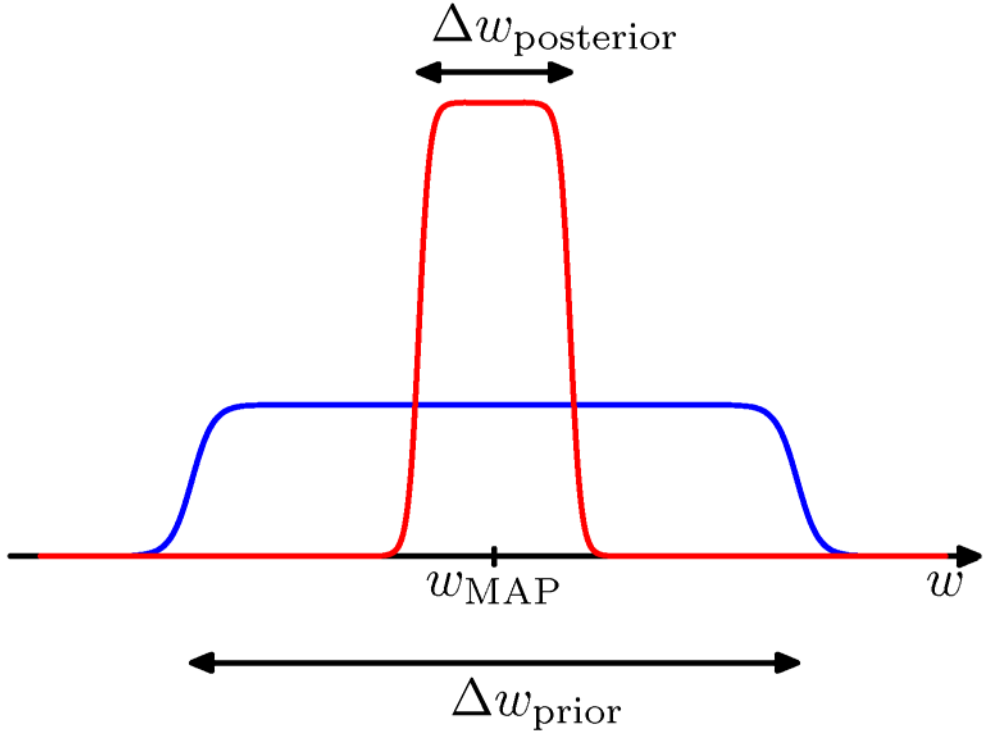
Bayesian Model Comparison (4)

For a given model with a single parameter, $\mathbf{w}$, consider the approximation

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w)\,\mathrm{d}w$$

$$\simeq \quad p(\mathcal{D}|w_{\mathrm{MAP}})\frac{\Delta w_{\mathrm{posterior}}}{\Delta w_{\mathrm{prior}}}$$

where the posterior is assumed to be sharply peaked.



Bayesian Model Comparison (5)

Taking logarithms, we obtain

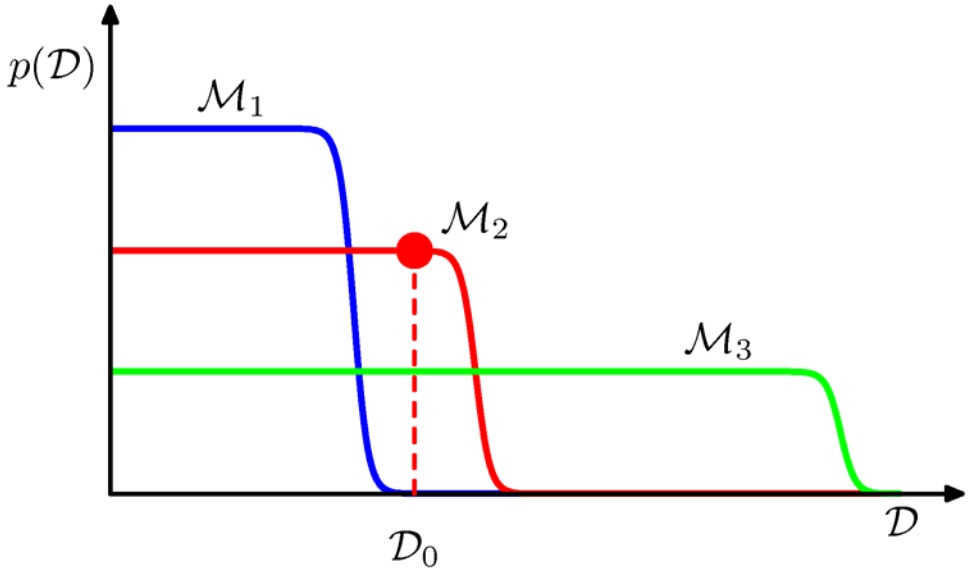
$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|w_{\mathrm{MAP}}) + \underbrace{\ln\left(\frac{\Delta w_{\mathrm{posterior}}}{\Delta w_{\mathrm{prior}}}\right)}_{\text{Negative}}.$$

With M parameters, all assumed to have the same ratio $\Delta w_{\mathrm{posterior}}/\Delta w_{\mathrm{prior}}$, we get

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\mathbf{w}_{\mathrm{MAP}}) + \underbrace{M \ln\left(\frac{\Delta w_{\mathrm{posterior}}}{\Delta w_{\mathrm{prior}}}\right)}_{\text{Negative and linear in } M}.$$

Bayesian Model Comparison (6)

Matching data and model complexity



- A simple model (e.g. M_1) has little variability, so it will generate datasets that are fairly similar to each other, with distribution confined narrowly on the D axis
- Complex models (e.g. M_3) can generate a large variety of datasets and assign a relatively low probability to any one of them

The Evidence Approximation (1)

The full Bayesian predictive distribution is given by

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

but this integral is intractable. Approximate with

$$p(t|\mathbf{t}) \simeq p\left(t|\mathbf{t},\widehat{\alpha},\widehat{\beta}\right) = \int p\left(t|\mathbf{w},\widehat{\beta}\right) p\left(\mathbf{w}|\mathbf{t},\widehat{\alpha},\widehat{\beta}\right)\,\mathrm{d}\mathbf{w}$$

where $\left(\widehat{\alpha},\widehat{\beta}\right)$ is the mode of $p(\alpha,\beta|\mathbf{t})$, which is assumed to be sharply peaked; a.k.a. *empirical Bayes, type II* or *generalized maximum likelihood,* or *evidence approximation*.

The Evidence Approximation (2)

From Bayes' theorem we have

$$p(\alpha, \beta|\mathbf{t}) \propto p(\mathbf{t}|\alpha, \beta)p(\alpha, \beta)$$

and if we assume $p(\alpha, \beta)$ to be flat we see that

$$\begin{aligned} p(\alpha, \beta|\mathbf{t}) &\propto p(\mathbf{t}|\alpha, \beta) \\ &= \int p(\mathbf{t}|\mathbf{w}, \beta)p(\mathbf{w}|\alpha)\,\mathrm{d}\mathbf{w}. \end{aligned}$$

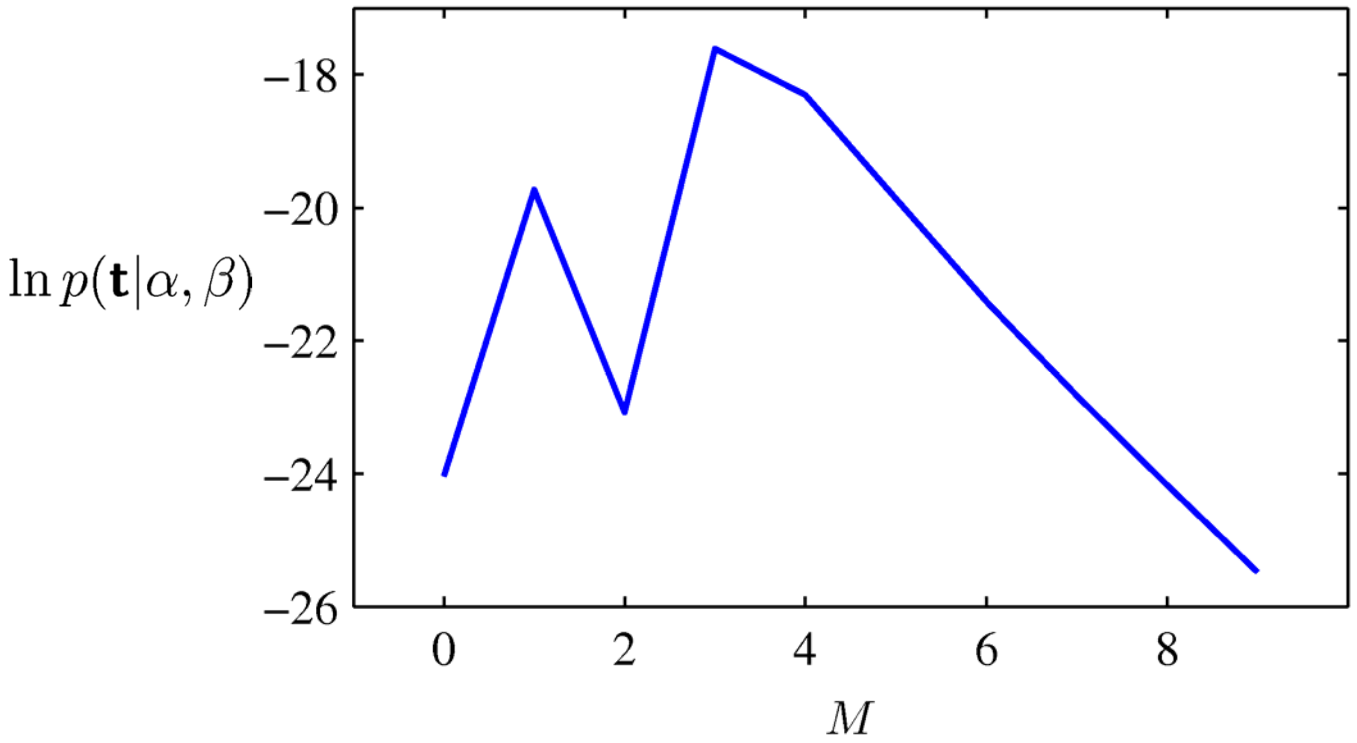
General results for Gaussian integrals give

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

The Evidence Approximation (3)

Example: sinusoidal data, M^{th} degree polynomial,

$\alpha = 5 \times 10^{-3}$



Readings

Bishop

Ch. 3, sections 3.1 – 3.4 (optional 3.5)