Bayesian Regression: Basis Functions, MLE & Regularized Least Squares, Multiple Outputs, Inference with σ^2 unknown, Zellner's g-Prior, Uninformative Priors

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, Indiana, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

September 24, 2018

Contents

Following closely:

- Chris Bishops' PRML book, Chapter 3
- Kevin Murphy, Machine Learning: A probabilistic Perspective, Chapter 7
- Regression using parametric discriminative models in pmtk3 (run TutRegr.m in Pmtk3)

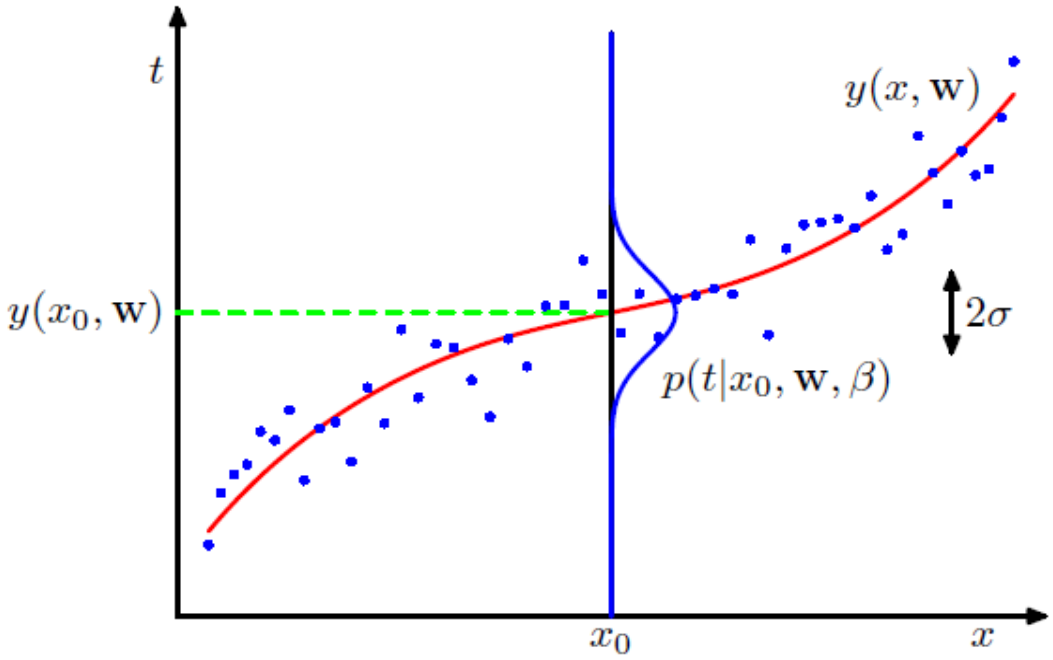
Linear Regression

- We already considered in an earlier lecture an example of linear regression – polynomial curve fitting.
- We are interested in a linear combination – regression – of a ``fixed set'' of nonlinear basis functions.
- Supervised learning: N observations $\{\boldsymbol{x}_n\}$ with corresponding target values $\{t_n\}$ are provided.
- The goal is to predict t for a new value $\boldsymbol{x}$.
- We construct a function such that $y(\boldsymbol{x})$ is a prediction of t.
- We follow a Bayesian perspective and model the predictive distribution $p(t|\boldsymbol{x})$.

Linear Regression

- From the conditional distribution $p(t|\boldsymbol{x})$, we can make point estimates of t for a given $\boldsymbol{x}$ by minimizing a `loss function'.

- For a quadratic loss function, the point estimate is the conditional mean $y(\boldsymbol{x}, \boldsymbol{w}) = \mathbb{E}[t|\boldsymbol{x}]$.



Linear Regression

- The simplest linear model for regression is one that involves a linear combination of the input variables

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D = \sum_{i=0}^{D} w_i x_i$$

where $\boldsymbol{x} = (x_1 \quad x_2 \quad . \quad . \quad x_D)^T$ and we have defined $x_0 = 1$.

- This is often simply known as linear regression. D is the input dimensionality.

Linear Basis Functions Models

- More generally:

$$y(\boldsymbol{x},\boldsymbol{w}) = w_0 + w_1\phi_1(\boldsymbol{x}) + \ldots + w_{M-1}\phi_{M-1}(\boldsymbol{x}) = \sum_{i=0}^{M-1} w_i\phi_i(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}), \quad \text{where}: \phi_0(\boldsymbol{x}) = 1$$

where $\phi_i(\boldsymbol{x})$ are known as basis functions and

$$\boldsymbol{\phi}(\boldsymbol{x}) = \left(\phi_0(\boldsymbol{x}) \quad \phi_1(\boldsymbol{x}) \quad . \quad . \quad \phi_{M-1}(\boldsymbol{x})\right)^T$$

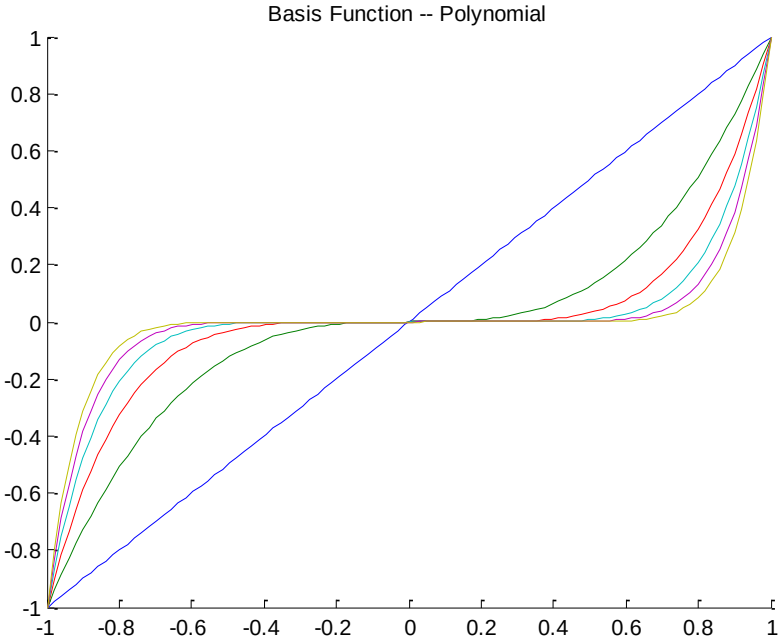
- The parameter w_0 allows for any fixed offset in the data and is called the ***bias parameter.***

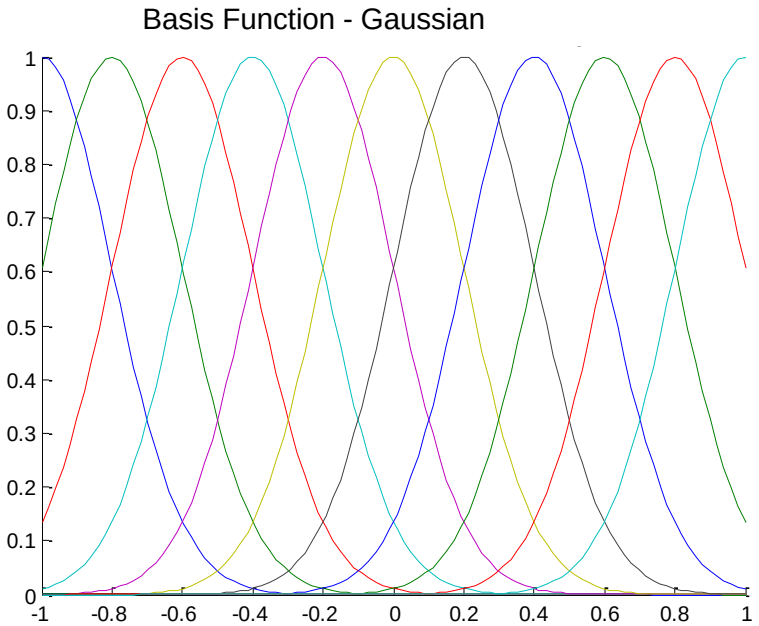
- For convenience, we define an additional dummy 'basis function' so that,

$$\phi_0(\boldsymbol{x}) = 1$$

- Often $\{\phi_i(\boldsymbol{x})\}$ represent features extracted from the data $\boldsymbol{x}$.

Polynomial and Gaussian Basis Functions





MatLab code

$$\mu = -1:0.2:1$$

$$s = 0.2$$

Polynomial basis functions (scalar input, global support):

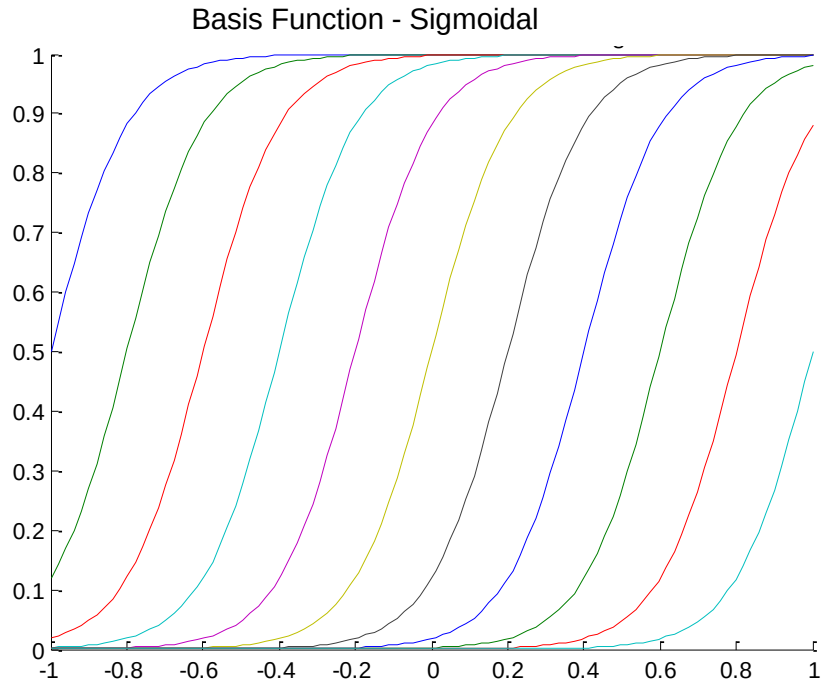
$$\phi_j(x) = x^j$$

Gaussian basis functions:

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

Logistic Sigmoidal Basis Functions

$$\mu = -1 : 0.2 : 1$$
$$s = 0.1$$



MatLab code

Sigmoidal basis functions:

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

$$\tanh(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}} = 2\sigma(2a) - 1$$

$\sigma(a)$: *logistic sigmoid function*

Sigmoidal and Tanh Basis Functions

- The sigmoidal and tanh basis functions are related:

$$\tanh(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}} = 2\sigma(2a) - 1 \quad \text{where} \quad \sigma(a) = \frac{1}{1 + e^{-a}}$$

- *A general linear combination of logistic sigmoidal functions is equivalent to a linear combination of* tanh *functions*:

$$y(x, w) = w_0 + \sum_{j=1}^{M} w_j \sigma\left(\frac{x - \mu_j}{s}\right) = w_0 + \sum_{j=1}^{M} w_j \sigma\left(2\frac{x - \mu_j}{2s}\right)$$

$$= w_0 + \sum_{j=1}^{M} w_j \frac{1 + \tanh\left(\frac{x - \mu_j}{2s}\right)}{2} = u_0 + \sum_{j=1}^{M} u_j \tanh\left(\frac{x - \mu_j}{2s}\right)$$

$$where: \; u_0 = w_0 + \frac{1}{2}\sum_{j=1}^{M} w_j, \; u_j = \frac{w_j}{2}$$

Choice of Basis Functions

- We are interested in functions of ***local support*** to explore adaptivity.
- Local support functions comprise a spectrum of different spatial frequencies.
- An example is wavelets that are local both spatially and in frequency.
  - They are however useful only when the input is defined on a lattice.

Maximum Likelihood and Least Squares

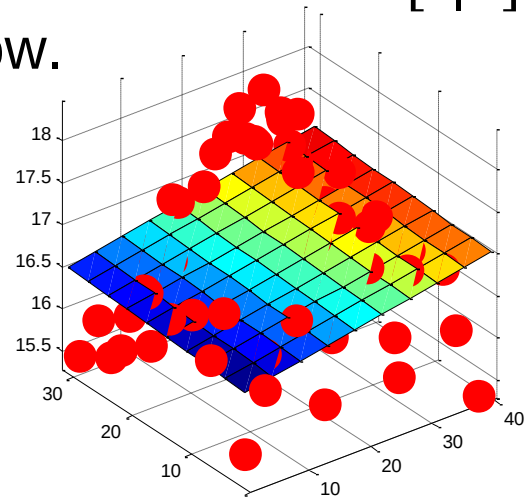
- Assume observations from a deterministic function with added Gaussian noise:

$$t = y(\boldsymbol{x}, \boldsymbol{w}) + \varepsilon \qquad \text{where} \qquad p(\varepsilon \mid \beta) = \mathcal{N}\left(\varepsilon \mid 0, \beta^{-1}\right)$$

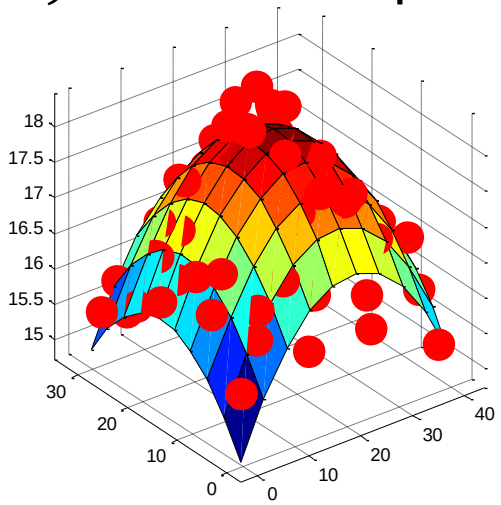
which is the same as saying,

$$p(t \mid \boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}\left(t \mid y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}\right)$$

- Here β is the precision. This is based on a squared loss function for which $\mathbb{E}[t|\boldsymbol{x}] = y(\boldsymbol{x}, \boldsymbol{w})$. A 2D example is shown below.



$$\mathbb{E}[t \mid \boldsymbol{x}] = w_0 + w_1 x_1 + w_2 x_2$$



$$\mathbb{E}[t \mid \boldsymbol{x}] = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2$$

Run surfaceFitDemo from PMTK3

Maximum Likelihood and Least Squares

- Given observed inputs, $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$, and targets $\boldsymbol{t} = \{t_1, ..., t_N\}^T$, we obtain the likelihood function

$$p(\boldsymbol{t} \mid \boldsymbol{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n \mid \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}\right)$$

- We often use the *log-likelihood*:

$$\ell(\boldsymbol{w}, \beta) = \log p(\mathcal{D} \mid \boldsymbol{\theta}) = \sum_{n=1}^{N} \log \mathcal{N}\left(t_n \mid \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}\right), \boldsymbol{\theta} = (\boldsymbol{w}, \beta)$$

- Instead of maximizing the log-likelihood, one equivalently can minimize *the negative log-likelihood (NLL)*

$$NLL(\boldsymbol{w}, \beta) = -\sum_{n=1}^{N} \log \mathcal{N}\left(t_n \mid \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}\right)$$

Maximum Likelihood and Least Squares

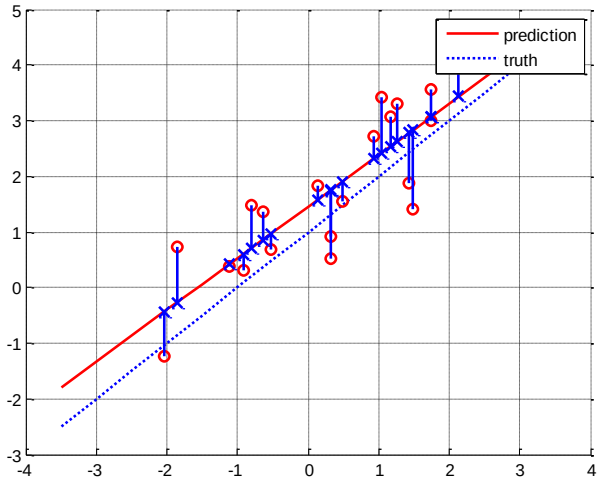
- Taking the log of the likelihood, we obtain:

$$\ln p(\boldsymbol{t} \mid \boldsymbol{w}, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}\left(t_n \mid \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}\right) = \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(\boldsymbol{w})$$

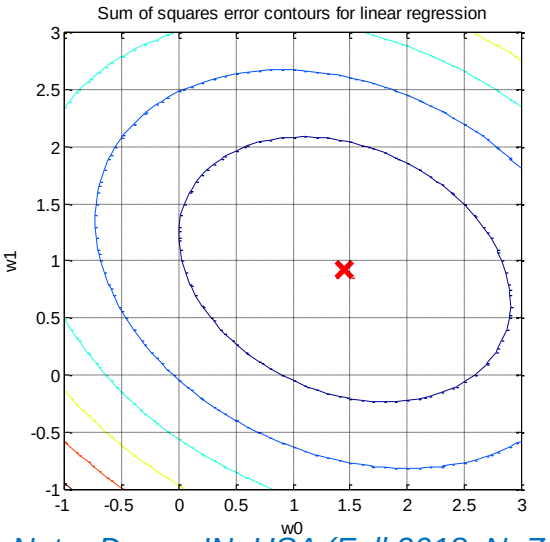
where with $E_D(\boldsymbol{w})$ we have denoted:

$$E_D(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} \left(t_n - \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n)\right)^2, \; RSS = \sum_{n=1}^{N} \left(t_n - \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n)\right)^2, \; MSE = RSS / N$$

- RSS is often known as the *residual sum of squares* or *sum of squared errors (SSE)* and MSE *is the mean squared error*.
- Computing $\boldsymbol{w}$ via *MLE* is the same as *Least Squares*.



Run residualsDemo from PMTK3



The NLL is a quadratic bowl with a unique minimum (the MLE estimate)

Run contoursSSEdemo, from PMTK3

Maximum Likelihood and Least Squares

$$\ln p(\boldsymbol{t} \mid \boldsymbol{w}, \beta) = \frac{N}{2} \ln \beta - \frac{N}{2} \ln (2\pi) - \beta E_D(\boldsymbol{w}), \; E_D(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} \left(t_n - \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n) \right)^2$$

- Setting the gradient of the log-likelihood (written here as a row vector) wrt $\boldsymbol{w}$ equal to zero:

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

- This equation can be solved for $\boldsymbol{w}$.

Maximum Likelihood and Least Squares

$$\boldsymbol{w}^T \sum_{n=1}^{N} \boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_n)^T = \sum_{n=1}^{N} t_n \boldsymbol{\phi}(\boldsymbol{x}_n)^T \Rightarrow \boldsymbol{w}^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} = \left(\boldsymbol{\Phi}^T \boldsymbol{t}\right)^T \Rightarrow \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{w} = \boldsymbol{\Phi}^T \boldsymbol{t}$$

- We obtain (normal equation; ordinary least squares solution):

$$\boldsymbol{w}_{ML} = \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T \boldsymbol{t} \equiv \boldsymbol{\Phi}^{\dagger} \boldsymbol{t}, \quad \boldsymbol{\Phi}^{\dagger} : \text{Moore-Penrose pseudo-inverse}$$

where we have defined:

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_1) \\ \boldsymbol{\phi}^T(\boldsymbol{x}_2) \\ \vdots \\ \boldsymbol{\phi}^T(\boldsymbol{x}_N) \end{pmatrix} = \begin{pmatrix} \phi_0(\boldsymbol{x}_1) & \phi_1(\boldsymbol{x}_1) & .. & \phi_{M-1}(\boldsymbol{x}_1) \\ \phi_0(\boldsymbol{x}_2) & \phi_1(\boldsymbol{x}_2) & .. & \phi_{M-1}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_0(\boldsymbol{x}_N) & \phi_1(\boldsymbol{x}_N) & .. & \phi_{M-1}(\boldsymbol{x}_N) \end{pmatrix}, \quad \boldsymbol{\Phi}^T = \begin{pmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1) & \boldsymbol{\phi}(\boldsymbol{x}_2) & .. & \boldsymbol{\phi}(\boldsymbol{x}_N) \end{pmatrix}$$

$$\boldsymbol{\Phi}^T = \begin{pmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & .. & \boldsymbol{\phi}_N \end{pmatrix}$$

$$\boldsymbol{\phi}_i(\boldsymbol{x}) = \begin{pmatrix} \phi_0(\boldsymbol{x}_i) & \phi_1(\boldsymbol{x}_i) & . & . & \phi_{M-1}(\boldsymbol{x}_i) \end{pmatrix}^T$$

- Note that indeed:

$$\boldsymbol{\Phi}^T \boldsymbol{\Phi} = \sum_{n=1}^{N} \boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_n)^T, \quad \boldsymbol{\Phi}^T \boldsymbol{t} = \sum_{n=1}^{N} t_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

Maximum Likelihood and Least Squares

$$\ln p\left(\boldsymbol{t} \mid \boldsymbol{w}, \beta\right) = \sum_{n=1}^{N} \ln \mathcal{N}\left(t_n \mid \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}\right) = \frac{N}{2} \ln \beta - \frac{N}{2} \ln\left(2\pi\right) - \beta E_D(\boldsymbol{w})$$

- Taking now the log of the likelihood with respect to β gives:

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

- So the MLE variance β_{ML} is equal to the residual variance of the target values around the regression function.

Computing the Bias Parameter

- If we make the bias parameter w_0 explicit, then the error function becomes

$$E_D(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - w_0 - \sum_{j=1}^{M-1} w_j \phi_j(\boldsymbol{x}_n) \right)^2$$

- Setting the derivative with respect to w_0 equal to zero, and solving for w_0, we obtain

$$w_0 = \frac{1}{N}\sum_{n=1}^{N} t_n - \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{M-1} w_j \phi_j(\boldsymbol{x}_n) \Rightarrow$$

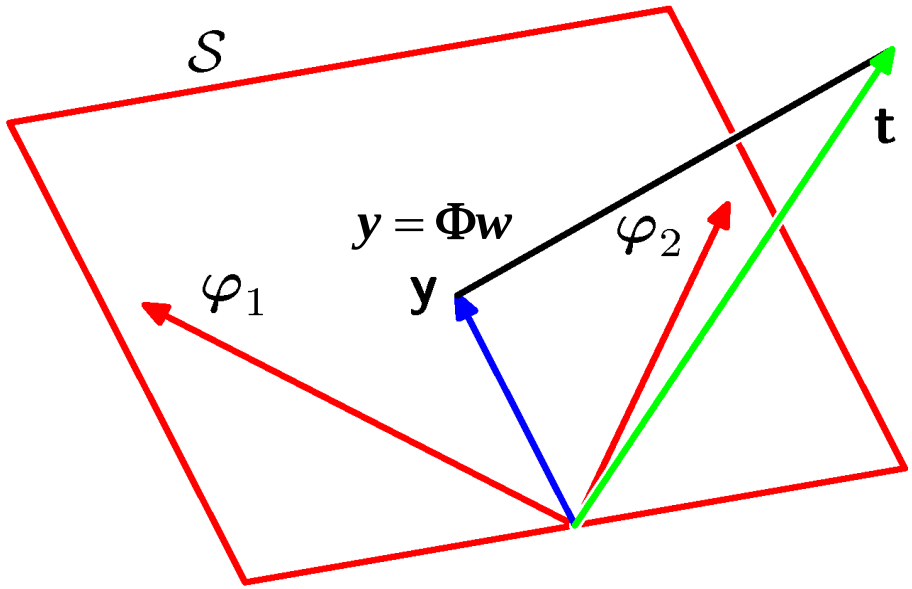
$$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j \bar{\phi}_j, \quad \bar{t} = \frac{1}{N}\sum_{n=1}^{N} t_n, \quad \bar{\phi}_j = \frac{1}{N}\sum_{n=1}^{N} \phi_j(\boldsymbol{x}_n)$$

- The bias parameter w_0 compensates for the difference between the averages of the target values and the weighted sum of the averages of the basis function values.

Geometry of Least Squares

- We look for a geometrical interpretation of the least-squares solution in an N-dimensional space. $\boldsymbol{t}$ is a vector in that space with components $t_1, \ldots, t_N$ $(N > M)$.
- The least-squares regression function is obtained by finding the orthogonal projection of the data vector $\boldsymbol{t}$ onto the subspace spanned by the basis functions $\boldsymbol{\varphi}_j(\boldsymbol{x})$.
- Note $\boldsymbol{\varphi}_j$ is here the j-th column of $\boldsymbol{\Phi}$.

$$\boldsymbol{\varphi}_j \equiv \left(\phi_j(\boldsymbol{x}_1), \ldots, \phi_j(\boldsymbol{x}_N)\right)^T$$



$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\varphi}_0 & \boldsymbol{\varphi}_1 & \cdots & \boldsymbol{\varphi}_{M-1} \end{pmatrix} = \begin{pmatrix} \phi_0(\boldsymbol{x}_1) & \phi_1(\boldsymbol{x}_1) & .. & \phi_{M-1}(\boldsymbol{x}_1) \\ \phi_0(\boldsymbol{x}_2) & \phi_1(\boldsymbol{x}_2) & .. & \phi_{M-1}(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_0(\boldsymbol{x}_N) & \phi_1(\boldsymbol{x}_N) & .. & \phi_{M-1}(\boldsymbol{x}_N) \end{pmatrix}$$

Geometry of Least Squares

- We are looking for $\boldsymbol{w}$ such that the projection error

$$\boldsymbol{t} - \boldsymbol{y} = \boldsymbol{t} - \boldsymbol{\Phi w}$$

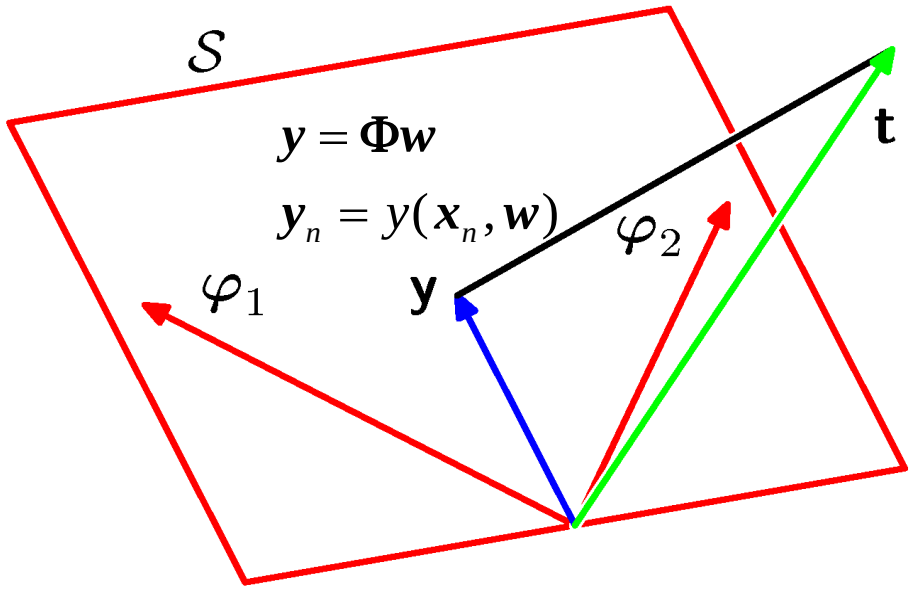
is orthogonal to the basis $\boldsymbol{\varphi}_j$, i.e. such that:

$$\boldsymbol{\Phi}^T \left(\boldsymbol{t} - \boldsymbol{\Phi w} \right) = 0$$

- These are the normal equations we derived earlier.

$$\boldsymbol{\Phi}^T = \begin{pmatrix} \boldsymbol{\varphi}_0^T \\ \boldsymbol{\varphi}_1^T \\ \vdots \\ \boldsymbol{\varphi}_{M-1}^T \end{pmatrix}$$

$$\boldsymbol{\varphi}_j(\boldsymbol{x}) = \left(\phi_j(\boldsymbol{x}_1), \ldots, \phi_j(\boldsymbol{x}_N) \right)^T$$



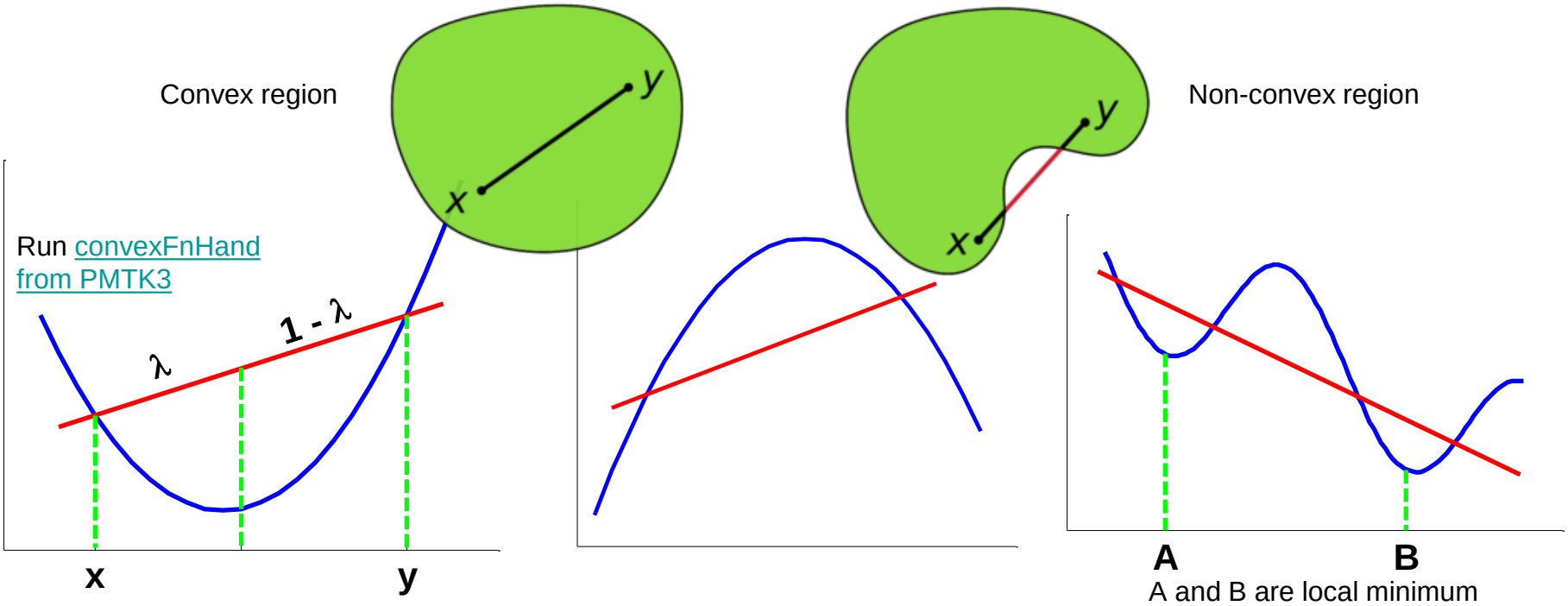
$\mathcal{S}$: M −dimensional subspace spanned by $\boldsymbol{\varphi}_j(\boldsymbol{x})$

$$\boldsymbol{\Phi} = \left[\boldsymbol{\varphi}_0(\boldsymbol{x}) \quad \boldsymbol{\varphi}_1(\boldsymbol{x}) \quad .. \quad \boldsymbol{\varphi}_{M-1}(\boldsymbol{x}) \right]$$

Convexity of the NLL

- *Convexity of the NLL (positive definite Hessian)* leads to a unique globally optimal MLE.

- Some models of interest don't have concave likelihoods and locally optimal MLE estimates are needed.



Sequential Learning: LMS Algorithm

- If the data set is large, we use sequential (on-line) algorithms
- We apply the technique of ***stochastic (sequential) gradient descent***.
- If the error function comprises a sum over data points $E = \sum_n E_n$

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

then after presentation of pattern n, the stochastic gradient descent algorithm updates the parameter vector $\boldsymbol{w}$ using

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} - \eta\nabla E_n = \boldsymbol{w}^{(\tau)} + \eta\underbrace{\left(t_n - \boldsymbol{w}^{(\tau)^T}\boldsymbol{\phi}(\boldsymbol{x}_n)\right)}_{-\nabla E_n}\boldsymbol{\phi}(\boldsymbol{x}_n)$$

τ is the iteration number & η the learning rate parameter.

- This is known as ***least-mean-squares*** *or the* ***LMS algorithm***.

Robust Linear Regression

- Using a Gaussian distribution for the noise,

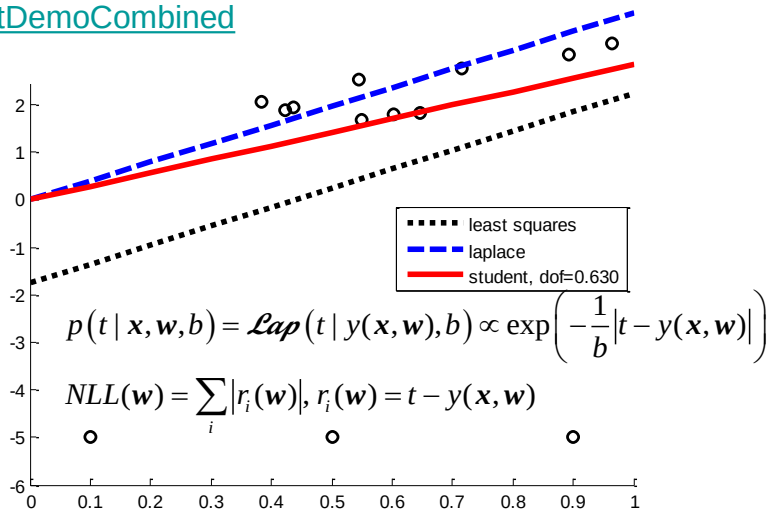
$$t = y(\boldsymbol{x}, \boldsymbol{w}) + \varepsilon,\ \varepsilon \sim \mathcal{N}\left(\varepsilon \mid 0, \beta^{-1}\right)$$

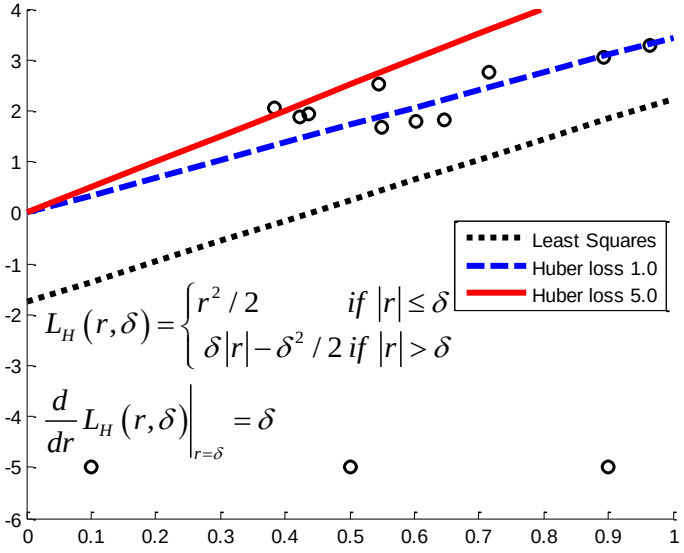
  can result in poor fit especially if we have outliers in the data.

- Squared error penalizes deviations quadratically, so points far from the line have more affect on the fit than points near the line.

- *To achieve robustness to outliers one can replace the Gaussian with a distribution that has heavy tails (e.g. the Laplace distribution).* Such a distribution assigns higher likelihood to outliers, without having to perturb the regression line to "explain" them.

Run linregRobustDemoCombined from PMTK3





Robust Linear Regression

- Using the Laplace distribution leads to L_1 error norm (non-linear objective function) that is difficult to optimize.

- A solution is to transform the problem (by increasing its dimension to $2N + M$) to a *linear programming problem*.

$$r_i \triangleq r_i^+ - r_i^-$$

$$\min_{\boldsymbol{w}, r_i^+, r_i^-} \sum_i \left(r_i^+ + r_i^- \right) \quad s.t. \;\; r_i^+ \geq 0, r_i^- \geq 0, \boldsymbol{w}^T \boldsymbol{x}_i + r_i^+ - r_i^- = t_i$$

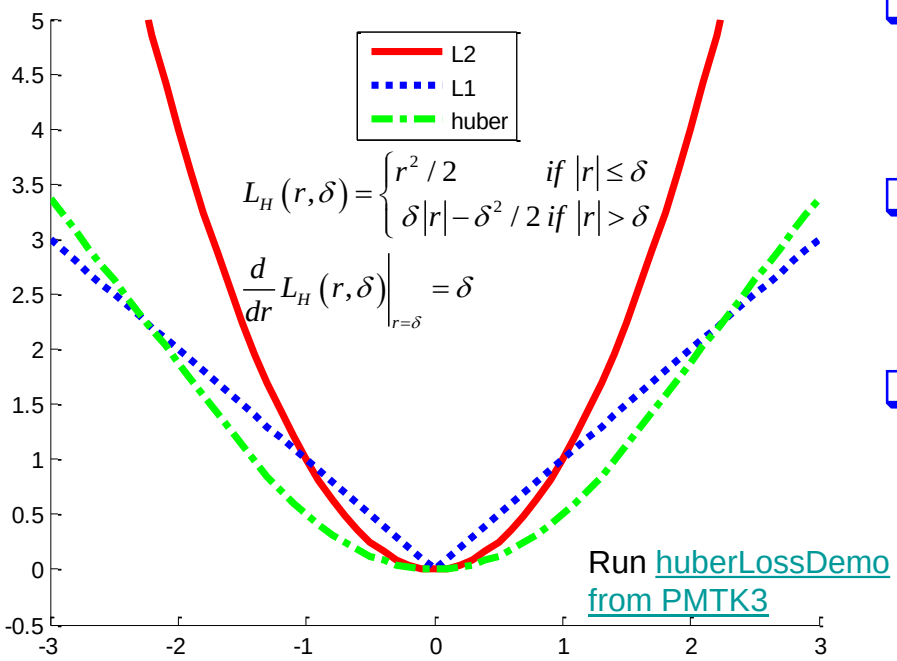
- Note that with our definition above:

$$r_i^+ = \frac{1}{2}(r_i + |r_i|) = \begin{cases} r_i \; if \;\; r_i \geq 0 \\ 0 \; if \;\; r_i < 0 \end{cases}, \quad r_i^- = \frac{1}{2}(|r_i| - r_i) = \begin{cases} 0 \quad if \;\; r_i > 0 \\ -r_i \; if \;\; r_i \leq 0 \end{cases}$$

$$|r_i| = r_i^+ + r_i^-$$

- Boyd, S. and L. Vandenberghe (2004). *Convex optimization*. Cambridge

Huber Loss Function



Run huberLossDemo from PMTK3

- *This is equivalent to L_2 for errors that are smaller than δ, and is equivalent to L_1 for larger errors.*
- This loss function is everywhere differentiable, using the fact that $d/dr\,|r| = sign(r)\ if\ r \neq 0$.
- The function is also C_1 continuous, since the gradients of the two parts of the function match at $r = \pm\delta$.

- Optimizing the Huber loss is much faster than using the Laplace likelihood, since we can use standard optimization methods (quasi-Newton) rather than linear programming.
- The Huber method also has a probabilistic interpretation, although it is rather unnatural (Pontil et al. 1998).

- Pontil, M., S. Mukherjee, and F. Girosi (1998). On the Noise Model of Support Vector Machine Regression. Technical report, MIT AI Lab.
- Huber, P. (1964). Robust estimation of a location parameter. *Annals of Statistics 53*, 73a ˘A, S101.

Regularized LS – Ridge Regression

- Consider the error function:

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

data term + regularization term

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

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

- Specifically, setting the gradient with respect to $\boldsymbol{w}$ to zero, and solving for $\boldsymbol{w}$ as before, we obtain

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

- This is a trivial extension of the least-squares solution we encountered earlier (*Regularized Least Squares – Ridge Regression*)

Regularized Least Squares

- Regularized solution:

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

- Regularization limits the effective model complexity (the appropriate number of basis functions).

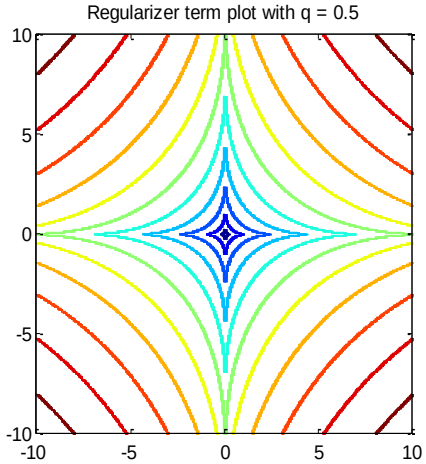
- This is replaced with the problem of finding a suitable value of the regularization coefficient λ.

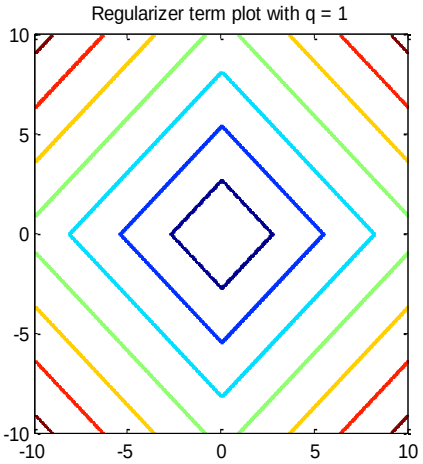
- *λ controls how many non-zero $\boldsymbol{w}$'s (i.e. basis functions) you have.*

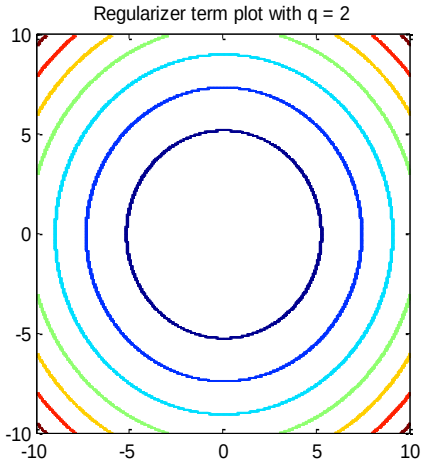
Regularized Least Squares

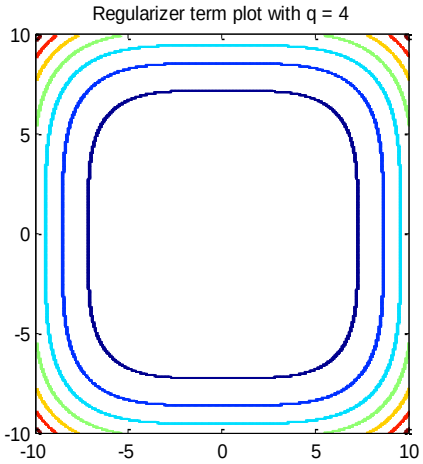
- With a more general regularizer, we have

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









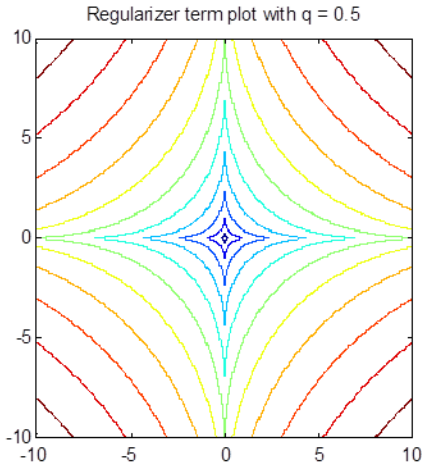
MatLab code

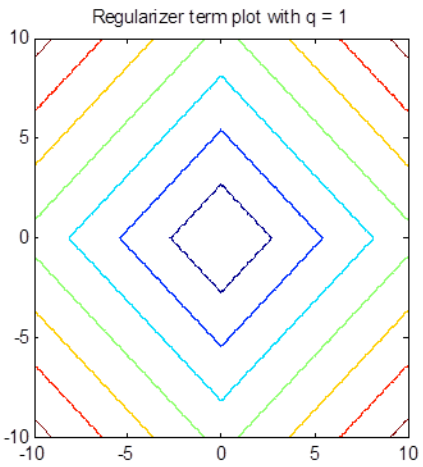
- $q = 1$ is known as the Lasso regularizer. These plots show only the regularizer term with $\lambda = 0.7334$.

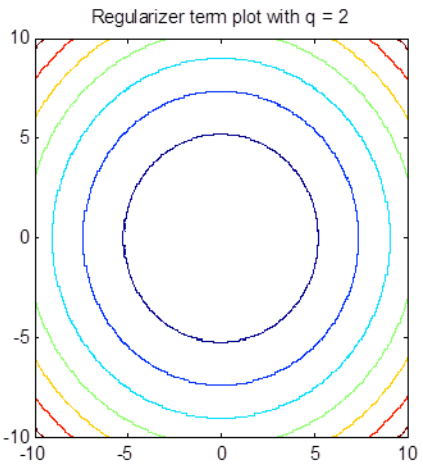
Regularized Least Squares

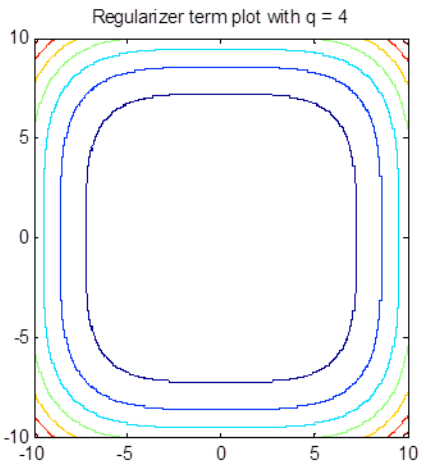
- With a more general regularizer, we have

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







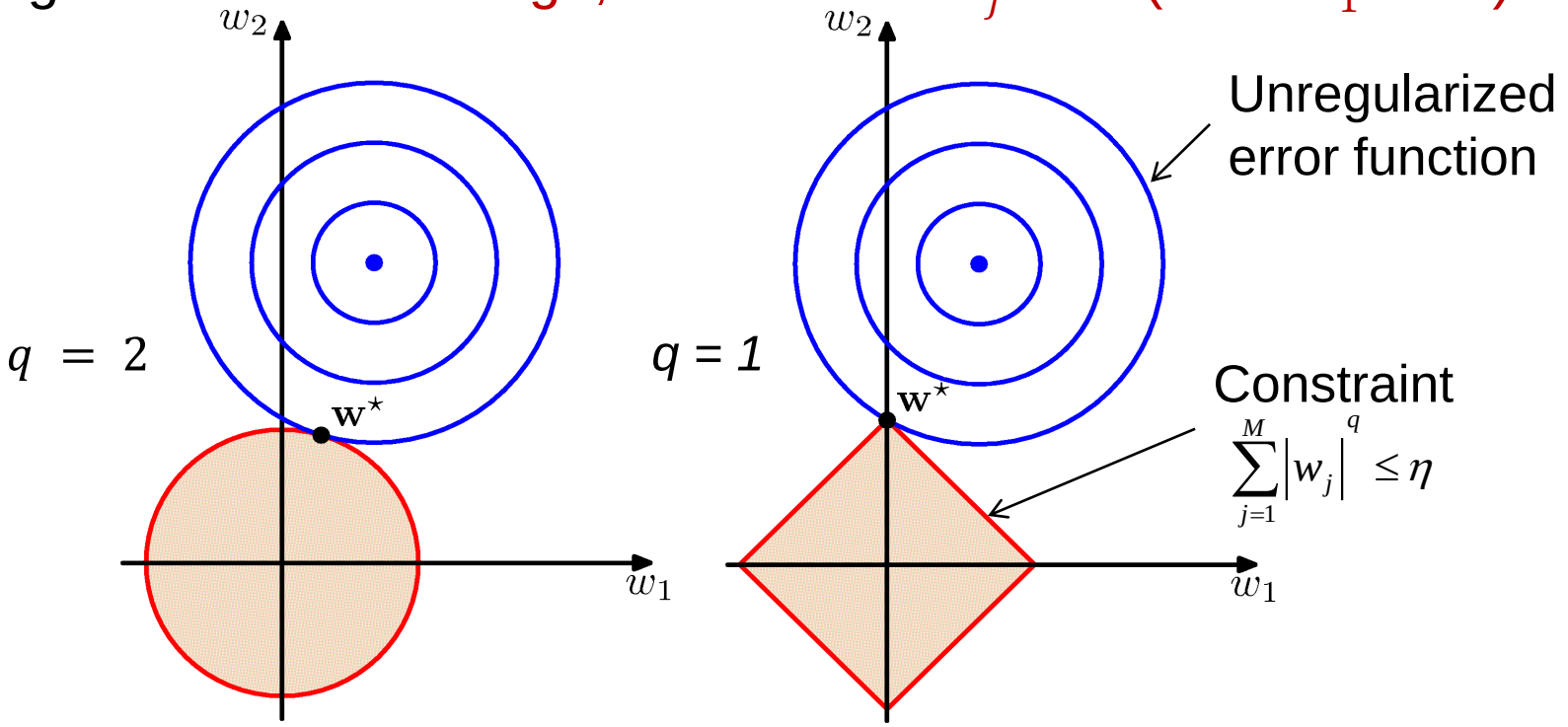


MatLab code

- $q = 2$ corresponds to the quadratic regularizer.

Regularized Least Squares

- Lasso tends to generate sparser solutions than a quadratic regularizer – if λ is large, some of the $w_j \rightarrow 0$ (here $w_1 = 0$).



- Here, we consider that the regularized least squares solution is equivalent to minimizing the unregularized sum of squares with the constraint shown for some η (see proof next).

Regularized Least Squares

- Let us write the constraint in the equivalent form: $\frac{1}{2}\left(\sum_{j=1}^{M}\left|w_j\right|^q - \eta\right) \le 0$

- This leads to the following Lagrangian function:

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

- This is identical to our regularized least squares (RLS) in the dependence on $\boldsymbol{w}$.

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

- For a particular $\lambda > 0$, let $\boldsymbol{w}^*(\lambda)$ be the solution of the RLS in (*).

- From the Kuhn-Tucker optimality conditions for $L(\boldsymbol{w},\lambda)$ we then see:

$$\eta = \sum_{j=1}^{M}\left|w_j^*\right|^q$$

Kuhn-Tucker Optimality Conditions

- Consider the following constraint minimization problem:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}), \textit{subject to } g(\boldsymbol{x}) \geq 0$$

- This is equivalent as the minimization with respect to $\boldsymbol{x}$ and λ of the following Lagrangian:

$$\min_{\boldsymbol{x},\lambda} L(\boldsymbol{x},\lambda) = f(\boldsymbol{x}) - \lambda\, \mathrm{g}(\boldsymbol{x})$$

subject to the following (Kuhn-Tucker) conditions:

$$\lambda \geq 0,\ \mathrm{g}(\boldsymbol{x}) \geq 0,\ \lambda\, \mathrm{g}(\boldsymbol{x}) = 0$$

- Note for maximization problems, the Lagrangian should be modified as: $L(\boldsymbol{x},\lambda) = f(\boldsymbol{x}) + \lambda\, \mathrm{g}(\boldsymbol{x})$

Multiple Outputs-Isotropic Covariance

- If we want to predict $K > 1$ target variables, we use the same basis for all components of the target vector):

$$p(\boldsymbol{t} \mid \boldsymbol{x}, \boldsymbol{W}, \beta) = \mathcal{N}\left(\boldsymbol{t} \mid \boldsymbol{y}(\boldsymbol{x}, \boldsymbol{W}), \beta^{-1}\boldsymbol{I}\right) = \mathcal{N}\left(\boldsymbol{t} \mid \boldsymbol{W}^T\boldsymbol{\phi}(\boldsymbol{x}), \beta^{-1}\boldsymbol{I}\right)$$

$\boldsymbol{W}$ is an $M \times K$ matrix and $\boldsymbol{t}$ is K dimensional.

- Given observed inputs, $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$, and targets, $\boldsymbol{T} = \{\boldsymbol{t}_1, ..., \boldsymbol{t}_N\}^T$ we obtain the log likelihood function

$$\ln p(\boldsymbol{T} \mid \boldsymbol{X}, \boldsymbol{W}, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}\left(t_n \mid \boldsymbol{W}^T\boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}\boldsymbol{I}\right) = \frac{NK}{2}\ln\frac{\beta}{2\pi} - \frac{\beta}{2}\sum_{n=1}^{N}\left\|\boldsymbol{t}_n - \boldsymbol{W}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right\|^2$$

K-Independent Regression Problems

$$\ln p\left(\boldsymbol{T} \mid \boldsymbol{X}, \boldsymbol{W}, \beta\right) = \frac{NK}{2}\ln\frac{\beta}{2\pi} - \frac{\beta}{2}\sum_{n=1}^{N}\left\|\boldsymbol{t}_n - \boldsymbol{W}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right\|^2$$

- As before, we can maximize this function with respect to $\boldsymbol{W}$, giving

$$\underset{M\times K}{\boldsymbol{W}_{ML}} = \underbrace{\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}}_{M\times M}\underset{M\times N}{\boldsymbol{\Phi}^T}\ \underset{N\times K}{\boldsymbol{T}}$$

- If we examine this result for each target variable t_k, we have (take the k^{th} column of $\boldsymbol{W}$ and $\boldsymbol{T}$):

$$\boldsymbol{w}_{k_{ML}} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{t}_k = \boldsymbol{\Phi}^{\dagger}\boldsymbol{t}_k$$

  which is identical with the single output case (so there is decoupling between the target variables)

- As expected, we obtain K –independent regression problems.

Multiple Outputs – Full Covariance

- Let us repeat the earlier formulation but with covariance matrix $\boldsymbol{\Sigma}$. If we want to predict $K > 1$ target variables, we use the same basis for all components of the target vector):

$$p(\boldsymbol{t} \mid \boldsymbol{x}, \boldsymbol{W}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{t} \mid \boldsymbol{y}(\boldsymbol{x}, \boldsymbol{W}), \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{t} \mid \boldsymbol{W}^T \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\Sigma})$$

where $\boldsymbol{W}$ is an $M \times K$ matrix of parameters

- Given observed inputs, $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$, and targets, $\boldsymbol{T} = \{\boldsymbol{t}_1, ..., \boldsymbol{t}_N\}^T$ we obtain the log likelihood function $\ln p(\boldsymbol{T} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\Sigma})$:

$$\sum_{n=1}^{N} \ln \mathcal{N}(t_n \mid \boldsymbol{W}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \boldsymbol{\Sigma}) = -\frac{N}{2} \ln |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{n=1}^{N} (\boldsymbol{t}_n - \boldsymbol{W}^T \boldsymbol{\phi}(\boldsymbol{x}_n))^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{t}_n - \boldsymbol{W}^T \boldsymbol{\phi}(\boldsymbol{x}_n))$$

Multiple Outputs – Full Covariance

$$\ln p\left(\boldsymbol{T} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\Sigma}\right) = -\frac{N}{2}\ln\left|\boldsymbol{\Sigma}\right| - \frac{1}{2}\sum_{n=1}^{N}\left(\boldsymbol{t}_n - \boldsymbol{W}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)^T\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{t}_n - \boldsymbol{W}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)$$

- As before, we maximize this function with respect to $\boldsymbol{W}$,

$$0 = -\sum_{n=1}^{N}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{t}_n - \boldsymbol{W}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\boldsymbol{\phi}(\boldsymbol{x}_n)^T \Rightarrow \underset{M\times K}{\boldsymbol{W}_{ML}} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{T}$$

- For the ML estimate for $\boldsymbol{\Sigma}$, use the result for the MLE of the covariance of a multivariate Gaussian:

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{t}_n - \boldsymbol{W}_{ML}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\left(\boldsymbol{t}_n - \boldsymbol{W}_{ML}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)^T$$

- Note that each column of $\boldsymbol{W}_{ML}$ is of the form

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

seen for isotropic noise distribution and is independent of $\boldsymbol{\Sigma}$!

Bayesian Linear Regression

- Effective model complexity in MLE is governed by the number of basis functions and is controlled by the size of the data set.
- With regularization, the effective model complexity is controlled mainly by λ and still by the number and form of the basis functions.
- *Thus the model complexity for a particular problem cannot be decided simply by maximizing the likelihood function as this leads to excessively complex models and over-fitting.*
- Independent hold-out data can be used to determine model complexity but this wastes data and it is computationally expensive.

Bayesian Linear Regression

- A Bayesian treatment of linear regression avoids the over-fitting of maximum likelihood.

- Bayesian approaches lead to automatic methods of determining model complexity using the training data alone.

Bayesian Linear Regression

- Assume additive Gaussian noise with known precision β. The likelihood function $p(\boldsymbol{t}|\boldsymbol{w})$ is the exponential of a quadratic function of $\boldsymbol{w}$

$$p\left(\boldsymbol{t} \mid \boldsymbol{X}, \boldsymbol{w}, \beta\right) = \prod_{n=1}^{N} \mathcal{N}\left(t_n \mid \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}\right) = \left(\frac{\beta}{2\pi}\right)^{N/2} \exp\left(-\frac{\beta}{2} \sum_{n=1}^{N} \left\{t_n - \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n)\right\}^2\right)$$

and its conjugate prior is Gaussian:

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{m}_0, \boldsymbol{S}_0)$$

- Combining this with the likelihood and using results for marginal and conditional Gaussian distributions, gives the posterior

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

where

$$\boldsymbol{m}_N = \boldsymbol{S}_N \left(\boldsymbol{S}_0^{-1} \boldsymbol{m}_0 + \beta \boldsymbol{\Phi}^T \boldsymbol{t}\right)$$

$$\boldsymbol{S}_N^{-1} = \boldsymbol{S}_0^{-1} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

Posterior Distribution: Derivation

$$p(\boldsymbol{w} \mid \boldsymbol{m}_0, \boldsymbol{S}_0) \propto \exp\left\{-\frac{1}{2}\left(\boldsymbol{w}-\boldsymbol{m}_0\right)^T \boldsymbol{S}_0^{-1}\left(\boldsymbol{w}-\boldsymbol{m}_0\right)\right\},$$

$$p(\boldsymbol{t} \mid \boldsymbol{x}, \boldsymbol{w}, \beta) \propto \exp\left\{-\frac{\beta}{2}\sum_{n=1}^{N}\left(\boldsymbol{\phi}(x_n)^T \boldsymbol{w} - t_n\right)^2\right\} \Rightarrow$$

$$p(\boldsymbol{t} \mid \boldsymbol{x}, \boldsymbol{w}, \beta) \propto \exp\left\{-\frac{1}{2}\boldsymbol{w}^T \sum_{n=1}^{N} \beta\boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_n)^T \boldsymbol{w} + \beta\sum_{n=1}^{N} t_n \boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{w}\right\}$$

- We now have the product of two Gaussians and the posterior is easily computed as:

Complete the square in $\boldsymbol{w}$

$$p(\boldsymbol{w} \mid \boldsymbol{x}, \boldsymbol{t}, \boldsymbol{S}_0, \beta) \propto$$

$$\exp\left(-\frac{1}{2}\boldsymbol{w}^T \boldsymbol{S}_0^{-1}\boldsymbol{w} - \boldsymbol{w}^T \boldsymbol{S}_0^{-1}\boldsymbol{m}_0 - \frac{1}{2}\boldsymbol{w}^T \sum_{n=1}^{N} \beta\boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_n)^T \boldsymbol{w} + \beta\boldsymbol{w}^T \sum_{n=1}^{N} t_n \boldsymbol{\phi}(\boldsymbol{x}_n)\right)$$

$$\propto \mathcal{N}\left(\boldsymbol{w} \mid \underbrace{\boldsymbol{S}_N\left(\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 + \beta\Phi^T \boldsymbol{t}\right)}_{\boldsymbol{m}_N}, \boldsymbol{S}_N\right), \quad \boldsymbol{S}_N^{-1} = \boldsymbol{S}_0^{-1} + \sum_{n=1}^{N} \beta\boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_n)^T = \boldsymbol{S}_0^{-1} + \beta\Phi^T\Phi$$

Sequential Posterior Calculation

- Note that because the posterior distribution is Gaussian, its posterior mode coincides with its mean.

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

$$\boldsymbol{m}_N = \boldsymbol{S}_N \left(\boldsymbol{S}_0^{-1} \boldsymbol{m}_0 + \beta \Phi^T \boldsymbol{t} \right)$$

$$\boldsymbol{S}_N^{-1} = \boldsymbol{S}_0^{-1} + \beta \Phi^T \Phi$$

$$\boldsymbol{w}_{MAP} = \boldsymbol{m}_N$$

- The above expressions for the posterior mean and variance can also be written for a sequential calculation (we already have observed N data points and now considering an additional data point $(\boldsymbol{x}_{N+1}, t_{N+1})$). In this case, we have:

$$p(\boldsymbol{w} \mid t_{N+1}, \boldsymbol{x}_{N+1}, \boldsymbol{m}_N, \boldsymbol{S}_N) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{m}_{N+1}, \boldsymbol{S}_{N+1})$$

$$\boldsymbol{m}_{N+1} = \boldsymbol{S}_{N+1} \left(\boldsymbol{S}_N^{-1} \boldsymbol{m}_N + \beta \boldsymbol{\phi}_{n+1} t_{n+1} \right)$$

$$\boldsymbol{S}_{N+1}^{-1} = \boldsymbol{S}_N^{-1} + \beta \boldsymbol{\phi}_{n+1} \boldsymbol{\phi}_{n+1}^T$$

Bayesian Linear Regression

- Let us consider for a prior, a zero-mean isotropic Gaussian governed by a single precision parameter α so that

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid 0, \alpha^{-1}\boldsymbol{I})$$

and the corresponding posterior distribution over $\boldsymbol{w}$ is then given by

$$\boldsymbol{m}_N = \beta \boldsymbol{S}_N \boldsymbol{\Phi}^T \boldsymbol{t}$$
$$\boldsymbol{S}_N^{-1} = \alpha \boldsymbol{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

- The log of the posterior is the sum of the log likelihood and the log of the prior and, as a function of $\boldsymbol{w}$, takes the form

$$\ln p(\boldsymbol{w} \mid \boldsymbol{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\left\{t_n - \boldsymbol{w}^T\boldsymbol{\phi}(x_n)\right\}^2 - \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} + const$$

- Thus the MAP estimate is the same as regularized least squares (Ridge Regression) with $\lambda = \alpha / \beta$.

A Note on Data Centering

- In linear regression, it helps to center the data in a way that does not require us to compute the offset term μ. Write the likelihood as:

$$p(\boldsymbol{t} \mid \boldsymbol{x}, \boldsymbol{w}, \mu, \beta) \propto \exp\left(-\frac{\beta}{2} \left(\boldsymbol{t} - \mu \mathbf{1}_N - \boldsymbol{\Phi} \boldsymbol{w} \right)^T \left(\boldsymbol{t} - \mu \mathbf{1}_N - \boldsymbol{\Phi} \boldsymbol{w} \right) \right)$$

- Let us assume that the input data are centered in each dimension such that:

$$\sum_{j=1}^{N} \phi_i \left(\boldsymbol{x}_j \right) = 0 \; \forall \; i = 1, \ldots, M$$

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_1) \\ \boldsymbol{\phi}^T(\boldsymbol{x}_2) \\ \vdots \\ \boldsymbol{\phi}^T(\boldsymbol{x}_N) \end{pmatrix} = \begin{pmatrix} \phi_1(\boldsymbol{x}_1) & \phi_2(\boldsymbol{x}_1) & .. & \phi_M(\boldsymbol{x}_1) \\ \phi_1(\boldsymbol{x}_2) & \phi_2(\boldsymbol{x}_2) & .. & \phi_M(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(\boldsymbol{x}_N) & \phi_2(\boldsymbol{x}_N) & .. & \phi_M(\boldsymbol{x}_N) \end{pmatrix}, \quad \boldsymbol{\Phi}^T = \begin{pmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1) & \boldsymbol{\phi}(\boldsymbol{x}_2) & .. & \boldsymbol{\phi}(\boldsymbol{x}_N) \end{pmatrix}$$

$$\boldsymbol{\Phi}^T = \begin{pmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & .. & \boldsymbol{\phi}_N \end{pmatrix}$$

$$\boldsymbol{\phi}_i(\boldsymbol{x}) = \begin{pmatrix} \phi_0(\boldsymbol{x}_i) & \phi_1(\boldsymbol{x}_i) & . & . & \phi_{M-1}(\boldsymbol{x}_i) \end{pmatrix}^T$$

- The mean of the output is equally likely to be positive or negative. Let us put an improper prior $p(\mu) \propto 1$ and integrate μ out.

A Note on Data Centering

- Introducing, $\bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i$, the marginal likelihood becomes :

$$p(\boldsymbol{t} \mid \boldsymbol{x}, \boldsymbol{w}, \beta) \propto \int \exp\left(-\frac{\beta}{2} \left(\underbrace{\boldsymbol{t} - \bar{t}\mathbf{1}_N - \boldsymbol{\Phi}\boldsymbol{w}}_{\boldsymbol{A}} - \left(\mu - \bar{t}\right)\mathbf{1}_N \right)^T \left(\boldsymbol{t} - \bar{t}\mathbf{1}_N - \boldsymbol{\Phi}\boldsymbol{w} - \left(\mu - \bar{t}\right)\mathbf{1}_N \right) \right) d\mu$$

- Completing the square in μ gives (use the centering of the input):

$$p(\boldsymbol{t} \mid \boldsymbol{x}, \boldsymbol{w}, \beta) \propto \int \exp\left(-\frac{\beta}{2} \left(\left(\mu - \bar{t}\right)^2 N - 2\left(\mu - \bar{t}\right) \underbrace{\boldsymbol{A}^T \mathbf{1}_N}_{\underbrace{\bar{t}N - \bar{t}N}_{0} - \boldsymbol{w}^T \underbrace{\boldsymbol{\Phi}^T \mathbf{1}_N}_{=0}} + \boldsymbol{A}^T \boldsymbol{A} \right) \right) d\mu$$

$$\propto \exp\left(-\frac{\beta}{2} \left(\boldsymbol{t} - \bar{t}\mathbf{1}_N - \boldsymbol{\Phi}\boldsymbol{w} \right)^T \left(\boldsymbol{t} - \bar{t}\mathbf{1}_N - \boldsymbol{\Phi}\boldsymbol{w} \right) \right)$$

- Our model is now simplified if instead of $\boldsymbol{t}$ we use (centered output) $\boldsymbol{t} \leftarrow \boldsymbol{t} - \bar{t}\mathbf{1}_N$ and the likelihood is simply written as:

$$p(\boldsymbol{t} \mid \boldsymbol{x}, \boldsymbol{w}, \beta) \propto \exp\left(-\frac{\beta}{2} \left(\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{w} \right)^T \left(\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{w} \right) \right)$$

$$\hat{\mu} = \bar{t} - \sum_{j=1}^{D} \bar{\boldsymbol{\phi}}_j^T \boldsymbol{w}_j \,,\ \left[\bar{\boldsymbol{\phi}}_1, \ldots, \bar{\boldsymbol{\phi}}_D\right] \text{is formed by averaging each column of } \boldsymbol{\Phi}$$

- Recall that the MLE estimate for μ is:

A Note on Data Centering

- To simplify the earlier notation, consider a linear regression model of the form

$$\mathbb{E}[y \mid \boldsymbol{x}] = w_0 + \boldsymbol{w}^T \boldsymbol{x}$$

- In the context e.g. of MLE, we need to minimize

$$\min_{w_0, \boldsymbol{w}} = \sum_{i=1}^{N} \left(t_i - w_0 - \boldsymbol{w}^T \boldsymbol{x}_i \right)^2$$

- Minimization wrt w_0 gives:

$$\sum_{i=1}^{N} \left(t_i - w_0 - \boldsymbol{w}^T \boldsymbol{x}_i \right) = 0 \Rightarrow w_0 N = \bar{t} N - N \boldsymbol{w}^T \bar{\boldsymbol{x}} \Rightarrow w_0 = \bar{t} - \boldsymbol{w}^T \bar{\boldsymbol{x}}$$

where:

$$\bar{\boldsymbol{x}} \equiv \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_D \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N} x_{i1} \Big/ N \\ \sum_{i=1}^{N} x_{i2} \Big/ N \\ \vdots \\ \sum_{i=1}^{N} x_{iD} \Big/ N \end{pmatrix}, \quad \bar{t} \equiv \sum_{i=1}^{N} t_i \Big/ N$$

- Thus:

$$\hat{w}_0 = \bar{t} - \bar{\boldsymbol{x}}^T \boldsymbol{w}$$

A Note on Data Centering

- Substituting the bias term in our objective function gives:

$$\min_{\boldsymbol{w}} \sum_{i=1}^{N} \left(t_i - \bar{t} + \boldsymbol{w}^T \bar{\boldsymbol{x}} - \boldsymbol{w}^T \boldsymbol{x}_i \right)^2 = \min_{\boldsymbol{w}} \sum_{i=1}^{N} \left(t_i - \bar{t} + \boldsymbol{w}^T \left(\bar{\boldsymbol{x}} - \boldsymbol{x}_i \right) \right)^2$$

- Minimization wrt $\boldsymbol{w}$ gives:

$$\sum_{i=1}^{N} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \hat{\boldsymbol{w}} = \sum_{i=1}^{N} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{t}_i - \bar{\boldsymbol{t}})$$

- We thus first compute the MLE of $\boldsymbol{w}$ using the centered input and output as follows:

$$\hat{\boldsymbol{w}} = \left(\boldsymbol{X}_c^T \boldsymbol{X}_c \right)^{-1} \boldsymbol{X}_c^T \boldsymbol{t}_c = \left[\sum_{i=1}^{N} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \right] \left[\sum_{i=1}^{N} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{t}_i - \bar{\boldsymbol{t}}) \right]$$

$$\boldsymbol{X}_c = \boldsymbol{X} - \bar{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{1}_N \bar{\boldsymbol{x}}^T = \begin{pmatrix} \boldsymbol{x}_1^T - \bar{\boldsymbol{x}}^T \\ \boldsymbol{x}_2^T - \bar{\boldsymbol{x}}^T \\ \vdots \\ \boldsymbol{x}_N^T - \bar{\boldsymbol{x}}^T \end{pmatrix} = \begin{pmatrix} x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & .. & x_{1D} - \bar{x}_D \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & .. & x_{2D} - \bar{x}_D \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} - \bar{x}_1 & x_{N2} - \bar{x}_2 & .. & x_{ND} - \bar{x}_D \end{pmatrix}, \quad \bar{\boldsymbol{x}} \equiv \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_D \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N} x_{i1} \Big/ N \\ \sum_{i=1}^{N} x_{i2} \Big/ N \\ \vdots \\ \sum_{i=1}^{N} x_{iD} \Big/ N \end{pmatrix}, \quad \boldsymbol{t}_c = \boldsymbol{t} - \bar{\boldsymbol{t}} = \boldsymbol{t} - \boldsymbol{1}_N \bar{t}, \quad \bar{t} \equiv \sum_{i=1}^{N} t_i \Big/ N$$

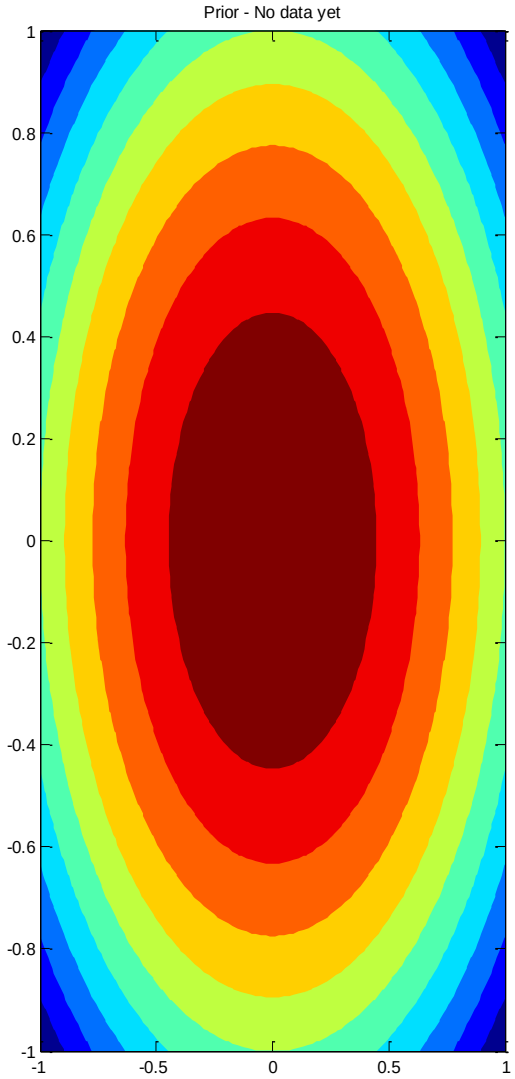
- We can then estimate the MLE estimate of w_0 as follows:

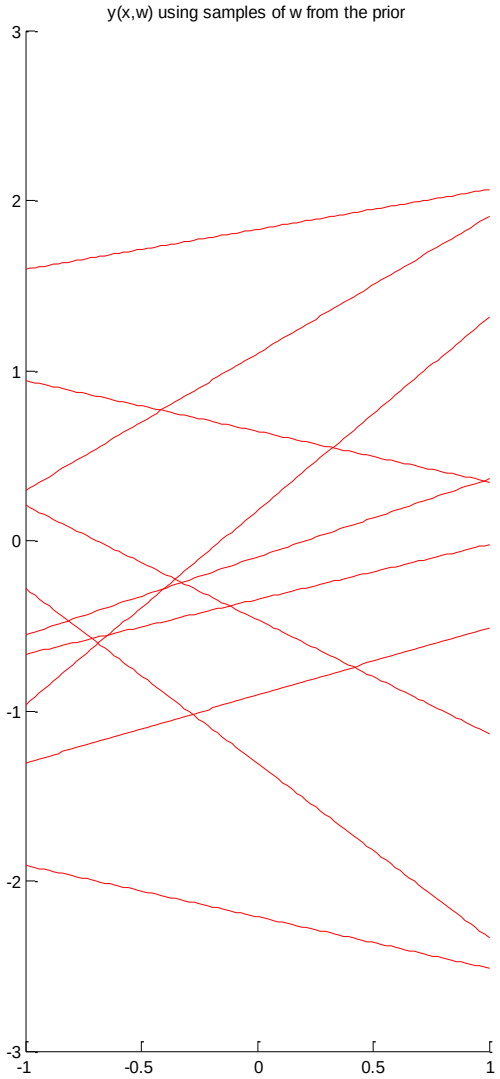
$$\hat{w}_0 = \bar{t} - \bar{\boldsymbol{x}}^T \boldsymbol{w}$$

Bayesian Regression: Example

- We generate synthetic data from the function $f(x, a) = a_0 + a_1 x$ with parameter values $a_0 = -0.3$ and $a_1 = 0.5$ by first choosing values of x_n from the uniform distribution $\mathcal{U}(x|-1, 1)$, then evaluating $f(x_n, a)$, and finally adding Gaussian noise with standard deviation of 0.2 to obtain the target values t_n.
- We assume $\beta = (1/0.2)^2 = 25$ and $\alpha = 2.0$.
- We perform Bayesian inference sequentially – one point at a time – so the posterior at each level becomes the new prior.
- We show results after $1, 2$ and 22 points have been collected.
- The results include the likelihood contours (for 1 point), the posterior and samples of the regression function from the posterior.

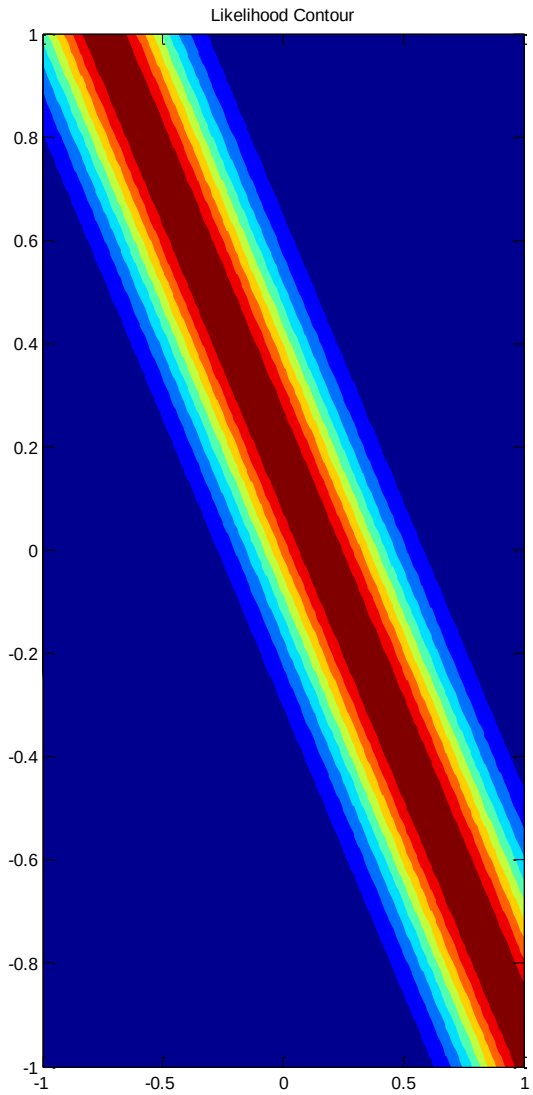
Bayesian Regression: Example

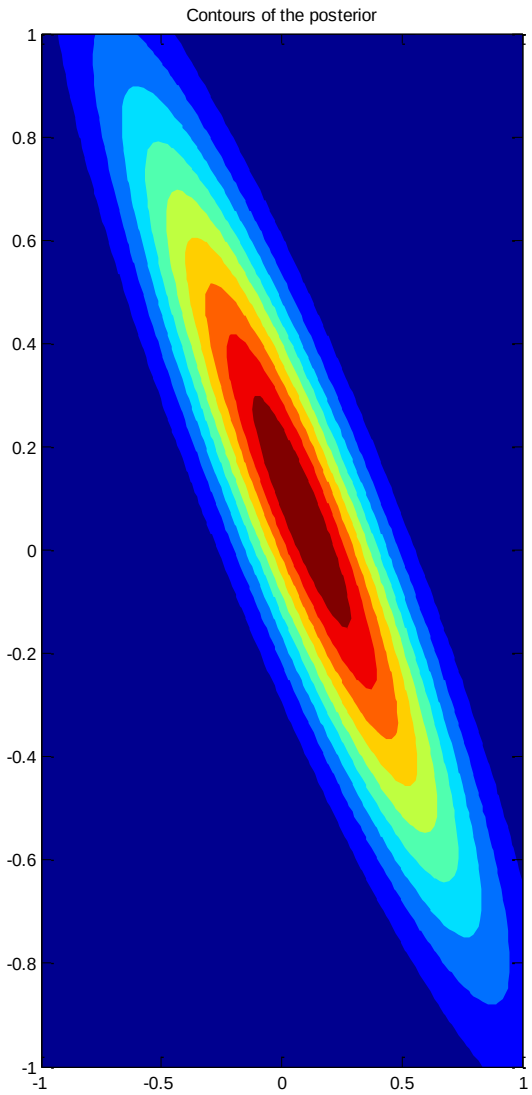


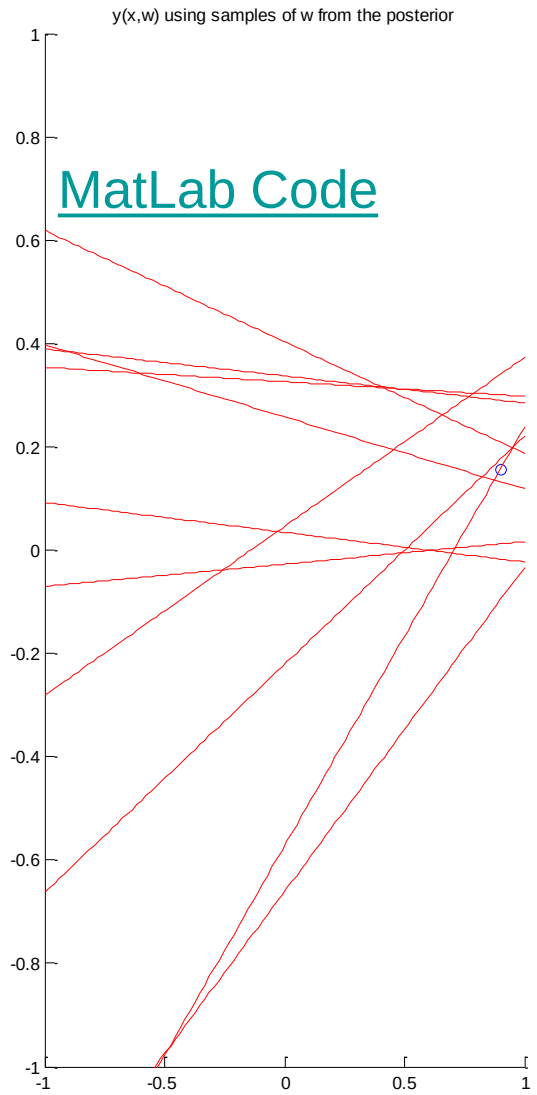


MatLab Code

Example: One Data Point Collected

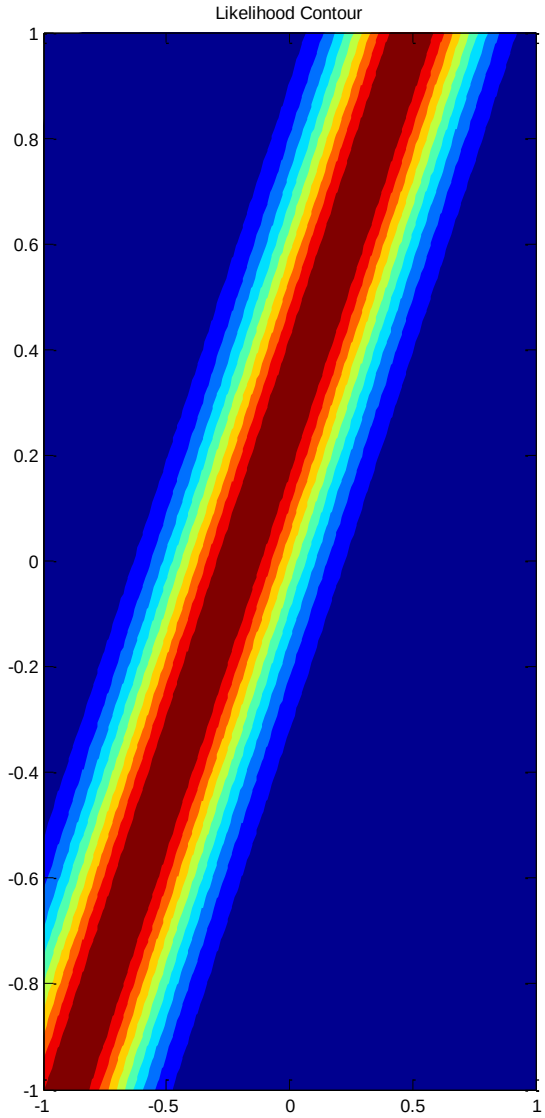


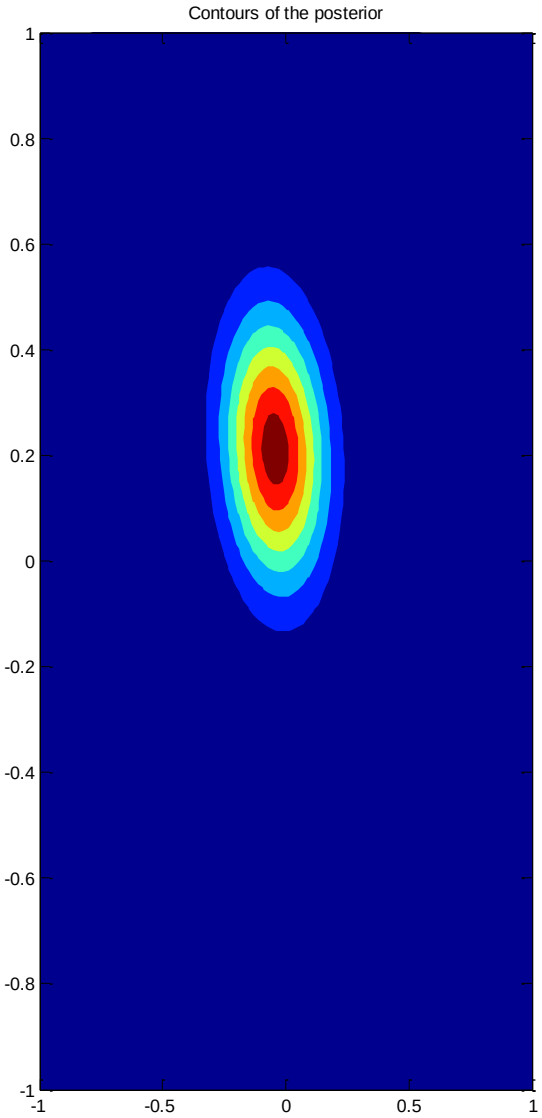


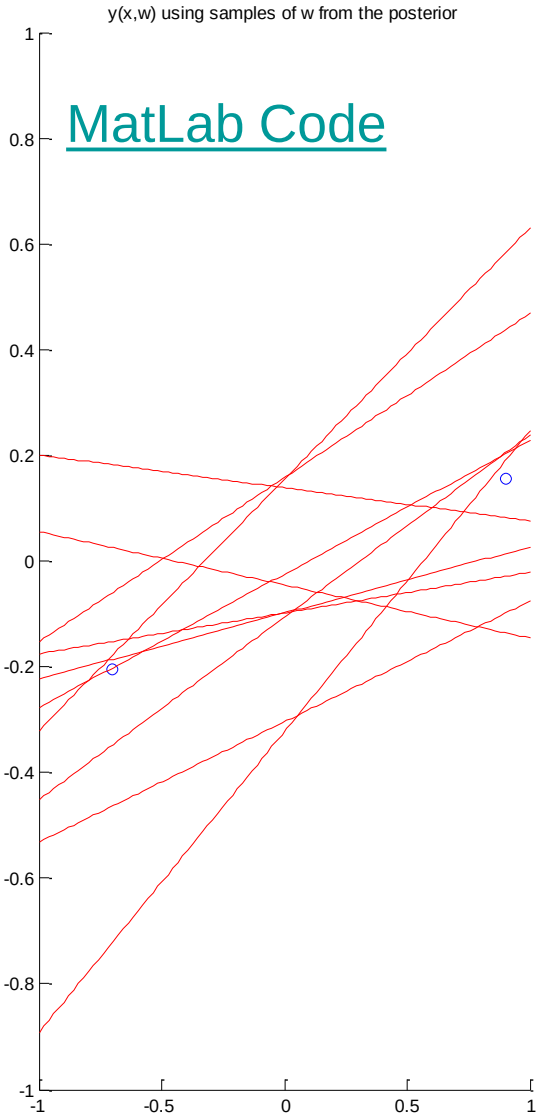


Note that the regression lines pass close to the data point (shown with a circle)

Example: 2nd Data Point Collected



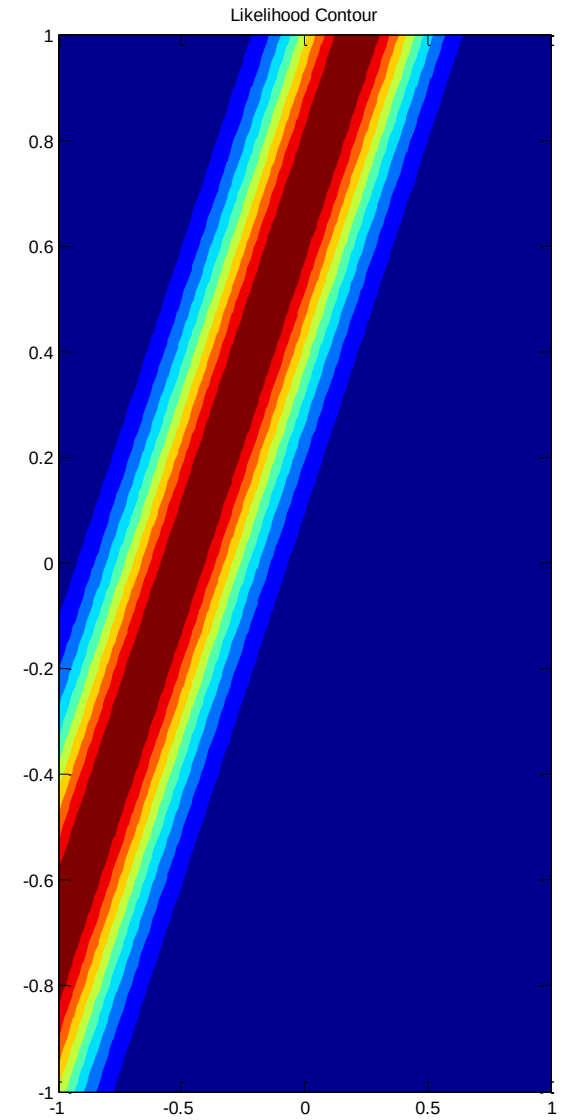


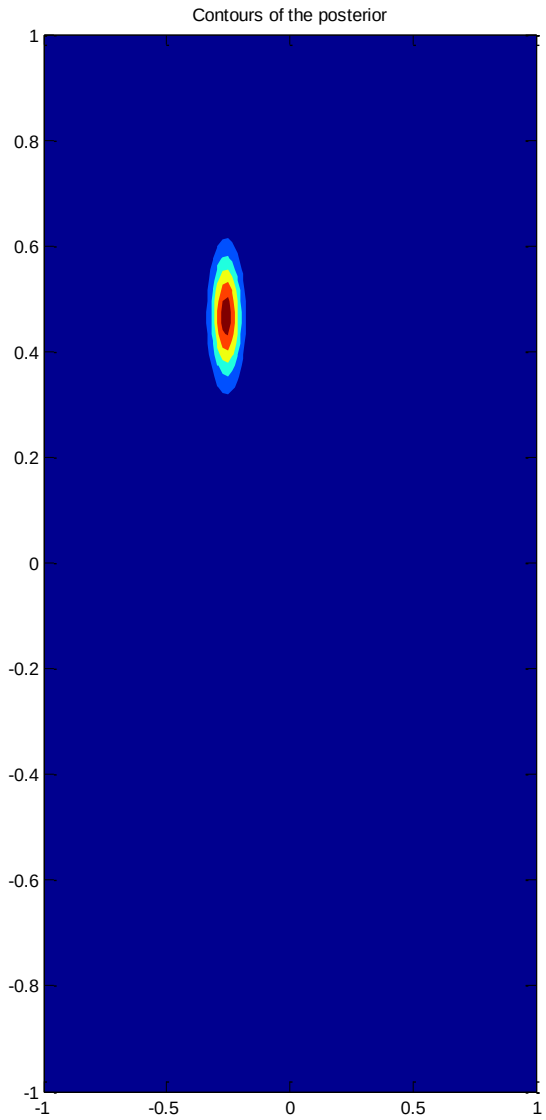


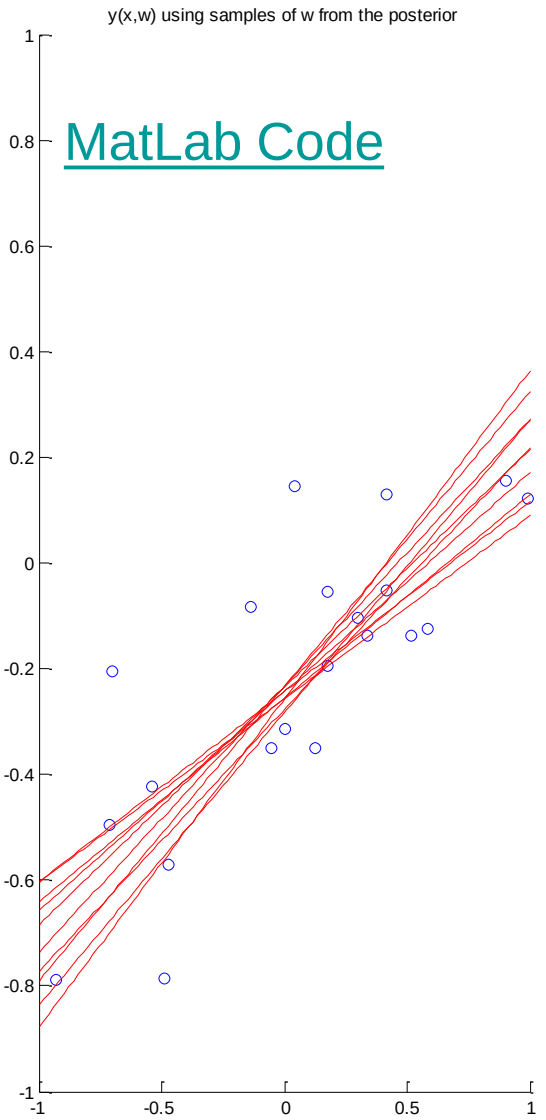
MatLab Code

Note that the regression lines now pass close to both data points

Example: 22 Data Points Collected





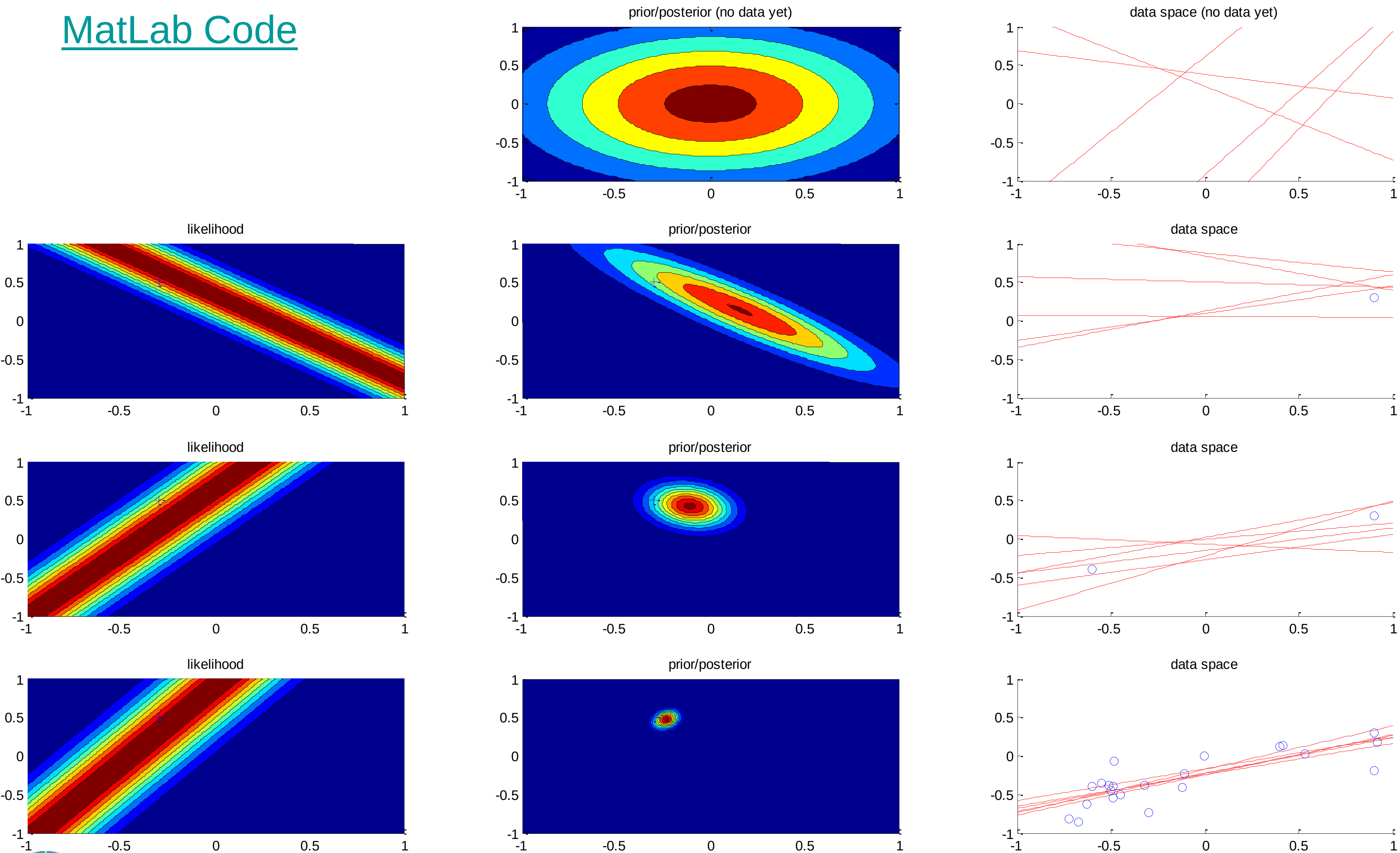


MatLab Code

Note that the regression lines after 22 data points have been collected

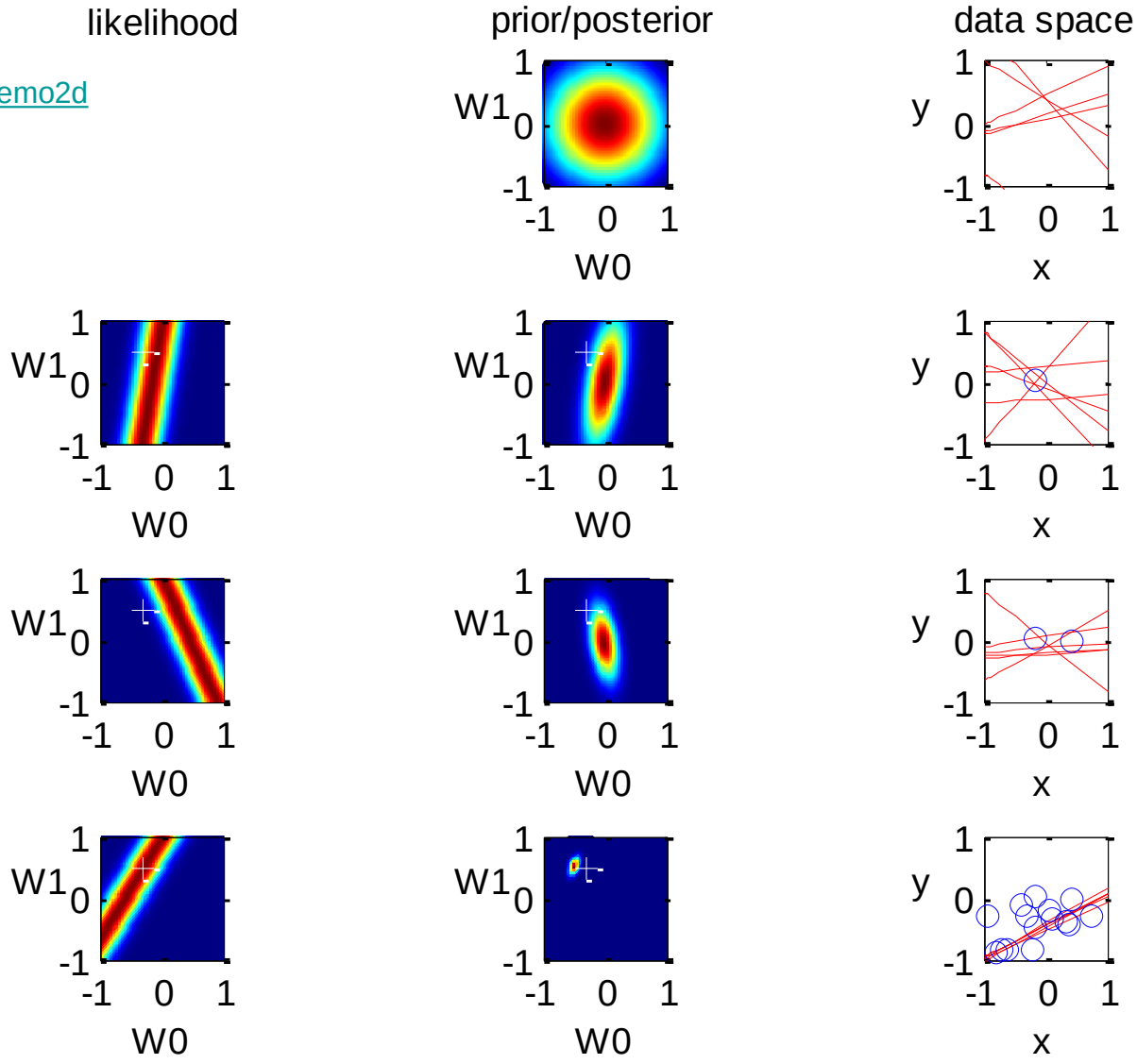
Summary of Results

[MatLab Code]



Summary of Results

Run bayesLinRegDemo2d from PMTK3



Predictive Distribution

- We are not interested in $\boldsymbol{w}$ itself but in making predictions of t for new values of x. This requires the predictive distribution

$$p(t \mid x, \mathbf{x}, \mathbf{t}) = \int p(t \mid x, \mathbf{w}, \beta)\, p(\mathbf{w} \mid \mathbf{x}, \mathbf{t}, \alpha, \beta) d\mathbf{w} = \mathcal{N}\left(t \mid \mathbf{m}_N^T \boldsymbol{\phi}(x), \sigma_N^2(x)\right)$$

where $$\sigma_N^2(x) = \frac{1}{\beta} + \boldsymbol{\phi}(x)^T S_N \boldsymbol{\phi}(x), \quad \boldsymbol{S}_N^{-1} = \alpha \boldsymbol{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

- The 1st term represents the noise on the data whereas the 2nd term reflects the uncertainty associated with $\boldsymbol{w}$.
- Because the noise process and the distribution of $\boldsymbol{w}$ are independent Gaussians, their variances are additive.
- The error bars get larger as we move away from the training points. By contrast, in the plug-in approximation, the error bars are of constant size.
- As additional data points are observed, the posterior distribution becomes narrower.

Predictive Distribution

- In a full Bayesian treatment, we want to compute the predictive distribution, i.e. given the training data $\boldsymbol{x}$ and $\boldsymbol{t}$ and a new test point x, we want the distribution:

$$p(t \mid x, \boldsymbol{x}, \boldsymbol{t}) = \int p(t \mid x, \boldsymbol{w})\, p(\boldsymbol{w} \mid \boldsymbol{x}, \boldsymbol{t}) d\boldsymbol{w}, \quad where$$

$$p(t \mid x, \boldsymbol{w}, \beta) = \mathcal{N}(t \mid y(x, \boldsymbol{w}), \beta^{-1}) \; and$$

$$p(\boldsymbol{w} \mid \boldsymbol{x}, \boldsymbol{t}, \alpha, \beta) \propto$$

$$\exp\left(-\frac{1}{2} \boldsymbol{w}^T \alpha \boldsymbol{I}_{M \times M} \boldsymbol{w} - \frac{1}{2} \boldsymbol{w}^T \sum_{n=1}^{N} \beta \boldsymbol{\phi}(x_n) \boldsymbol{\phi}(x_n)^T \boldsymbol{w} - \beta \sum_{n=1}^{N} t_n \boldsymbol{\phi}(x_n) \boldsymbol{w} \right)$$

$$\propto \mathcal{N}\left(\boldsymbol{w} \mid, \beta \boldsymbol{S}_N \sum_{n=1}^{N} t_n \boldsymbol{\phi}(x_n), \boldsymbol{S}_N \right), \; \boldsymbol{S}_N^{-1} = \alpha \boldsymbol{I}_{M \times M} + \sum_{n=1}^{N} \beta \boldsymbol{\phi}(x_n) \boldsymbol{\phi}(x_n)^T$$

- To compute the marginal, we use the standard equations for Gaussian Linear Systems from an earlier lecture.

Appendix: Useful Result

$$p(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \Lambda^{-1}\right)$$

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{L}^{-1}\right)$$

- For the above linear model, we proved in an earlier lecture that the following very useful results about marginal and conditional Gaussian models hold:

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{L}^{-1} + \boldsymbol{A}\Lambda^{-1}\boldsymbol{A}^{T}\right)$$

$$p(\boldsymbol{x} \mid \boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{x} \mid \left(\Lambda + \boldsymbol{A}^{T}\boldsymbol{L}\boldsymbol{A}\right)^{-1}\left(\Lambda\boldsymbol{\mu} + \boldsymbol{A}^{T}\boldsymbol{L}(\boldsymbol{y} - \boldsymbol{b})\right), \left(\Lambda + \boldsymbol{A}^{T}\boldsymbol{L}\boldsymbol{A}\right)^{-1}\right)$$

Predictive Distribution

$$p(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right)$$

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{L}^{-1}\right)$$

$$p(\boldsymbol{w} \mid \boldsymbol{x}, \boldsymbol{t}, \alpha, \beta) = \mathcal{N}\left(\boldsymbol{w} \mid, \beta \boldsymbol{S}_N \sum_{n=1}^{N} t_n \boldsymbol{\phi}(x_n), \boldsymbol{S}_N\right)$$

$$p(t \mid x, \boldsymbol{w}, \beta) = \mathcal{N}(t \mid y(x, \boldsymbol{w}), \beta^{-1})$$

- Thus for our problem:

$$\boldsymbol{x} \leftarrow \boldsymbol{w}, \boldsymbol{\mu} = \beta \boldsymbol{S}_N \sum_{n=1}^{N} t_n \phi(x_n), \boldsymbol{\Lambda}^{-1} = \boldsymbol{S}_N$$

$$\boldsymbol{y} \leftarrow t, \boldsymbol{A} = \phi(x)^T, \boldsymbol{b} = 0, \boldsymbol{L}^{-1} = \beta^{-1}$$

- The predictive distribution now takes the form:

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{L}^{-1} + \boldsymbol{A}\boldsymbol{\Lambda}^{-1}\boldsymbol{A}^T\right) \Rightarrow$$

$$p(t) = \mathcal{N}\left(t \mid \boldsymbol{\phi}(x)^T \beta \boldsymbol{S}_N \sum_{n=1}^{N} t_n \boldsymbol{\phi}(x_n), \beta^{-1} + \boldsymbol{\phi}(x)^T \boldsymbol{S}_N \boldsymbol{\phi}(x)\right)$$

Predictive Distribution

- In a full Bayesian treatment, we want to compute the predictive distribution, i.e. given the training data $\boldsymbol{x}$ and $\boldsymbol{t}$ and a new test point x, we want the distribution:

$$p(t \mid x, \mathbf{x}, \mathbf{t}) = \int p(t \mid x, \boldsymbol{w})\, p(\boldsymbol{w} \mid \mathbf{x}, \mathbf{t})\, d\boldsymbol{w}, \quad p(t \mid x, \boldsymbol{w}, \beta) = \mathcal{N}(t \mid y(x, \boldsymbol{w}), \beta^{-1}) \Rightarrow$$

$$p(t \mid x, \mathbf{x}, \mathbf{t}) = \mathcal{N}(t \mid m(x), \sigma_N^2(x))$$

where the mean and variance are given by

$$m(x) = \beta \boldsymbol{\phi}(x)^T \boldsymbol{S}_N \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) t_n = \boldsymbol{\phi}(x)^T \boldsymbol{m}_N, \quad \boldsymbol{m}_N = \beta \boldsymbol{S}_N \boldsymbol{\Phi}^T \mathbf{t}$$

$$\sigma_N^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^T \boldsymbol{S}_N \boldsymbol{\phi}(x) \quad (\textit{uncertainty in the data} + \textit{uncertainty in } \boldsymbol{w})$$

$$\boldsymbol{S}_N^{-1} = \alpha \boldsymbol{I} + \beta \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) \boldsymbol{\phi}(x_n)^T = \alpha \boldsymbol{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

- Note that: $\sigma_N^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^T \boldsymbol{S}_N \boldsymbol{\phi}(x) \overset{N \to \infty}{\rightarrow} \beta^{-1}$ and $\sigma_{N+1}^2(x) \le \sigma_N^2(x)$

Predictive Distribution

- It is easy to show:

$$\sigma_N^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^T \boldsymbol{S}_N \boldsymbol{\phi}(x) \overset{N\to\infty}{\to} \beta^{-1} \text{ and } \sigma_{N+1}^2(x) \le \sigma_N^2(x)$$

- Note: $\boldsymbol{S}_{N+1}^{-1} = \alpha \boldsymbol{I} + \beta \sum_{n=1}^{N+1} \boldsymbol{\phi}(x_n)\boldsymbol{\phi}(x_n)^T = \boldsymbol{S}_N^{-1} + \beta \boldsymbol{\phi}(x_n)\boldsymbol{\phi}(x_n)^T$

  and the identity:

$$\left(\boldsymbol{M} + \boldsymbol{v}\boldsymbol{v}^T\right)^{-1} = \boldsymbol{M}^{-1} - \frac{\left(\boldsymbol{M}^{-1}\boldsymbol{v}\right)\left(\boldsymbol{v}^T\boldsymbol{M}^{-1}\right)}{1 + \boldsymbol{v}^T\boldsymbol{M}^{-1}\boldsymbol{v}}$$

- Using these results, we can write:

$$\sigma_{N+1}^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^T \boldsymbol{S}_{N+1} \boldsymbol{\phi}(x) = \beta^{-1} + \boldsymbol{\phi}(x)^T \left(\boldsymbol{S}_N^{-1} + \beta \boldsymbol{\phi}(x_n)\boldsymbol{\phi}(x_n)^T\right)^{-1} \boldsymbol{\phi}(x) = \beta^{-1} +$$

$$\boldsymbol{\phi}(x)^T \left[\boldsymbol{S}_N - \frac{\beta\left(\boldsymbol{S}_N\boldsymbol{\phi}(x_n)\right)\left(\boldsymbol{\phi}(x_n)^T\boldsymbol{S}_N\right)}{1 + \beta\boldsymbol{\phi}(x_n)^T\boldsymbol{S}_N\boldsymbol{\phi}(x_n)}\right]\boldsymbol{\phi}(x) = \sigma_N^2(x) - \frac{\beta\left(\boldsymbol{\phi}(x_n)^T\boldsymbol{S}_N\boldsymbol{\phi}(x)\right)^2}{1 + \beta\boldsymbol{\phi}(x_n)^T\boldsymbol{S}_N\boldsymbol{\phi}(x_n)} \le \sigma_N^2(x)$$

Predictive Distribution: Summary

- The notation used here is as follows:

$$p(t \mid x, \mathbf{x}, \mathbf{t}) = \mathcal{N}(t \mid m(x), \sigma_N^2(x))$$

$$m(x) = \beta \boldsymbol{\phi}(x)^T \mathbf{S}_N \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) t_n$$

$$\sigma_N^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^T \mathbf{S}_N \boldsymbol{\phi}(x)$$

$$\mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) \boldsymbol{\phi}(x_n)^T$$

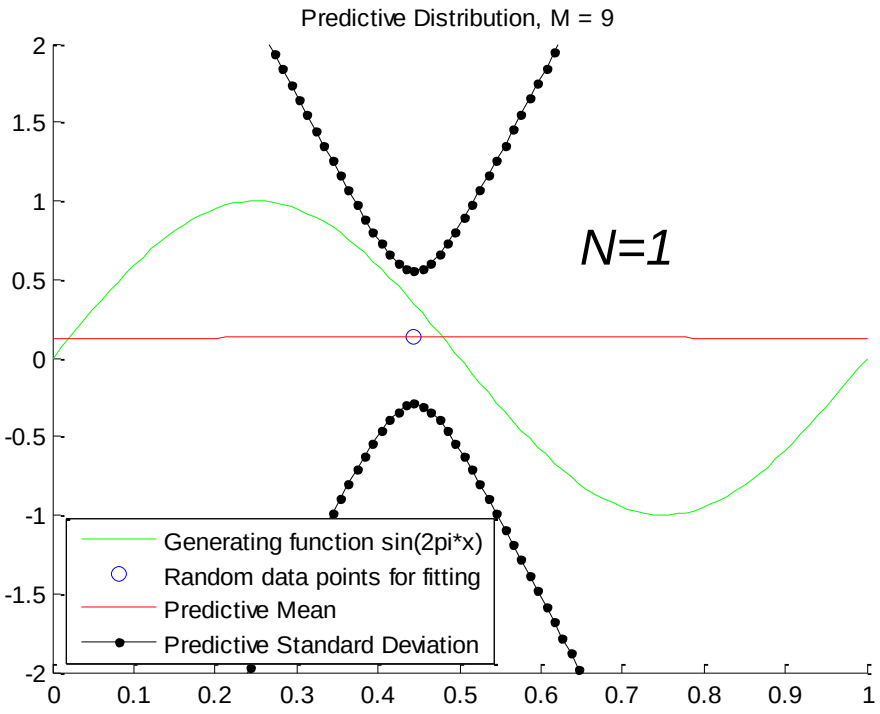
Note:
Predictive mean and variance are functions of x.

$$\boldsymbol{\phi}(x_n) = \begin{Bmatrix} \phi_0(x_n) \\ \phi_1(x_n) \\ \phi_2(x_n) \\ : \\ \phi_{M-1}(x_n) \end{Bmatrix}, \boldsymbol{\phi}(x)^T = \{\phi_0(x_n) \quad \phi_1(x_n) \quad \phi_2(x_n) \quad .. \quad \phi_{M-1}(x_n)\}, \mathbf{I} = \text{unit matrix } M \times M$$

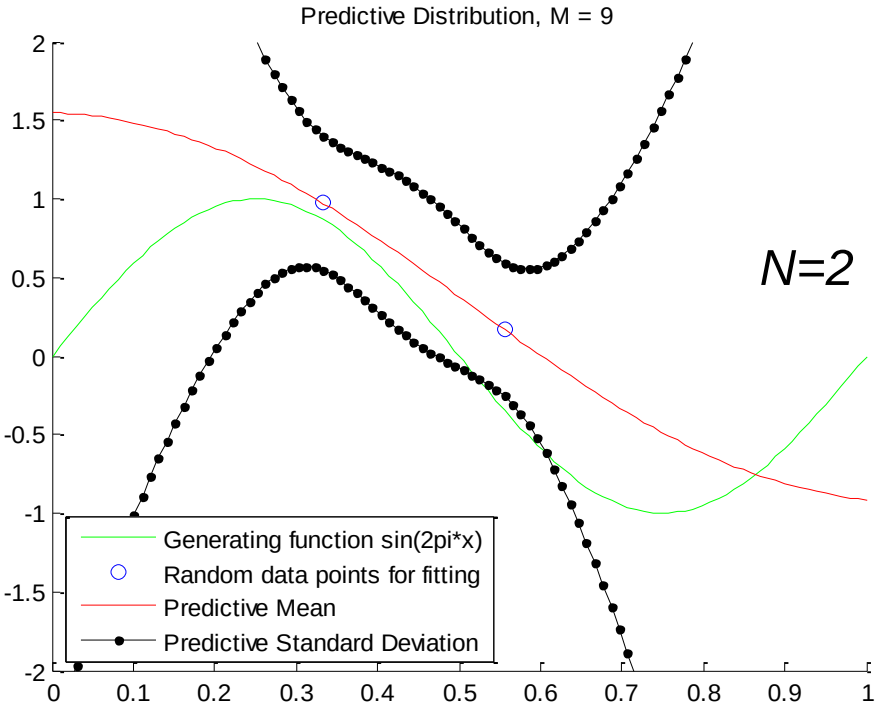
For Polynomial regression:

$$\boldsymbol{\phi}(x_n) = \begin{Bmatrix} 1 \\ x_n \\ x_n^2 \\ : \\ x_n^{M-1} \end{Bmatrix}, \boldsymbol{\phi}(x)^T = \{1 \quad x \quad x^2 \quad .. \quad x^{M-1}\}$$

Pointwise Uncertainty in the Predictions



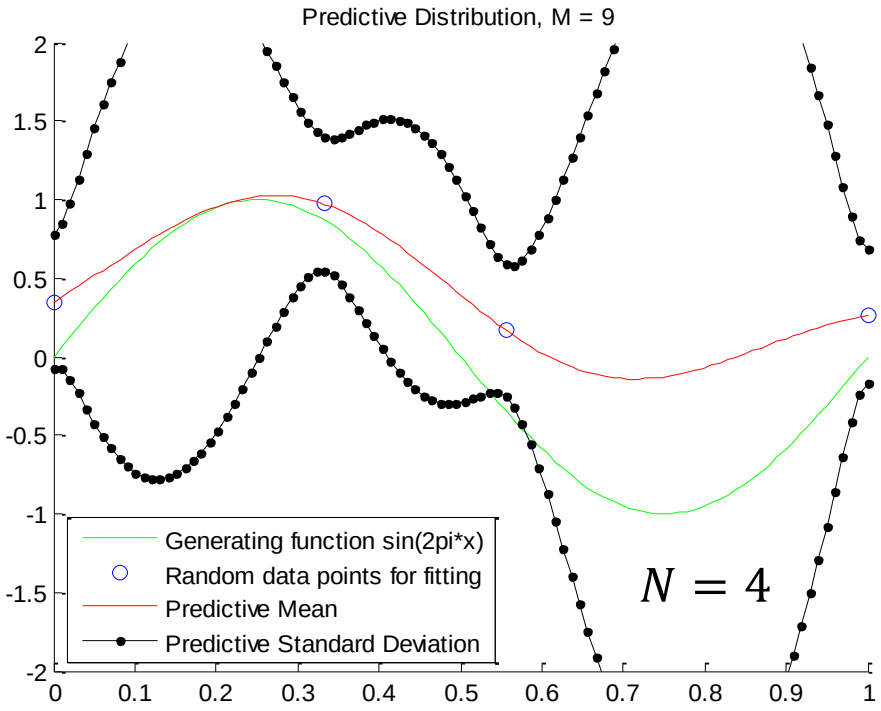
M=9 Gaussians, 10 parameters
Scale of Gaussians adjusted with data
$\alpha = 5 \times 10^{-3}$
$\beta = 11.1$
Using $N = 1,2,4,10$
Data are given here

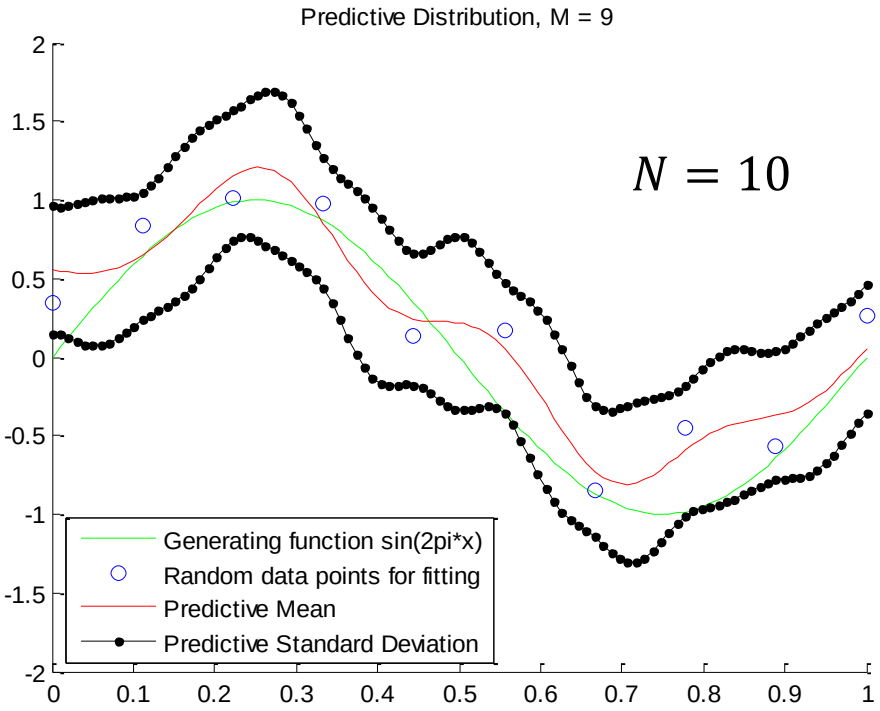


- The predictive uncertainty is smaller near the data.
- The level of uncertainty decreases with N

MatLab code

Pointwise Uncertainty in the Predictions

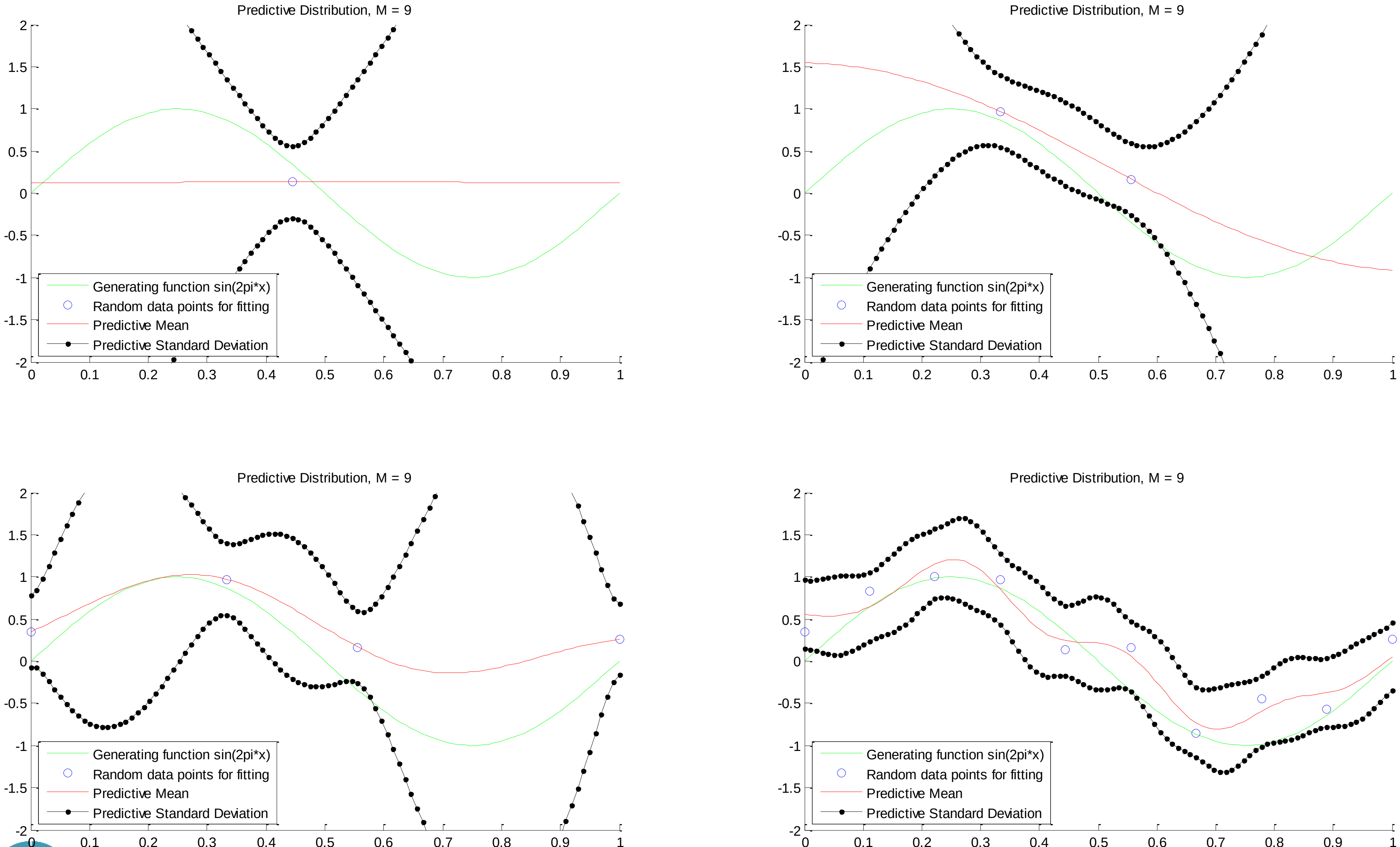




MatLab code

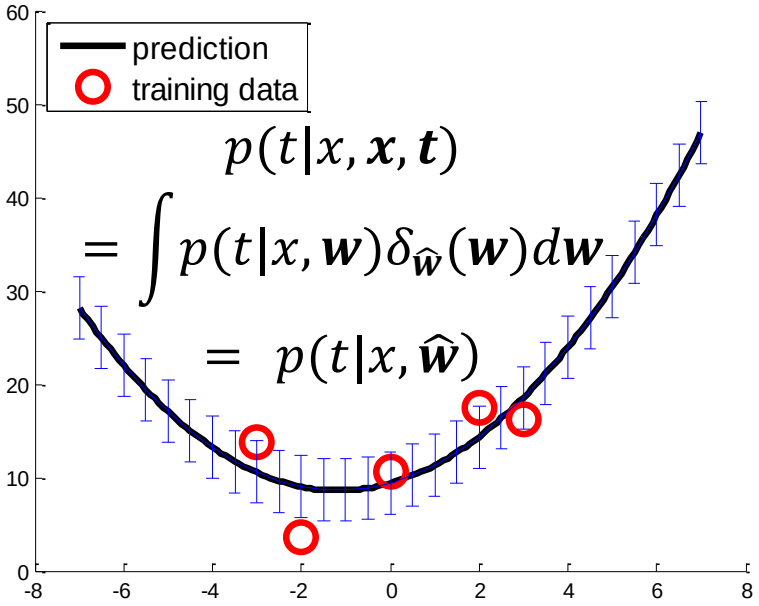
Summary of Results

MatLab code



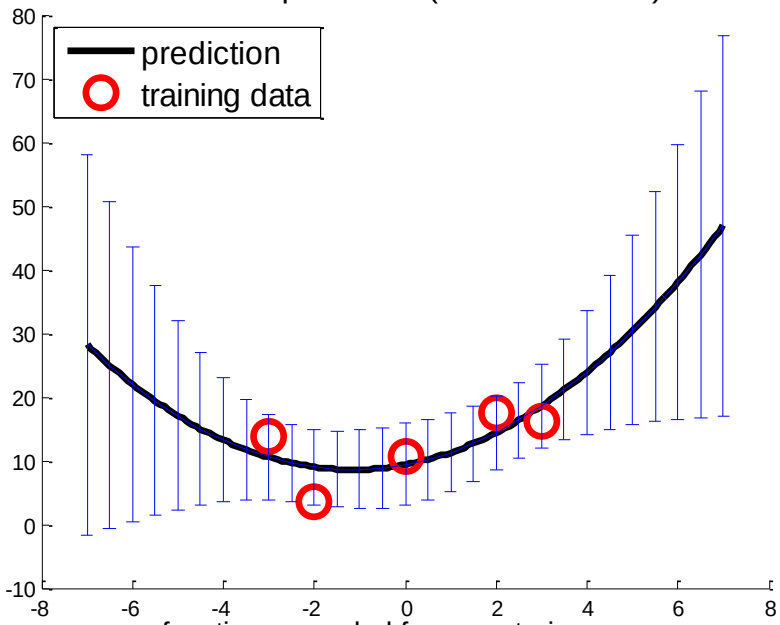
Plugin Approximation

plugin approximation (MLE)

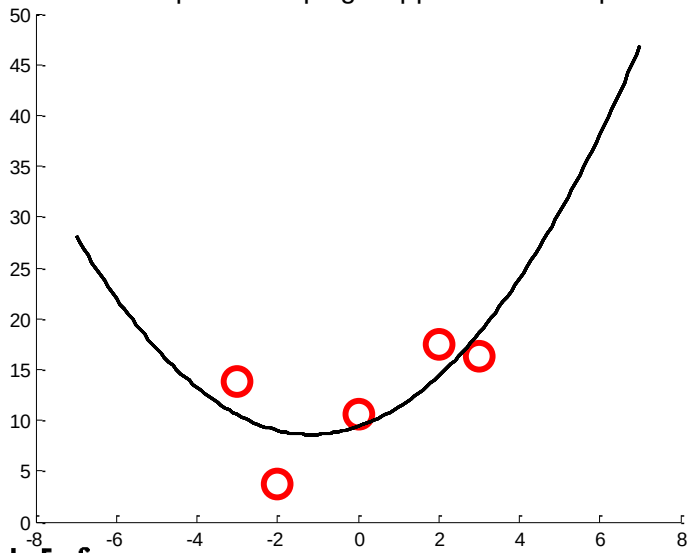


$$p(t|x, \boldsymbol{x}, \boldsymbol{t})$$
$$= \int p(t|x, \boldsymbol{w}) \delta_{\hat{\boldsymbol{w}}}(\boldsymbol{w}) d\boldsymbol{w}$$
$$= p(t|x, \hat{\boldsymbol{w}})$$

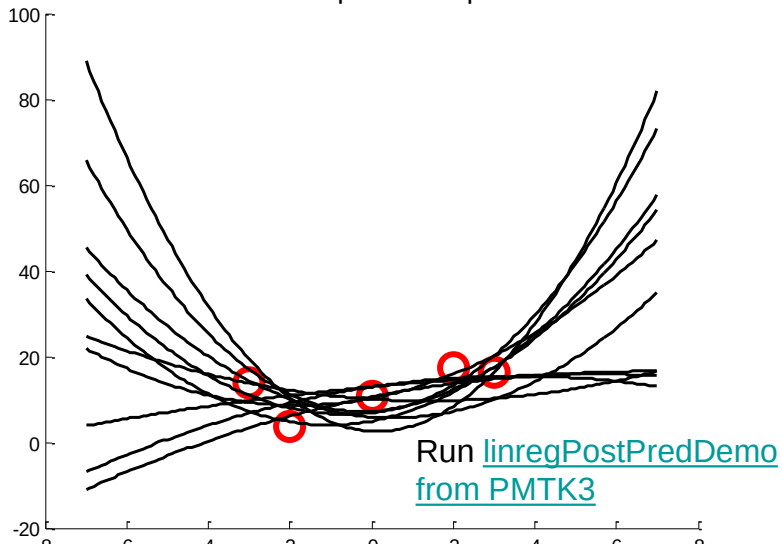
Posterior predictive (known variance)



functions sampled from plugin approximation to posterior

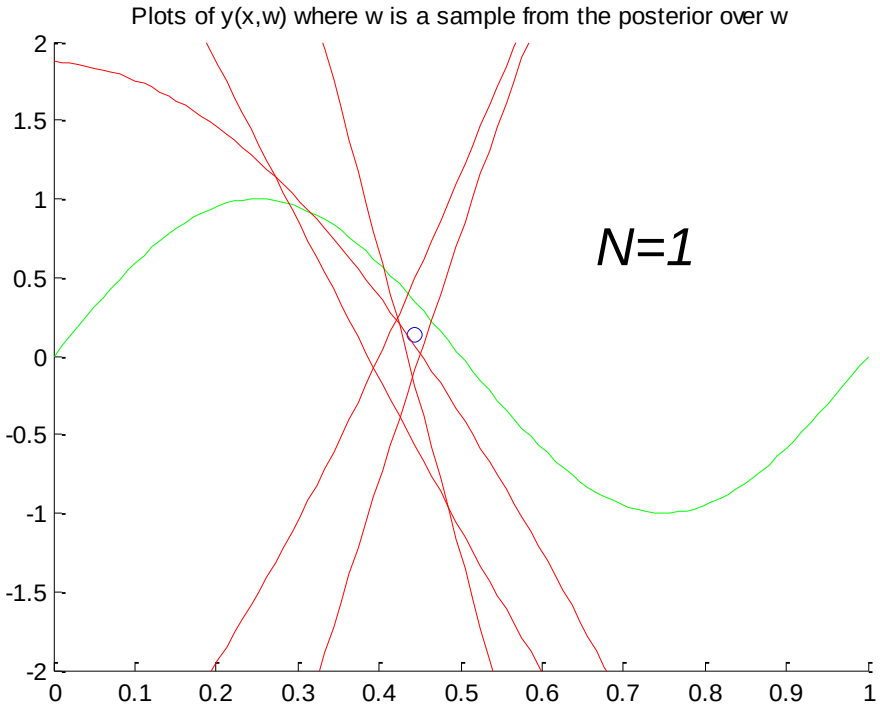


functions sampled from posterior

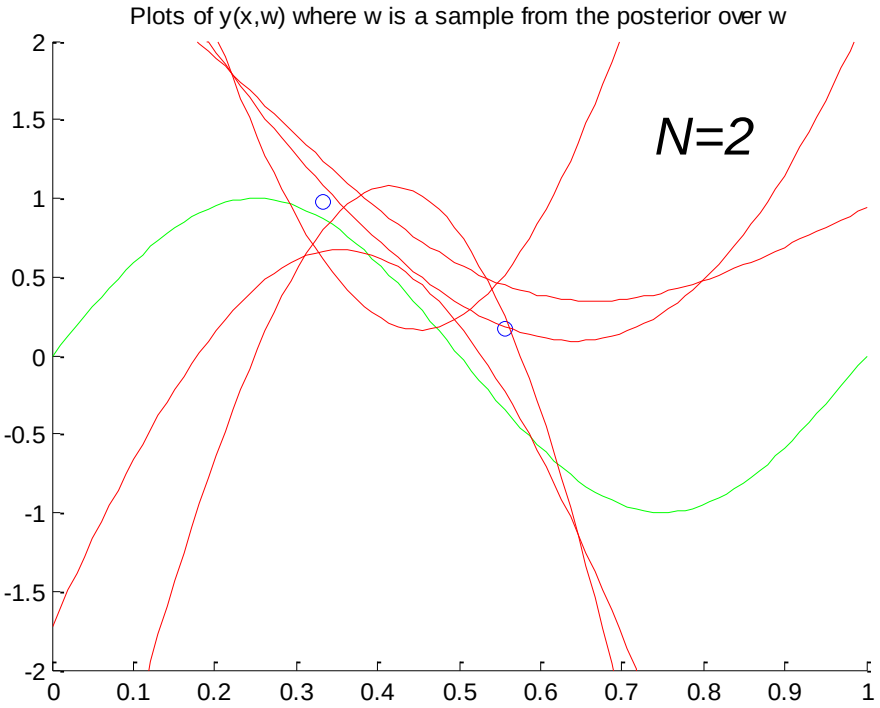


Run linregPostPredDemo from PMTK3

Covariance Between the Predictions



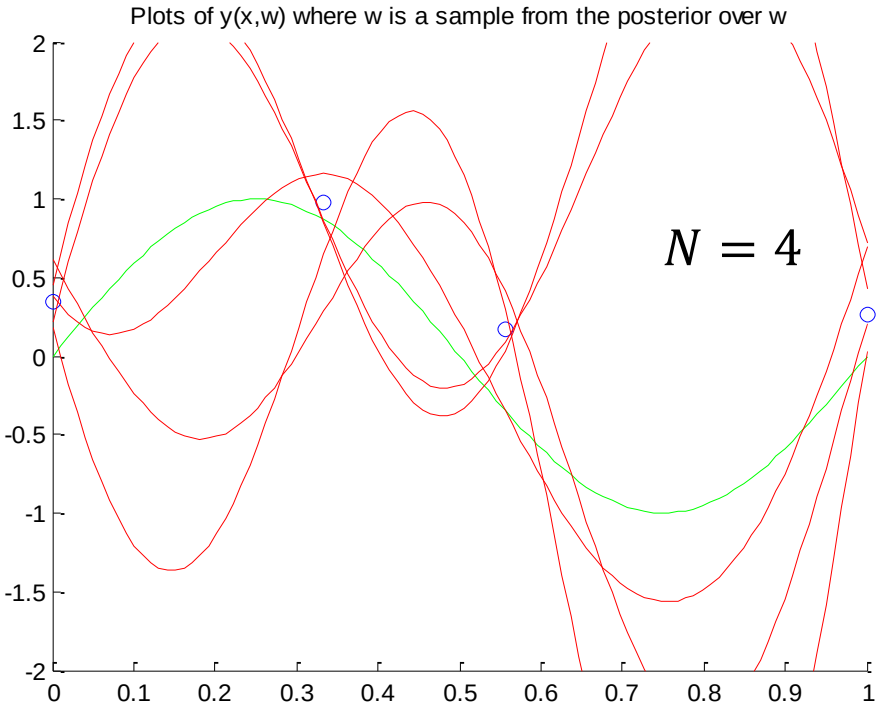
- Draw samples from the posterior of $\boldsymbol{w}$ and then plot $y(x, \boldsymbol{w})$. We use the same data as the earlier example.
- We are visualizing the joint uncertainty in the posterior distribution between the y values at two or more x values.



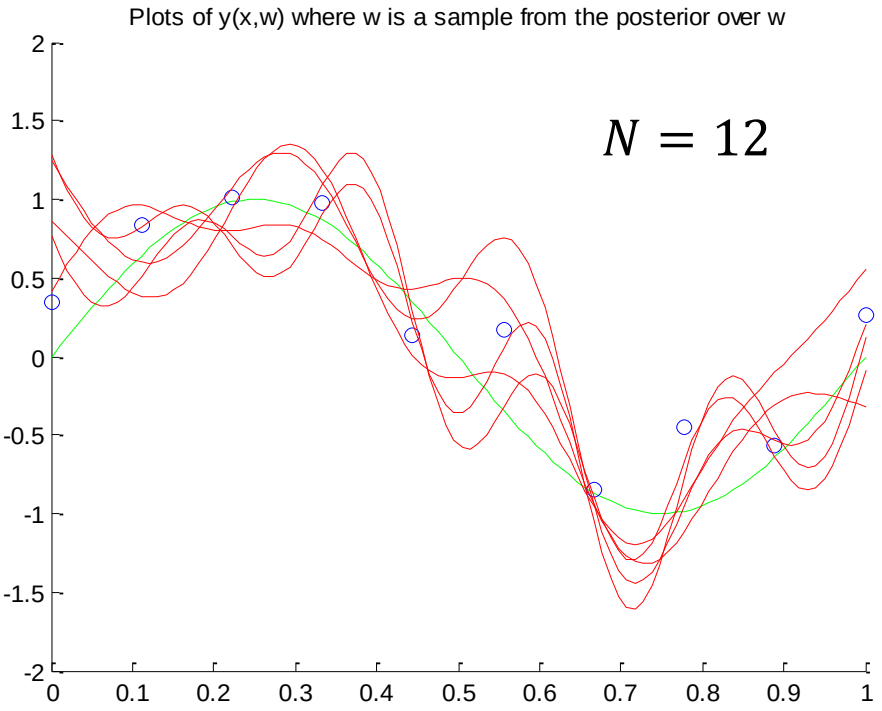
Same data and basis functions as in the earlier example

MatLab Code

Covariance Between the Predictions



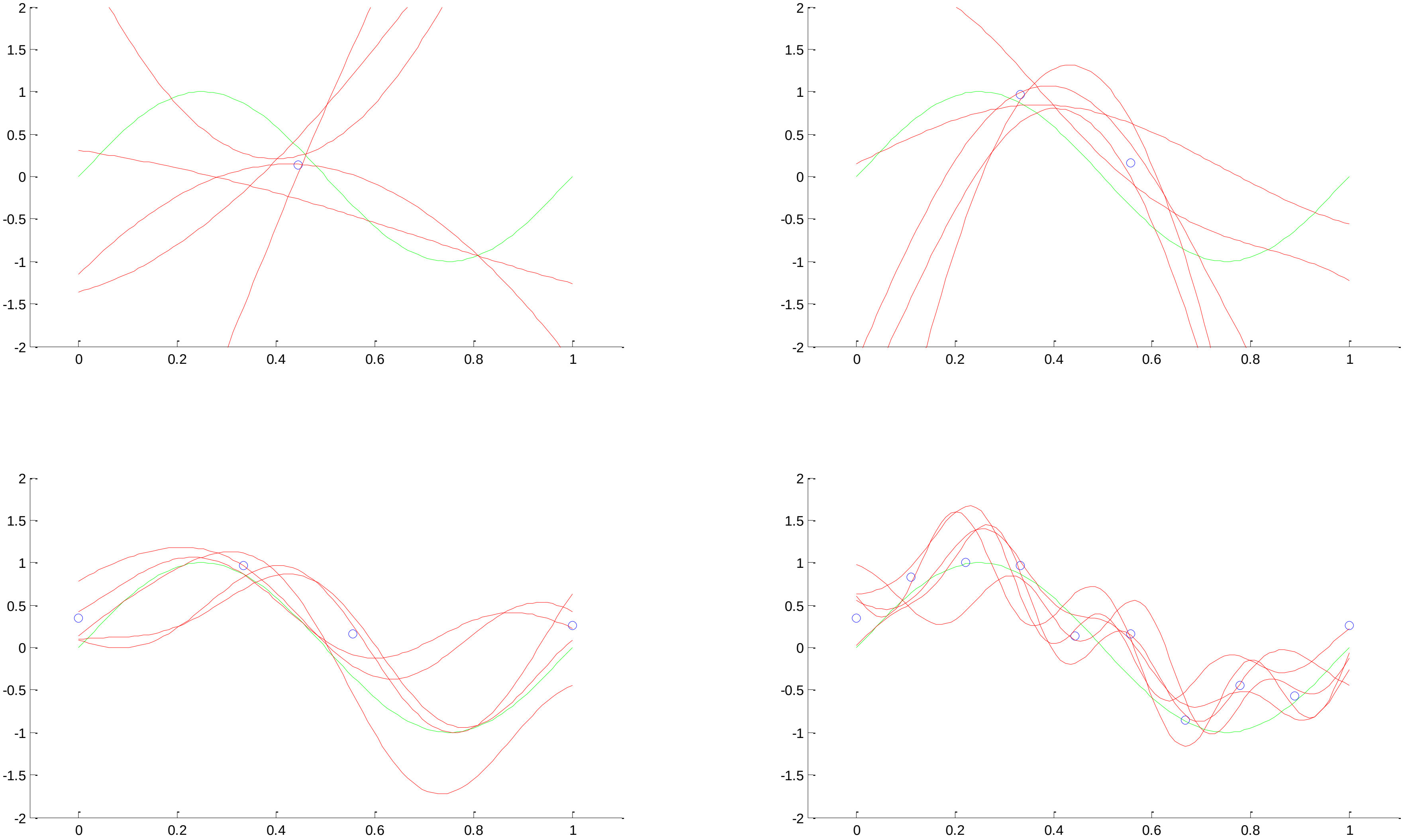
- Draw samples from the posterior of $\boldsymbol{w}$ and then plot $y(x, \boldsymbol{w})$
- We are visualizing the joint uncertainty in the posterior distribution between the y values at two or more x values.



MatLab Code

Summary of Results

MatLab Code



Gaussian Basis vs. Gaussian Process

- If we use localized basis functions such as Gaussians, then in regions away from the basis function support, the contribution from the second term in the predictive variance will go to zero, leaving only the noise contribution β^{-1}.

$$\sigma_N^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^T \boldsymbol{S}_N \boldsymbol{\phi}(x) \underset{\text{support of } \boldsymbol{\phi}(x)}{\overset{\text{away from the}}{\longrightarrow}} \beta^{-1}$$

- The model becomes very confident in its predictions when extrapolating outside the region occupied by the basis functions. This is an undesirable behavior.

- This problem can be avoided by adopting an alternative Bayesian approach to regression (Gaussian processes).

Bayesian Inference when σ^2 is Unknown

- Let us extend the previous results for linear regression assuming now that σ^2 is unknown.
- Assume a likelihood of the form:*

$$p\left(\boldsymbol{y}\,|\,\boldsymbol{X},\boldsymbol{w},\sigma^2\right)=\mathcal{N}\left(\boldsymbol{y}\,|\,\boldsymbol{X}\boldsymbol{w},\sigma^2\boldsymbol{I}_N\right)=\frac{1}{\left(2\pi\right)^{N/2}}\left(\sigma^2\right)^{-N/2}\exp\left(-\frac{\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)^T\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)}{2\sigma^2}\right)$$

- *A conjugate prior* has the following form:

$$p(\boldsymbol{w},\sigma^2)=\mathcal{NIG}(\boldsymbol{w},\sigma^2|\boldsymbol{w}_0,\boldsymbol{V}_0,a_0,b_0)\triangleq\mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_0,\sigma^2\boldsymbol{V}_0)\mathcal{IG}(\sigma^2|a_0,b_0)$$

$$\frac{b_0^{a_0}}{(2\pi)^{D/2}|V_0|^{1/2}\Gamma(a_0)}(\sigma^2)^{-(a_0+D/2+1)}\exp\left(-\frac{(\boldsymbol{w}-\boldsymbol{w}_0)^T\boldsymbol{V}_0^{-1}(\boldsymbol{w}-\boldsymbol{w}_0)+2b_0}{2\sigma^2}\right)$$

- The posterior is now derived as:

$$p\left(\boldsymbol{w},\sigma^2\,|\,\mathcal{D}\right)=\frac{b_0^{a_0}}{\left(2\pi\right)^{(N+D)/2}\left|\boldsymbol{V}_0\right|^{1/2}\Gamma\left(a_0\right)}\left(\sigma^2\right)^{-\left(a_0+(D+N)/2+1\right)}\exp\left(-\frac{\left(\boldsymbol{w}-\boldsymbol{w}_0\right)^T\boldsymbol{V}_0^{-1}\left(\boldsymbol{w}-\boldsymbol{w}_0\right)+\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)^T\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)+2b_0}{2\sigma^2}\right)$$

- In the remaining of this lecture, the response is denoted as $\boldsymbol{y}$, the dimensionality of $\boldsymbol{w}$ as D and the design matrix as $\boldsymbol{X}$.

Bayesian Inference when σ^2 is Unknown

$$p\left(\boldsymbol{w},\sigma^2 \mid \mathcal{D}\right)=\frac{b_0^{a_0}}{(2\pi)^{(N+D)/2}\left|\boldsymbol{V}_0\right|^{1/2}\Gamma(a_0)}\left(\sigma^2\right)^{-(a_0+(D+N)/2+1)}\exp\left(-\frac{\left(\boldsymbol{w}-\boldsymbol{w}_0\right)^T\boldsymbol{V}_0^{-1}\left(\boldsymbol{w}-\boldsymbol{w}_0\right)+\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)^T\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)+2b_0}{2\sigma^2}\right)$$

- Let us define the following:

$$\boldsymbol{V}_N=\left(\boldsymbol{V}_0^{-1}+\boldsymbol{X}^T\boldsymbol{X}\right)^{-1},\quad \boldsymbol{w}_N=\boldsymbol{V}_N\left(\boldsymbol{V}_0^{-1}\boldsymbol{w}_0+\boldsymbol{X}^T\boldsymbol{y}\right)$$

$$a_N=a_0+N/2,\quad b_N=b_0+\frac{1}{2}\left(\boldsymbol{w}_0^T\boldsymbol{V}_0^{-1}\boldsymbol{w}_0+\boldsymbol{y}^T\boldsymbol{y}-\boldsymbol{w}_N^T\boldsymbol{V}_N^{-1}\boldsymbol{w}_N\right)$$

- With these definitions, one with simple algebra can show:

$$p\left(\boldsymbol{w},\sigma^2 \mid \mathcal{D}\right)\propto\left(\sigma^2\right)^{-(a_N+D/2+1)}\exp\left(-\frac{\left(\boldsymbol{w}-\boldsymbol{w}_0\right)^T\boldsymbol{V}_0^{-1}\left(\boldsymbol{w}-\boldsymbol{w}_0\right)+\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)^T\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)+2b_N-\boldsymbol{w}_0^T\boldsymbol{V}_0^{-1}\boldsymbol{w}_0-\boldsymbol{y}^T\boldsymbol{y}+\boldsymbol{w}_N^T\boldsymbol{V}_N^{-1}\boldsymbol{w}_N}{2\sigma^2}\right)$$

$$\propto\left(\sigma^2\right)^{-(a_N+D/2+1)}\exp\left(-\frac{\left(\boldsymbol{w}-\boldsymbol{w}_N\right)^T\boldsymbol{V}_N^{-1}\left(\boldsymbol{w}-\boldsymbol{w}_N\right)+2b_N}{2\sigma^2}\right)$$

$$p(\boldsymbol{w},\sigma^2|\mathcal{D})=\mathcal{NIG}(\boldsymbol{w},\sigma^2|\boldsymbol{w}_N,\boldsymbol{V}_N,a_N,b_N)\triangleq\mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_N,\sigma^2\boldsymbol{V}_N)\mathcal{IG}(\sigma^2|a_N,b_N)$$

- The posterior marginals can now be derived explicitly:

$$p\left(\sigma^2 \mid \mathcal{D}\right)=\mathcal{IG}\left(\sigma^2 \mid a_N,b_N\right)$$

$$p\left(\boldsymbol{w} \mid \mathcal{D}\right)=\mathcal{T}_D\left(\boldsymbol{w}_N,\frac{b_N}{a_N}\boldsymbol{V}_N,2a_N\right)\propto\left[1+\frac{\left(\boldsymbol{w}-\boldsymbol{w}_N\right)^T\boldsymbol{V}_N^{-1}\left(\boldsymbol{w}-\boldsymbol{w}_N\right)}{2b_N}\right]^{-\frac{2a_N+D}{2}}$$

Posterior Marginals

$$p(\boldsymbol{w}\,|\,\mathcal{D}) \propto \int_0^{\infty} \left(\sigma^2\right)^{-(a_N+D/2+1)} \exp\left(-\frac{(\boldsymbol{w}-\boldsymbol{w}_N)^T \boldsymbol{V}_N^{-1}(\boldsymbol{w}-\boldsymbol{w}_N)+2b_N}{2\sigma^2}\right) d\sigma^2 \propto \left[1+\frac{(\boldsymbol{w}-\boldsymbol{w}_N)^T \boldsymbol{V}_N^{-1}(\boldsymbol{w}-\boldsymbol{w}_N)}{2b_N}\right]^{-\frac{2a_N+D}{2}}$$

- The marginal posterior can be directly written as:

$$p(\boldsymbol{w}\,|\,\mathcal{D}) = \mathcal{T}_D\left(\boldsymbol{w}_N, \frac{b_N}{a_N}\boldsymbol{V}_N, 2a_N\right) \propto \left[1+\frac{(\boldsymbol{w}-\boldsymbol{w}_N)^T \boldsymbol{V}_N^{-1}(\boldsymbol{w}-\boldsymbol{w}_N)}{2b_N}\right]^{-\frac{2a_N+D}{2}}$$

- To compute the integral above, simply set $\lambda = \sigma^{-2}, d\sigma^2 = -\lambda^{-2}d\lambda$ and use the normalizing factor of the Gamma distribution $\int_0^{\infty} \lambda^{a-1}e^{-b\lambda}d\lambda = \Gamma(a)b^{-a} \sim b^{-a}$.

Posterior Predictive Distribution

- Consider the posterior predictive for m new test inputs:

$$p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{X}},\mathcal{D}) \propto \iint \frac{1}{(2\pi)^{m/2}}(\sigma^2)^{-m/2}\exp\left(-\frac{(\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{X}}\boldsymbol{w})^T(\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{X}}\boldsymbol{w})}{2\sigma^2}\right)(\sigma^2)^{-(a_N+D/2+1)}\exp\left(-\frac{(\boldsymbol{w}-\boldsymbol{w}_N)^T\boldsymbol{V}_N^{-1}(\boldsymbol{w}-\boldsymbol{w}_N)+2b_N}{2\sigma^2}\right)d\boldsymbol{w}d\sigma^2$$

- As a first step, let us integrate in $\boldsymbol{w}$ by writing:

These terms cancel out from the integration in $\boldsymbol{w}$

$$(\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{X}}\boldsymbol{w})^T(\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{X}}\boldsymbol{w})+(\boldsymbol{w}-\boldsymbol{w}_N)^T\boldsymbol{V}_N^{-1}(\boldsymbol{w}-\boldsymbol{w}_N)+2b_N =$$

$$\left(\boldsymbol{w}-\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}+\boldsymbol{V}_N^{-1}\right)^{-1}\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{y}}+\boldsymbol{V}_N^{-1}\boldsymbol{w}_N\right)\right)^T\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}+\boldsymbol{V}_N^{-1}\right)\left(\boldsymbol{w}-\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}+\boldsymbol{V}_N^{-1}\right)^{-1}\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{y}}+\boldsymbol{V}_N^{-1}\boldsymbol{w}_N\right)\right)$$

$$-\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{y}}+\boldsymbol{V}_N^{-1}\boldsymbol{w}_N\right)^T\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}+\boldsymbol{V}_N^{-1}\right)^{-T}\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{y}}+\boldsymbol{V}_N^{-1}\boldsymbol{w}_N\right)+\boldsymbol{w}_N^T\boldsymbol{V}_N^{-1}\boldsymbol{w}_N+\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}}+2b_N$$

- Let us denote the last term in the Eq. above as

$$2\beta = -\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{y}}+\boldsymbol{V}_N^{-1}\boldsymbol{w}_N\right)^T\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}+\boldsymbol{V}_N^{-1}\right)^{-T}\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{y}}+\boldsymbol{V}_N^{-1}\boldsymbol{w}_N\right)+\boldsymbol{w}_N^T\boldsymbol{V}_N^{-1}\boldsymbol{w}_N+\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}}+2b_N$$

Posterior Predictive Distribution

- The posterior predictive

$$p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{X}},\mathcal{D})$$
$$\propto \iint \frac{1}{(2\pi)^{m/2}}(\sigma^2)^{-m/2}\exp\left(-\frac{(\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{X}}\boldsymbol{w})^T(\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{X}}\boldsymbol{w})}{2\sigma^2}\right)(\sigma^2)^{-(a_N+D/2+1)}\exp\left(-\frac{(\boldsymbol{w}-\boldsymbol{w}_N)^T\boldsymbol{V}_N^{-1}(\boldsymbol{w}-\boldsymbol{w}_N)+2b_N}{2\sigma^2}\right)d\boldsymbol{w}d\sigma^2$$

is now simplified using $\lambda = 1/\sigma^2$ and recalling the normalization of the Gamma distribution:

$$p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{X}},\mathcal{D}) \propto \int (\lambda)^{m/2+a_N-1}\exp(-\beta\lambda)d\lambda \sim \beta^{-(m/2+a_N)}$$

- Substituting β and by comparing the two Eqs. one can verify:

$$p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{X}},\mathcal{D}) \propto \left(-(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{y}}+\boldsymbol{V}_N^{-1}\boldsymbol{w}_N)^T(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}+\boldsymbol{V}_N^{-1})^{-T}(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{y}}+\boldsymbol{V}_N^{-1}\boldsymbol{w}_N)+\boldsymbol{w}_N^T\boldsymbol{V}_N^{-1}\boldsymbol{w}_N+\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}}+2b_N\right)^{-\left(\frac{m}{2}+a_N\right)}$$

$$\propto \left(1+\frac{(\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{X}}\boldsymbol{w}_N)^T\left(\frac{b_N}{a_N}(\boldsymbol{I}_m+\tilde{\boldsymbol{X}}\boldsymbol{V}_N\tilde{\boldsymbol{X}}^T)\right)^{-1}(\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{X}}\boldsymbol{w}_N)}{2a_N}\right)^{-\left(\frac{m}{2}+a_N\right)}$$

← Use the Sherman Morrison Woodbury formula here to show that (symmetry of $\boldsymbol{V}_0$ is assumed)

$$\left(\boldsymbol{I}_m+\tilde{\boldsymbol{X}}\boldsymbol{V}_N\tilde{\boldsymbol{X}}^T\right)^{-1} = \boldsymbol{I}_m-\tilde{\boldsymbol{X}}\left(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}+\boldsymbol{V}_N^{-1}\right)^{-1}\tilde{\boldsymbol{X}}^T$$

Bayesian Inference when σ^2 is Unknown

- The posterior predictive is also a Student's $\mathcal{T}$:

$$p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{X}}, \mathcal{D}) = \mathcal{T}_m\left(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{X}}\boldsymbol{w}_N, \frac{b_N}{a_N}\left(\boldsymbol{I}_m + \tilde{\boldsymbol{X}}\boldsymbol{V}_N\tilde{\boldsymbol{X}}^T\right), 2a_N\right)$$

- The predictive variance has two terms
  - $\frac{b_N}{a_N}\boldsymbol{I}_m$ due to the measurement noise
  - and $\frac{b_N}{a_N}\tilde{\boldsymbol{X}}\boldsymbol{V}_N\tilde{\boldsymbol{X}}^T$ due to the uncertainty in $\boldsymbol{w}$. *The second term depends on how close a test input is to the training data.*

Zellner's G-Prior

$$p(\boldsymbol{w}, \sigma^2) = \mathcal{NIG}(\boldsymbol{w}, \sigma^2 | \boldsymbol{w}_0, \boldsymbol{V}_0, a_0, b_0) \triangleq \mathcal{N}(\boldsymbol{w} | \boldsymbol{w}_0, \sigma^2 \boldsymbol{V}_0) \mathcal{IG}(\sigma^2 | a_0, b_0)$$

- It is common to set $a_0 = b_0 = 0$, corresponding to an uninformative prior for σ^2, and *to set* $\boldsymbol{w}_0 = \boldsymbol{0}$ *and* $\boldsymbol{V}_0 = g(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ *for any positive value* g.
- This is called Zellner's **g-prior**. Here g plays a role analogous to $1/\lambda$ in ridge regression. However, the prior covariance is proportional to $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ rather than $\boldsymbol{I}$.

$$p(\boldsymbol{w}, \sigma^2) = \mathcal{NIG}(\boldsymbol{w}, \sigma^2 | 0, g(\boldsymbol{X}^T\boldsymbol{X})^{-1}, 0, 0) \triangleq \mathcal{N}(\boldsymbol{w} | 0, \sigma^2 g(\boldsymbol{X}^T\boldsymbol{X})^{-1}) \mathcal{IG}(\sigma^2 | 0, 0)$$

- This ensures that the posterior is invariant to scaling of the inputs.

- Zellner, A. (1986). On assessing prior distributions and bayesian regression analysis with g-prior distributions. In *Bayesian inference and decision techniques*, *Studies of Bayesian and Econometrics and Statistics volume 6*. North Holland.
- Minka, T. (2000b). Bayesian linear regression. Technical report, MIT.

Unit Information Prior

$$p(\boldsymbol{w}, \sigma^2) = \mathcal{NIG}(\boldsymbol{w}, \sigma^2 | 0, g(\boldsymbol{X}^T\boldsymbol{X})^{-1}, 0, 0) \triangleq N(\boldsymbol{w}|0, \sigma^2 g(\boldsymbol{X}^T\boldsymbol{X})^{-1}) InvGamma(\sigma^2|0,0)$$

- We will see below that if we use an uninformative prior, the posterior precision given N measurements is $\boldsymbol{V}_N^{-1} = \boldsymbol{X}^T\boldsymbol{X}$.

- The **unit information prior** is defined to contain as much information as one sample.

- *To create a unit information prior for linear regression, we need to use* $\boldsymbol{V}_0^{-1} = \frac{1}{N}\boldsymbol{X}^T\boldsymbol{X}$ *which is equivalent to the g-prior with* $g = N$.

- *Zellner's prior depends on the data: This is contrary to much of our Bayesian inference discussion!*

  - Kass, R. and L. Wasserman (1995). A reference bayesian test for nested hypotheses and its relationship to the schwarz criterio. *J. of the Am. Stat. Assoc. 90*(431), 928–934.

Uninformative Prior

- An uninformative prior can be obtained by considering the *uninformative limit of the conjugate g-prior, which corresponds to setting* $g = \infty$. This is equivalent to an improper $\mathcal{NIG}$ prior with $\boldsymbol{w}_0 = 0, \boldsymbol{V}_0 = \infty \boldsymbol{I}$, $a_0 = 0$ and $b_0 = 0$, which gives $p(\boldsymbol{w}, \sigma^2) \propto \sigma^{-(D+2)}$.

$$p(\boldsymbol{w}, \sigma^2) = \mathcal{NIG}(\boldsymbol{w}, \sigma^2 | 0, \infty \boldsymbol{I}, 0, 0) \triangleq \mathcal{N}(\boldsymbol{w} | 0, \sigma^2 \infty I) \mathcal{IG}(\sigma^2 | 0, 0) \rightarrow \sigma^{-(D+2)}$$

- Alternatively, we can start with the *semi-conjugate prior* $p(\boldsymbol{w}, \sigma^2) = p(\boldsymbol{w}) p(\sigma^2)$, *and take each term to its uninformative limit individually*, which gives $p(\boldsymbol{w}, \sigma^2) \propto \sigma^{-2}$.

  This is equivalent to an improper $\mathcal{NIG}$ prior with $\boldsymbol{w}_0 = \boldsymbol{0}, \boldsymbol{V} = \infty \boldsymbol{I}, a_0 = -D/2$ and $b_0 = 0$.

$$p(\boldsymbol{w}, \sigma^2) = \mathcal{NIG}(\boldsymbol{w}, \sigma^2 | 0, \infty \boldsymbol{I}, 0, 0) \triangleq \mathcal{N}(\boldsymbol{w} | 0, \sigma^2 \infty I) \mathcal{IG}\left(\sigma^2 | -\frac{D}{2}, 0\right) \rightarrow \sigma^{-2}$$

Uninformative Prior

- Using the uninformative prior, $p\left(w,\sigma^2\right)\propto\sigma^{-2}$, the corresponding posterior and marginal posteriors are given by

$$p\left(\boldsymbol{w},\sigma^2\mid\mathcal{D}\right)=\mathcal{NIG}\left(\boldsymbol{w},\sigma^2\mid\boldsymbol{w}_N,\boldsymbol{V}_N,a_N,b_N\right)$$

$$p(\boldsymbol{w}|\mathcal{D})=\mathcal{T}_D\left(\boldsymbol{w}_N,\frac{b_N}{a_N}\boldsymbol{V}_N,2a_N\right)=\mathcal{T}_D\left(\boldsymbol{w}|\hat{\boldsymbol{w}}_{MLE},\frac{s^2}{N-D}\boldsymbol{C},N-D\right)$$

$$\boldsymbol{V}_N=\boldsymbol{C}=(\boldsymbol{V}_0^{-1}+\boldsymbol{X}^T\boldsymbol{X})^{-1}\rightarrow(\boldsymbol{X}^T\boldsymbol{X})^{-1},\quad \boldsymbol{w}_N=\boldsymbol{V}_N(\boldsymbol{V}_0^{-1}\boldsymbol{w}_0+\boldsymbol{X}^T\boldsymbol{y})\rightarrow(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}=\hat{\boldsymbol{w}}_{MLE}$$

$$a_N=a_0+N/2=(N-D)/2\,,$$

$$\boldsymbol{b}_N=\boldsymbol{b}_0+\frac{1}{2}(\boldsymbol{w}_0^T\boldsymbol{V}_0^{-1}\boldsymbol{w}_0+\boldsymbol{y}^T\boldsymbol{y}-\boldsymbol{w}_N^T\boldsymbol{V}_N^{-1}\boldsymbol{w}_N)=s^2/2\,,\ s^2=\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{w}}_{MLE}\right)^T\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{w}}_{MLE}\right)$$

$$\boldsymbol{w}_N=\hat{\boldsymbol{w}}_{MLE}=(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

- Note in the calculation of s^2:

$$s^2=\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{w}}_{MLE}\right)^T\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{w}}_{MLE}\right)=(\boldsymbol{y}-\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y})^T(\boldsymbol{y}-\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y})$$
$$=\boldsymbol{y}^T\boldsymbol{y}-\boldsymbol{y}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}=\boldsymbol{y}^T\boldsymbol{y}-\hat{\boldsymbol{w}}_{MLE}^T(\boldsymbol{X}^T\boldsymbol{X})(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}=\boldsymbol{y}^T\boldsymbol{y}-\hat{\boldsymbol{w}}_{MLE}^T\boldsymbol{V}_N^{-1}\hat{\boldsymbol{w}}_{MLE}$$

Frequentist Confidence Interval Vs. Bayesian Marginal Credible Interval

- *The use of a (semi-conjugate) uninformative prior is quite interesting since the resulting posterior turns out to be equivalent to the results obtained from frequentist statistics.*

$$p(\boldsymbol{w_j}|D) = \mathcal{T}\left(w_j|\hat{w}_j, \frac{C_{jj}s^2}{N-D}, N-D\right)$$

- This is equivalent to *the sampling distribution of the MLE* which is given by the following:

$$\frac{w_j - \hat{w}_j}{s_j} \sim \mathcal{T}_{N-D},\ s_j = \sqrt{\frac{C_{jj}s^2}{N-D}}$$

is the standard error of the estimated parameter.

- *The frequentist confidence interval and the Bayesian marginal credible interval for the parameters are the same.*

- Rice, J. (1995). *Mathematical statistics and data analysis*. Duxbury. 2nd edition (page 542)
- Casella, G. and R. Berger (2002). *Statistical inference*. Duxbury. 2nd edition (page 554)

The Caterpillar Example

- As a worked example of the uninformative prior, consider the caterpillar dataset. We can compute the posterior mean and standard deviation, and the 95% credible intervals (CI) for the regression coefficients.
- The 95% credible intervals are identical to the 95% confidence intervals computed using standard frequentist methods.

coeff	mean	stddev	95pc CI		sig
w0	10.998	3.06027	[4.652,	17.345]	*
w1	-0.004	0.00156	[-0.008,	-0.001]	*
w2	-0.054	0.02190	[-0.099,	-0.008]	*
w3	0.068	0.09947	[-0.138,	0.274]	
w4	-1.294	0.56381	[-2.463,	-0.124]	*
w5	0.232	0.10438	[0.015,	0.448]	*
w6	-0.357	1.56646	[-3.605,	2.892]	
w7	-0.237	1.00601	[-2.324,	1.849]	
w8	0.181	0.23672	[-0.310,	0.672]	
w9	-1.285	0.86485	[-3.079,	0.508]	
w10	-0.433	0.73487	[-1.957,	1.091]	

Run linregBayesCaterpillar from PMTK3

- Marin, J.-M. and C. Robert (2007). *Bayesian Core: a practical approach to computational Bayesian statistics*. Springer.

The Caterpillar Example

- We can use these marginal posteriors to compute if the coefficients are significantly different from 0 -- check if its 95% CI excludes 0.

- The CIs for coefficients $0, 1, 2, 4, 5$ are all significant.

- These results are the same as those produced by a frequentist approach using p-values at the 5% level.

- But note that the MLE does not even exist when $N < D$, so standard frequentist inference theory breaks down in this setting. Bayesian inference theory still works using proper priors.

- Maruyama, Y. and E. George (2008). A g-prior extension for *p* > *n*. Technical report, U. Tokyo.

Empirical Bayes for Linear Regression

- We describe next an empirical Bayes procedure for picking the hyper-parameters in the prior (we will come back to this and relevance determination in a forthcoming lecture).

- More precisely, we choose $\boldsymbol{\eta} = (\alpha, \lambda)$ to maximize the marginal likelihood, where $\lambda = 1/\sigma^2$ be the precision of the observation noise and $\boldsymbol{\alpha}$ is the precision of the prior, $p(\boldsymbol{w}) = N(\boldsymbol{w}|\boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$.

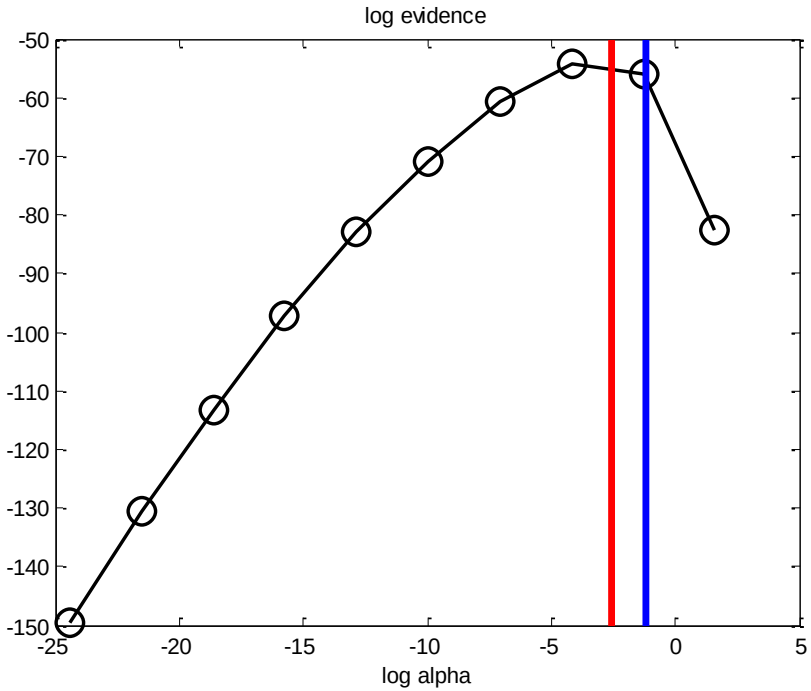
- This is known as the *evidence procedure*.

- MacKay, D. (1995b). Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks. *Network*.
- Buntine, W. and A. Weigend (1991). Bayesian backpropagation. *Complex Systems 5*, 603–643.
- MacKay, D. (1999). Comparision of approximate methods for handling hyperparameters. *Neural Computation 11*(5), 1035–1068.

Empirical Bayes for Linear Regression

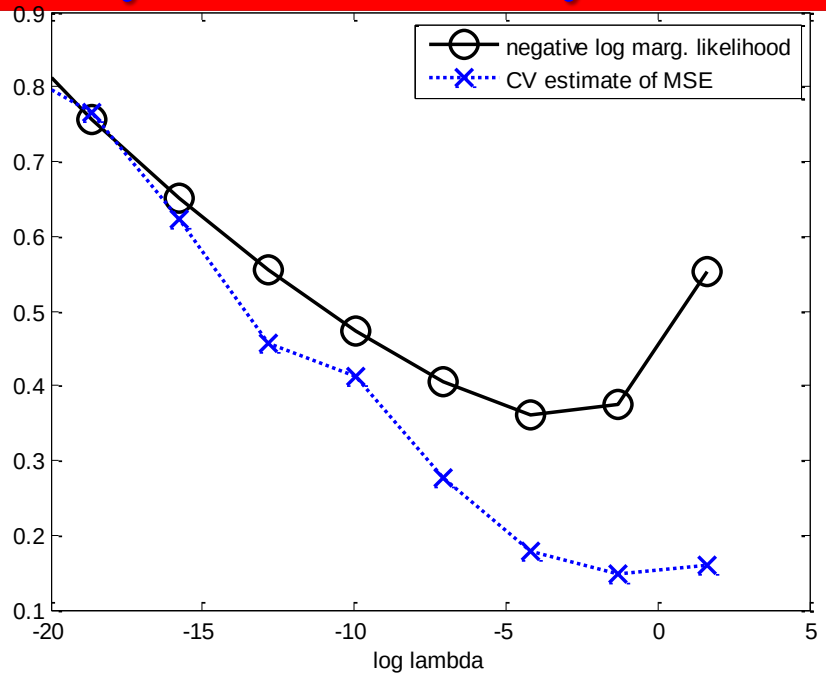
- The evidence procedure provides an alternative to using cross validation.

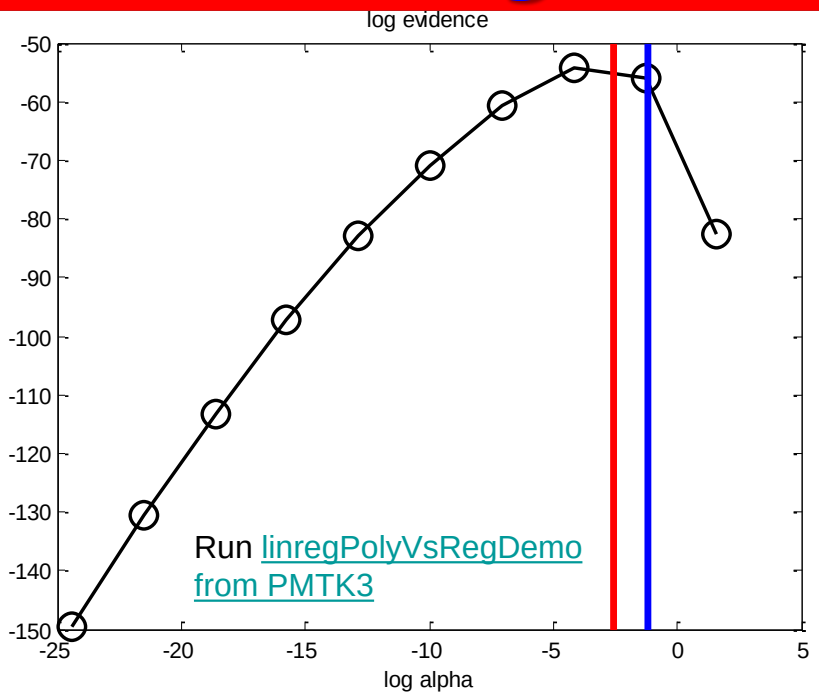
- In the Figure, the log marginal likelihood is plotted for different values of α, as well as the maximum value found by the optimizer.



Run linregPolyVsRegDemo from PMTK3

Empirical Bayes for Linear Regression





- We obtain the same result as 5-CV ($\lambda = 1/\sigma^2$ is fixed in both methods).
- The key advantage of the evidence procedure over CV is that it allows different α_j to be used for every feature.

Automatic Relevancy Determination

- The evidence procedure can be used to perform feature selection (automatic relevancy determination or ARD)

- The evidence procedure is also useful when comparing different kinds of models:

$$p(\mathcal{D} \mid m) = \iint p(\mathcal{D} \mid \boldsymbol{w}, m)\, p(\boldsymbol{w} \mid m, \boldsymbol{\eta})\, p(\boldsymbol{\eta} \mid m) d\boldsymbol{w} d\boldsymbol{\eta}$$
$$\approx \max_{\boldsymbol{\eta}} p(\mathcal{D} \mid \boldsymbol{w}, m)\, p(\boldsymbol{w} \mid m, \boldsymbol{\eta})\, p(\boldsymbol{\eta} \mid m) d\boldsymbol{w}$$

- It is important to (at least approximately) integrate over $\boldsymbol{\eta}$ rather than setting it arbitrarily.

- Using variation Bayes models our uncertainty on $\boldsymbol{\eta}$ rather than computing point estimates.