[Spring, 2017]

Linear Models for Regression

Pattern Recognition (BRI623)

Heung-Il Suk

hisuk@korea.ac.kr

http://www.ku-milab.org

Department of Brain and Cognitive Engineering,
Korea University

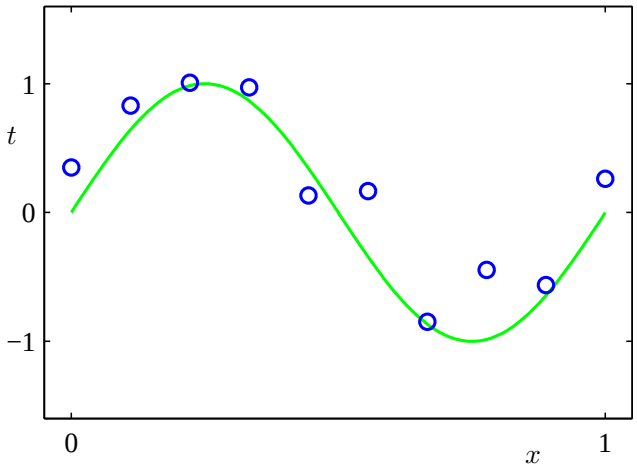
Contents

1. Introduction
2. Linear Basis Function Models
3. Bias-Variance Decomposition
4. Bayesian Linear Regression
5. Bayesian Model Comparison

Introduction

Regression

- Given a training data set of N observations $\{\mathbf{x}_n, t_n\}_{n=1}^{N}$, to predict the value of t for a new value of $\mathbf{x}$



Approaches

1. Directly constructing an appropriate function $f(\mathbf{x})$ such that $t = f(\mathbf{x})$
2. Modeling the predictive distribution $p(t|\mathbf{x})$ because this expresses our uncertainty about the value of t for each value of $\mathbf{x}$
   - From this conditional distribution, make predictions of t in a way as to minimize the expected value of a suitably chosen loss function, e.g., squared loss

Linear Basis Function Models

Linear Basis Function Models

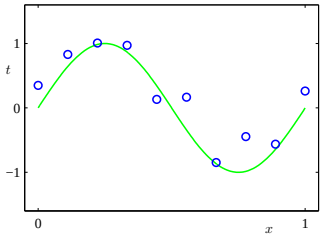
- Linear regression: simplest model for regression
  - Linear combination of input variables

$$y(\mathbf{x}, \mathbf{w}) = \sum_{d=1}^{D} w_d x_d + w_0$$

  - Limited as practical techniques for pattern recognition (e.g., high dimensionality)
  - Nice analytical properties; foundation for more sophisticated models

- More useful form: polynomial curve fitting

$$y(x, \mathbf{w}) = \sum_{j=1}^{M-1} w_j x^j + w_0$$



- Linear combination of non-linear functions of input variables $\phi(\mathbf{x})$, called '*basis functions*'

$$\begin{aligned} y(x, \mathbf{w}) &= \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) + w_0 = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) \\ &= \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) \end{aligned}$$

where $\mathbf{w} = [w_0, w_1, \ldots, w_{M-1}], \boldsymbol{\phi}(\mathbf{x}) = [\phi_0 = 1, \phi_1, \ldots, \phi_{M-1}]$

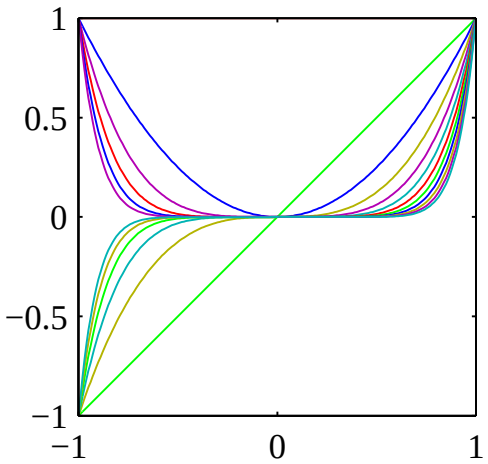
  - $\boldsymbol{\phi}(\mathbf{x})$: fixed preprocessing or feature extraction

Linear functions of parameters (still analytic);
Yet, non-linear with respect to the input variables

(Recap.) polynomial curve fitting

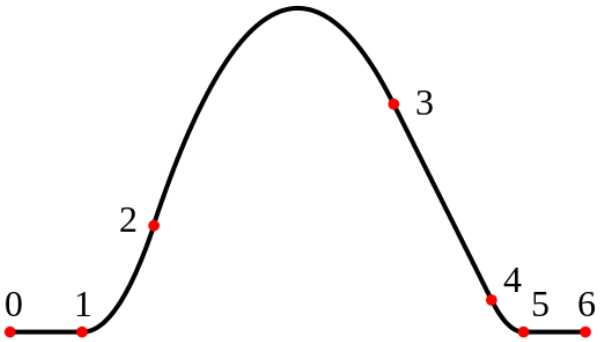
$$y(x, \mathbf{w}) = \sum_{j=1}^{M-1} w_j x^j + w_0$$

- Global function of the input variables: changes in one region of input space affect all other regions
- Difficult to formulate: number of polynomials/coefficients increases exponentially with M



Divide the input space into regions and
use different polynomials in each region!!!

Spline Function (from Wikipedia)

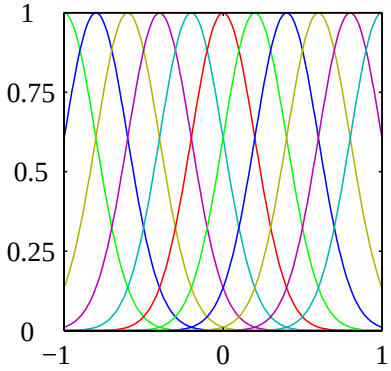


A quadratic spline composed of six polynomial segments. Between point 0 and point 1 a straight line. Between point 1 and point 2 a parabola with second derivative = 4. Between point 2 and point 3 a parabola with second derivative = -2. Between point 3 and point 4 a straight line. Between point 4 and point 5 a parabola with second derivative = 6. Between point 5 and point 6 a straight line.

Other Basis Functions

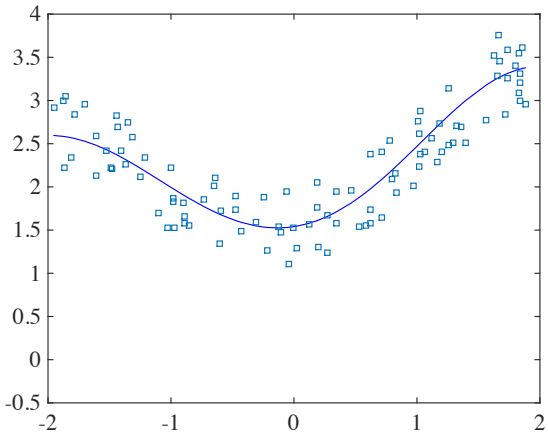
- (Gaussian) Radial Basis Functions (RBF)

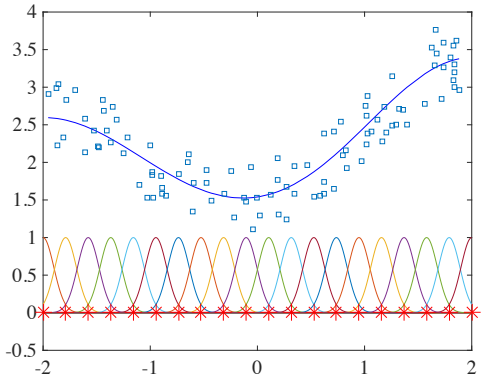
$$\phi_j(x) = \exp\left\{\frac{(x-\mu_j)^2}{2s^2}\right\}$$

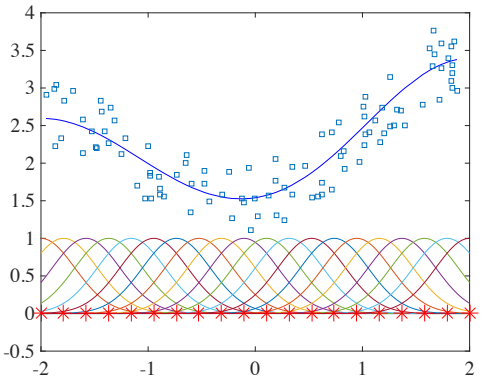


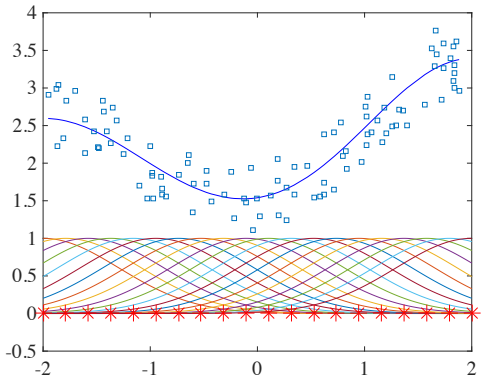
  - μ_j: governing the locations of the basis functions in input space
    - Can be arbitrary points in the data
  - s: governing the spatial scale
    - Can be chosen from the data set, *e.g.*, average variance

- Not required to have a probabilistic interpretation (normalization term is unimportant)









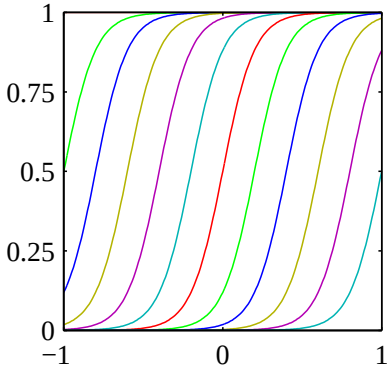
- Sigmoidal Basis Function

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

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

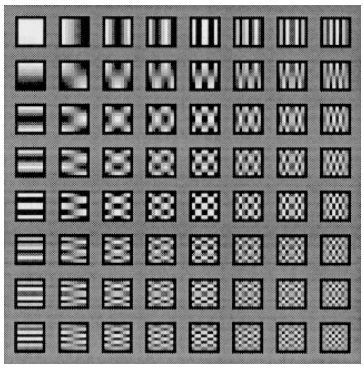
- Equivalently, 'tanh' function, which is related to the logistic sigmoid

$$\tanh(a) = 2\sigma(a) - 1$$



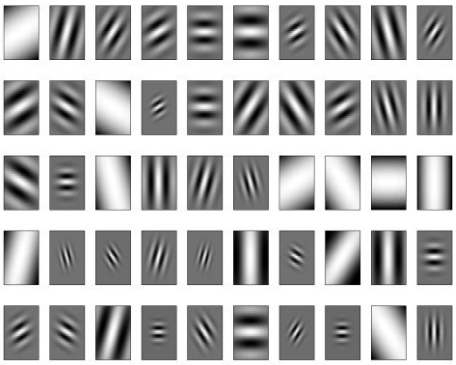
  - A general linear combination of logistic sigmoid functions is equivalent to a general linear combination of 'tanh' functions.

- Fourier
  - Expansion in sinusoidal functions
  - Infinite spatial extent



e.g., DCT Fourier basis

- Wavelet
  - Localized in both space and frequency
  - Useful for lattices such as images and time series



e.g., Gabor wavelet basis

Maximum Likelihood and Least Squares

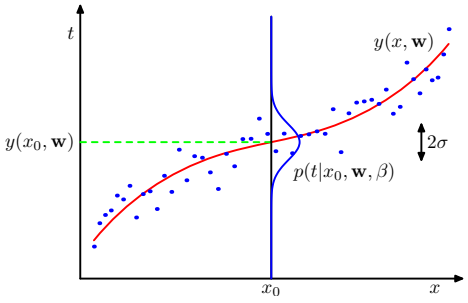
"Minimizing sum-of-squared errors is the same as maximum likelihood solution under a Gaussian noise model"

- Target variable is a scalar t given by a deterministic function $y(\mathbf{x}, \mathbf{w})$ with additive Gaussian noise

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \qquad \epsilon \sim \mathcal{N}\left(0, \beta^{-1}\right)$$

- Distribution of t is univariate normal

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



For a squared loss function, the optimal prediction, for a new value of $\mathbf{x}$, will be given by the conditional mean of the target variable.

- Gaussian (unimodal) case:

$$\mathbb{E}[t|\mathbf{x}] = \int tp(t|\mathbf{x})\, dt = y(\mathbf{x}, \mathbf{w})$$

- What if under multi-modal conditional distributions, *e.g.*, mixture of conditional Gaussian distributions? (discussed in Chapter 14.5)

- Data set: $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with their corresponding target values $\mathbf{t} = \{t_1, \ldots, t_N\}$
- Likelihood of the target data: probability of observing the data assuming they are independent

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

- Log-likelihood ($\mathbf{X}$: omitted for uncluttered)

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

$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left\{t_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\right\}^2$: sum-of-squares error function

Maximizing $\ln p(\mathbf{t}|\mathbf{w},\beta) \equiv$ minimizing $E_D(\mathbf{w})$

(under Gaussian noise distribution with a linear model)

- Taking derivative of $\ln p(\mathbf{t}|\mathbf{w},\beta)$ w.r.t. $\mathbf{w}$

$$\nabla \ln p(\mathbf{t}|\mathbf{w},\beta) = \beta \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^\top \phi(\mathbf{x}_n) \right\} \phi(\mathbf{x}_n)^\top$$

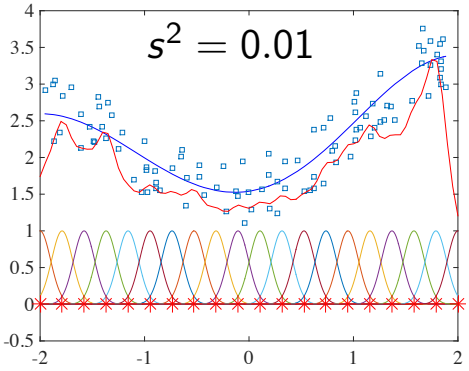
- Setting the gradient to zero

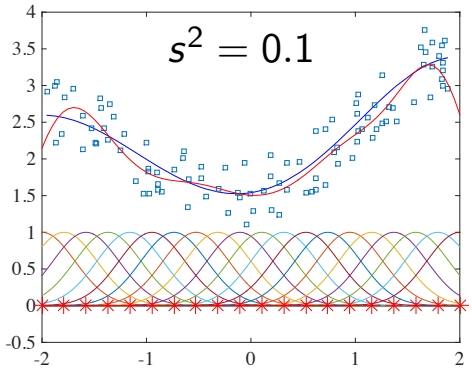
$$0 = \sum_{n=1}^{N} t_n \phi(\mathbf{x}_n)^\top - \mathbf{w}^\top \left(\sum_{n=1}^{N} \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^\top \right)$$

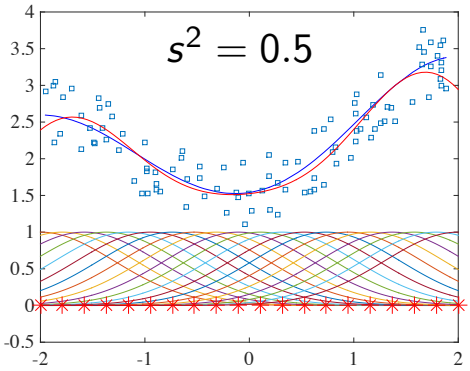
$$\therefore \mathbf{w}_{\mathrm{ML}} = \underbrace{\left(\Phi^\top \Phi\right)^{-1} \Phi^\top}_{\equiv \Phi^\dagger} \mathbf{t} = \Phi^\dagger \mathbf{t} \quad (\Phi^\dagger\text{: Moore-Penrose pseudo-inverse of } \Phi$$

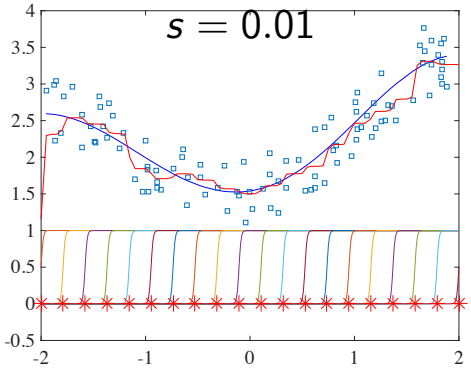
where $\Phi = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$, called '*design matrix*'

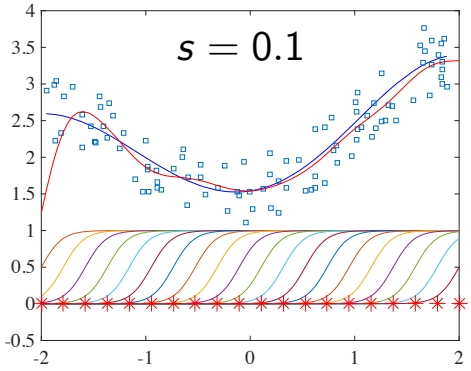
$(M-1)$ basis functions, *i.e.*, Gaussians entered on $(M-1)$ data points

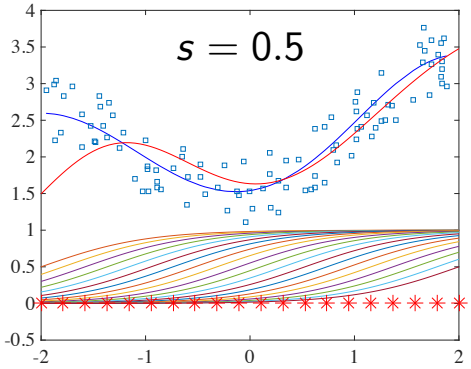












[Insight into the role of the bias parameter w_0]

- Explicit bias parameter

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

$$\nabla_{w_0} \ln p(\mathbf{t}|\mathbf{w},\beta) = \sum_{n=1}^{N}\left\{t_n - w_0 - \sum_{j=1}^{M-1} w_j\phi_j(\mathbf{x}_n)\right\} = 0$$

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

$$
\begin{aligned}
w_0 &= \frac{1}{N}\sum_{n=1}^{N} t_n - \sum_{j=1}^{M-1} w_j \left(\frac{1}{N}\sum_{n=1}^{N} \phi_j(\mathbf{x}_n) \right) \\
&= \bar{t} - \sum_{j=1}^{M-1} w_j \bar{\phi}_j
\end{aligned}
$$

$$
\bar{t} = \frac{1}{N}\sum_{n=1}^{N} t_n \qquad \bar{\phi}_j = \frac{1}{N}\sum_{n=1}^{N} \phi_j(\mathbf{x}_n)
$$

- w_0 compensate for the difference between the averages of the target values (over the training set) and the weighted sum of the averages of the basis function values

[Maximum likelihood for precision β]

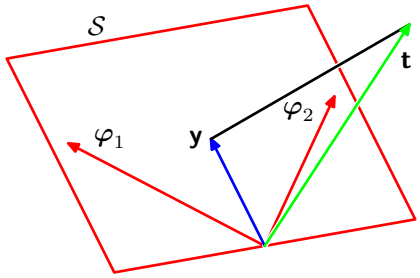
$$
\nabla_\beta \ln p(\mathbf{t}|\mathbf{w},\beta) = \frac{N}{2\beta} - E_D(\mathbf{w}) = 0
$$

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

- Inverse of the noise precision gives 'residual variance' of the target vales around the regression function

Geometry of Least Squares

- N-dimensional space whose axes are given by t_n
- $\mathbf{t}$ is a vector in this space
- Each basis function $\phi_j(\mathbf{x}_n)$ corresponding to j-th column of Φ is represented in this space as vector φ_j
- If $M < N$ then $\{\varphi_j\}_{j=1}^{M}$ will span a linear subspace S of dimensionality M
- Let $\mathbf{y}$ denote an N-dimensional vector, whose n-th element is given by $y(\mathbf{x}_n, \mathbf{w})$
- Since $\mathbf{y}$ is an arbitrary linear combination of the vectors φ_j, it can live anywhere in the M-dimensional subspace that is closest to $\mathbf{t}$
- Least-squares solution corresponds to orthogonal projection of $\mathbf{t}$ onto S



What if two or more of the basis vectors φ_j are co-linear, or nearly so (quite common in practice)

- Degeneracy problem

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

- Can be handled using '*Singular Value Decomposition*' (SVD)

- Adding a regularization term to ensure that the matrix Φ is non-singular, even in the presence of degeneracies

$$\mathbf{w}_{ML} = \left(\Phi^\top \Phi + \lambda I\right)^{-1} \Phi^\top \mathbf{t}$$

Sequential (On-line) Learning

- Batch learning: processing the entire training set in one go

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

  - Computationally costly for large datasets due to the huge design matrix Φ
- Sequential (or on-line) learning: samples are presented one at a time
  (by applying the technique of '*stochastic gradient descent*')
  - Denoting $E_D = \sum_n E_n$

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta\nabla E_n$$

  - Start with an initial vector $\mathbf{w}^{(0)}$
  - η should be chosen with care for convergence
  - For the case of sum-of-squares error function

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta\left(t_n - \mathbf{w}^{(\tau)T}\phi\left(\mathbf{x}_n\right)\right)\phi\left(\mathbf{x}_n\right)$$

    known as '*Least Mean Squares (LMS)*' algorithm

Regularized Least Squares

Regularization term in order to control overfitting

- Error function to minimize

$$E\left(\mathbf{w}\right) = E_D\left(\mathbf{w}\right) + \lambda E_{\mathbf{w}}\left(\mathbf{w}\right)$$

  - λ (regularization coefficient): controls relative importance of a data-dependent error $E_D\left(\mathbf{w}\right)$ and a regularization term $E_{\mathbf{w}}\left(\mathbf{w}\right)$

- "Quadratic" regularizer

$$E(\mathbf{w}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left\{t_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\right\}^2}_{E_D(\mathbf{w})} + \lambda \underbrace{\frac{1}{2}\mathbf{w}^\top \mathbf{w}}_{E_{\mathbf{w}}(\mathbf{w})}$$

  - a.k.a., '*weight decay*': in sequential learning, encouraging weight values to decay towards zero, unless supported by data
  - Error function remains a quadratic function of $\mathbf{w}$ $\rightarrow$ exact minimizer can be found in a closed form

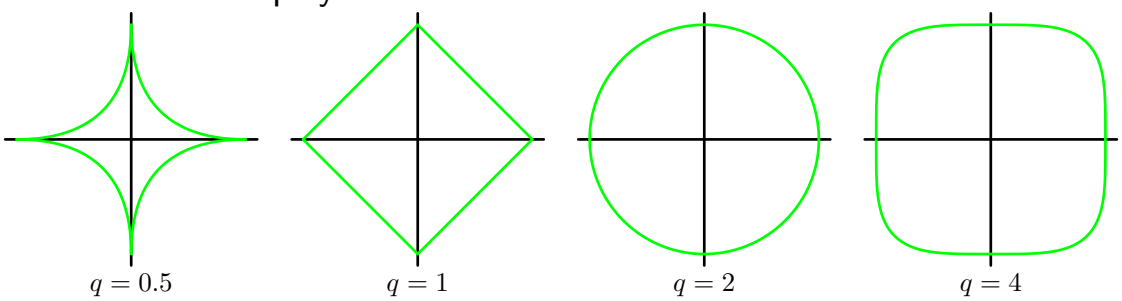
- Taking derivative w.r.t. $\mathbf{w}$ and setting to zero

$$\nabla^R_{\mathbf{w}} \ln p(\mathbf{t}|\mathbf{w}, \beta) \Rightarrow \mathbf{w}^R_{ML} = \left(\Phi^\top \Phi + \lambda I\right)^{-1} \Phi^\top \mathbf{t}$$

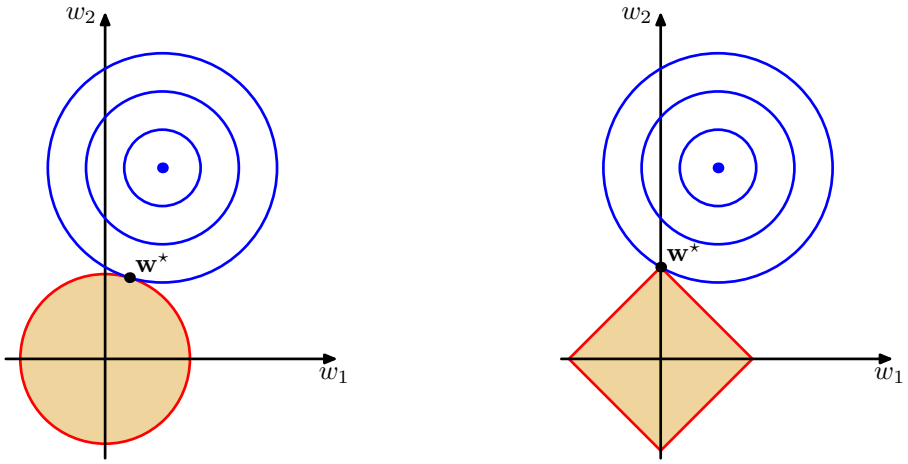
[Generalization of Quadratic Regularizer]

$$E(\mathbf{w}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\left\{t_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\right\}^2}_{E_D(\mathbf{w})} + \underbrace{\frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q}_{E_{\mathbf{w}}(\mathbf{w})}$$

- $q = 2$: quadratic regularizer, *ridge* regressor
- $q = 1$: known as *LASSO* (Least Absolute Shrinkage and Selection Operator)
  - If λ is sufficiently large, some of the coefficients w_j are driven to zero, leading to a *sparse* model in which the corresponding basis functions play no role



$|w_1|^{0.5} + |w_2|^{0.5} = \text{const}$; $|w_1| + |w_2| = \text{const}$; $w_1^2 + w_2^2 = \text{const}$; $w_1^4 + w_2^4 = \text{const}$



Plot of the contours of the unregularized error function (blue) along with the constraint region for $q = 2$ (left) and $q = 1$ (right), in which the optimum value for the parameter vector $\mathbf{w}$ is denoted by $\mathbf{w}^*$.

Minimizing

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

equivalent to minimizing

$$\frac{1}{2}\sum_{n=1}^{N}\left\{t_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\right\}^2 \text{ s.t. } \sum_{j=1}^{M}|w_j|^q \leq \eta$$

- can be related using Lagrange multipliers
- $\lambda \uparrow \Rightarrow$ an increasing number of parameters $\rightarrow 0$

Regularization

- Allows complex models to be trained on small data sets without severe overfitting
- Limits model complexity, *i.e.*, how many basis functions to use

- The problem of determining the optimal model complexity is shifted from one of finding the appropriate number of basis functions to one of determining suitable value of the regularization coefficient λ

Multiple Outputs

$$\mathbf{y}(\mathbf{x}, \mathbf{W}) = \mathbf{W}^{\top}\phi(\mathbf{x})$$

$\mathbf{y} \in \mathbb{R}^K, \mathbf{W} \in \mathbb{R}^{M \times K}, \phi(\mathbf{x}) \in \mathbb{R}^M$ with elements $\phi_j(\mathbf{x})$ and $\phi_0(\mathbf{x}) = 1$

- Conditional distribution of the target vector with an isotropic Gaussian form

$$p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) = \mathcal{N}\left(\mathbf{t}|\mathbf{W}^{\top}\phi(\mathbf{x}), \beta^{-1}\mathbf{I}\right)$$

- Log-likelihood

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

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

- For k-th target variable t_k

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

  - $\mathbf{t}_k$: N-dimensional column vector with components t_{nk} for $n = 1, \ldots, N$
- The solution decouples between the different target variables
- We need to compute $\Phi^\dagger$ once, shared by all of the vectors $\mathbf{w}_k$

Extension to general Gaussian noise distributions (i.e., having arbitrary covariance matrices)

- Still lead to a decoupling into K independent regression problems

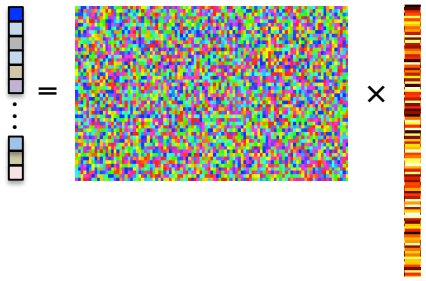
- Because the parameters $\mathbf{W}$ define only the mean of the Gaussian noise distribution, and we know that the maximum likelihood solution for the mean of a multivariate Gaussian is independent of the covariance.
  (refer to Section 2.3.4)

Sparse Linear Regression

- Least Absolute Shrinkage and Selection Operator (LASSO)

  [Tibshirani, 1996]

$$\min_{\mathbf{w}} \|\mathbf{y} - \Phi\mathbf{w}\|_2^2 + \lambda_1\|\mathbf{w}\|_1$$



$$(\mathbf{y} \in \mathbb{R}^N, \Phi \in \mathbb{R}^{N\times M}, \mathbf{w} \in \mathbb{R}^D)$$

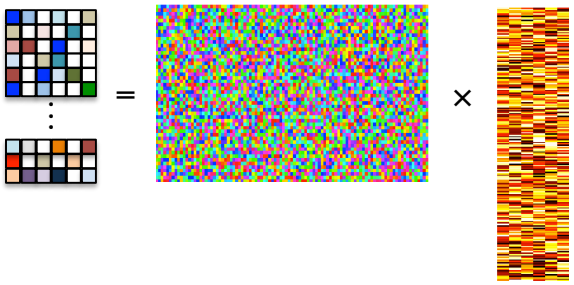
- Elastic Net [Zou and Hastie, 2005]

$$\min_{\mathbf{w}} \|\mathbf{y} - \Phi\mathbf{w}\|_2^2 + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{w}\|_2$$

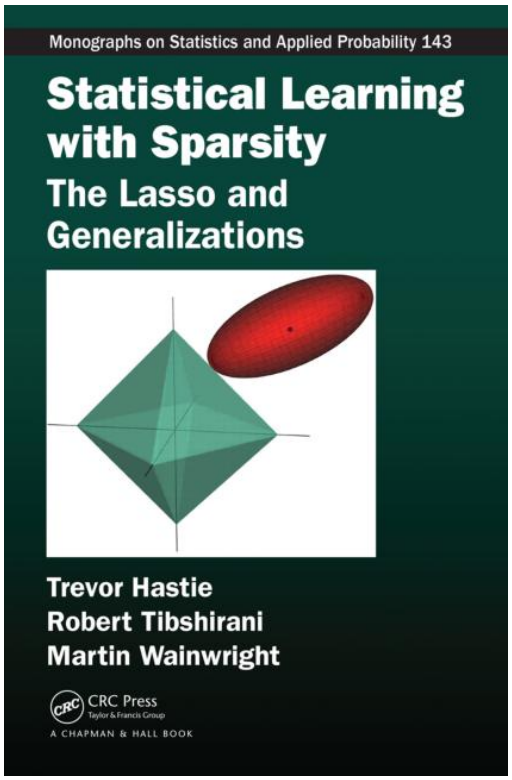
- Group sparsity (multi-task learning) [Yuan *et al.*, 2006]

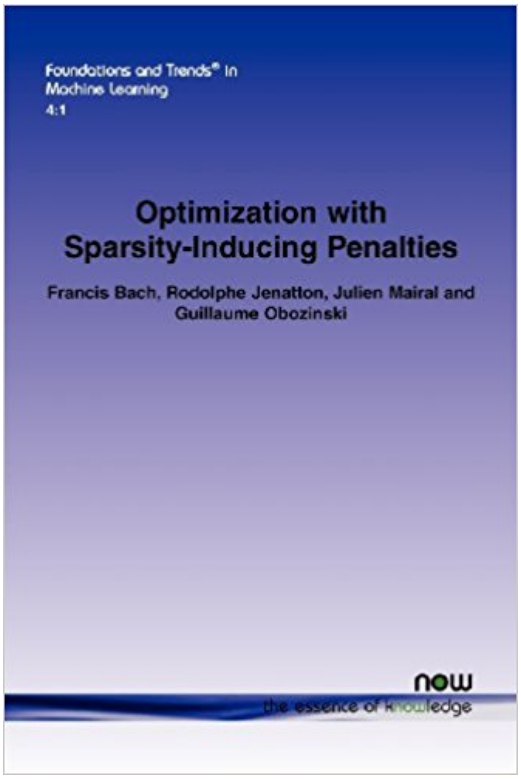
$$\min_{\mathbf{W}} \|\mathbf{Y} - \Phi\mathbf{W}\|_2^2 + \lambda\|\mathbf{W}\|_{2,1}$$

$$(\mathbf{Y} = \left[\mathbf{Y}^i\right]_{i=1}^N \in \mathbb{R}^{N\times K}, \Phi \in \mathbb{R}^{N\times M}, \mathbf{W} \in \mathbb{R}^{M\times K})$$



$$\|\mathbf{W}\|_{2,1} = \sum_{i=1}^M \left\|\mathbf{W}^i\right\|_2$$





Bayesian Linear Regression

Bayesian Linear Regression

Shortcomings of MLE

- Leaves the issue of deciding the appropriate model complexity
- How many basis functions ($M = ?$): $y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x})$
  - Controlled according to the size of the data set

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

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

- Overfitting problem
  - Regularization can somehow control it, though.
- Cross-validation: computationally expensive and wasteful of data

Bayesian treatment of linear regression

- Avoids the overfitting problem of maximum likelihood
- Leads to automatic methods of determining model complexity using the training data alone

Parameter Distribution

- Begin by introducing a prior probability distribution $p(\mathbf{w})$ and treating the noise precision parameter β as a known constant
- Focusing on the case of a single target variable t, for simplicity
- Likelihood function $p(\mathbf{t}|\mathbf{w})$: exponential of a quadratic function of $\mathbf{w}$

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

- (Conjugate prior) Gaussian distribution

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

- Due to the conjugate prior, the resulting posterior will also be Gaussian

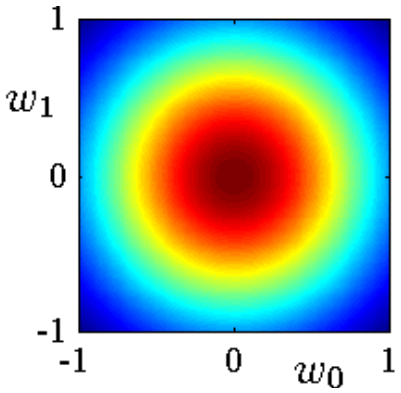
$$\begin{aligned} p(\mathbf{w}|\mathbf{t}) &= \frac{p(\mathbf{t}|\mathbf{w})\, p(\mathbf{w})}{p(\mathbf{t})} \\ &= \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \end{aligned}$$

$$\text{where} \begin{cases} \mathbf{m}_N = \mathbf{S}_N\left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\Phi^\top \mathbf{t}\right) \\ \mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\Phi^\top\Phi \end{cases}$$

- Posterior Gaussian distribution: the mode coincides with its mean

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

- Consider an infinitely broad prior $\mathbf{S}_0 = \alpha^{-1}\mathbf{I}$



when $\alpha \to 0$, $\mathbf{m}_N$ reduces to $\mathbf{w}_{ML} = \left(\Phi^\top \Phi\right)^{-1} \Phi^\top \mathbf{t}$

- If $N = 0$, *i.e.*, no training data available, the posterior reverts to the prior.
- Sequential learning: a posterior becomes the prior in the subsequent learning.

$$p\left(\mathbf{w}|\alpha\right) = \mathcal{N}\left(\mathbf{w}|0, \alpha^{-1}\mathbf{I}\right) \quad \Rightarrow \quad \begin{cases} \mathbf{m}_N = \beta \mathbf{S}_N \Phi^\top \mathbf{t} \\ \mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \Phi^\top \Phi \end{cases}$$

- Log of the posterior distribution

$$\ln\left(\mathbf{w}|\mathbf{t}\right) = -\frac{\beta}{2}\sum_{n=1}^{N}\left\{t_n - \mathbf{w}^\top \phi\left(\mathbf{x}_n\right)\right\}^2 \underbrace{-\frac{\alpha}{2}\mathbf{w}^\top \mathbf{w}}_{\text{regularization}} + \text{const.}$$

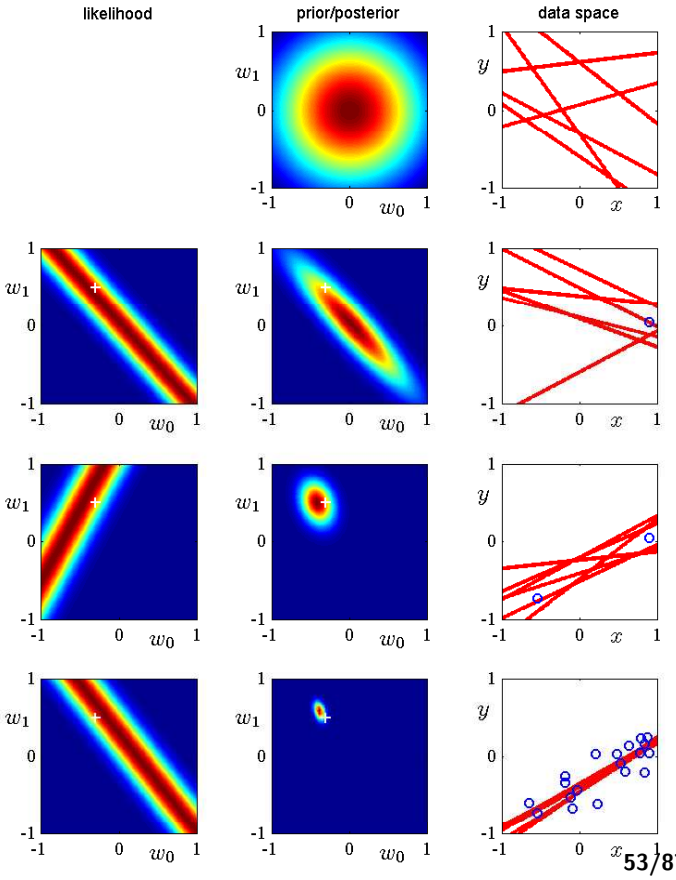
$$\max_{\mathbf{w}} \ln p\left(\mathbf{w}|\mathbf{t}\right) \equiv \min_{\mathbf{w}} E\left(\mathbf{w}\right) + \lambda\|\mathbf{w}\|_2^2$$

- $E\left(\mathbf{w}\right) = \frac{1}{2}\sum_{n=1}^{N}\left\{t_n - \mathbf{w}^\top \phi\left(\mathbf{x}_n\right)\right\}^2$ and $\lambda = \frac{\alpha}{\beta}$

Illustration of Bayesian learning in a linear basis function model

$y(x, \mathbf{w}) = w_0 + w_1 x$

- Generating synthetic data
  1. $f(x, \mathbf{a}) = a_0 + a_1 x$, $(a_0 = -0.3, a_1 = 0.5)$
  2. Choosing values of x_n from $U(x; -1, 1)$
  3. Obtaining target values by adding Gaussian noise: $\epsilon \sim \mathcal{N}(0, 0.2)$
- Set $\beta = (1/0.2)^2 = 25$, $\alpha = 2.0$
- Note that the posterior in the third row already relatively compact: two points are sufficient to define a line



Generalization of Gaussian Prior

$$p(\mathbf{w}|\alpha) = \left[\frac{q}{2}\left(\frac{\alpha}{2}\right)^{1/q}\frac{1}{\Gamma(1/q)}\right]^M \exp\left(-\frac{\alpha}{2}\sum_{j=1}^{M}|w_j|^q\right)$$

- $q = 2$: Gaussian distribution
  - Conjugate prior to Gaussian likelihood function
  - Finding the maximum of the posterior distribution over $\mathbf{w}$ corresponds to minimization of the regularized error function

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{t_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\right\}^2 + \frac{\lambda}{2}\sum_{j=0}^{M-1}|w_j|^q$$

  - Mode of the posterior distribution: equal to the mean

Predictive Distribution

- In practice, usually not interested in the value of $\mathbf{w}$ itself
- But rather in making predictions of t for new values of $\mathbf{x}$, requiring to evaluate the *predictive distribution*

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

  - Leaving out the conditioning variables $\mathbf{x}$ for notational simplicity

$$\begin{aligned} p(t|\mathbf{x}, \mathbf{w}, \beta) &= \mathcal{N}\left(t|y(\mathbf{x}; \mathbf{w}), \beta^{-1}\right) \\ p(\mathbf{w}|\mathbf{t}, \alpha, \beta) &= \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \end{aligned}$$

$$\text{where } \begin{cases} \mathbf{m}_N = \mathbf{S}_N\left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\Phi^\top \mathbf{t}\right) \\ \mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\Phi^\top\Phi \end{cases}$$

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

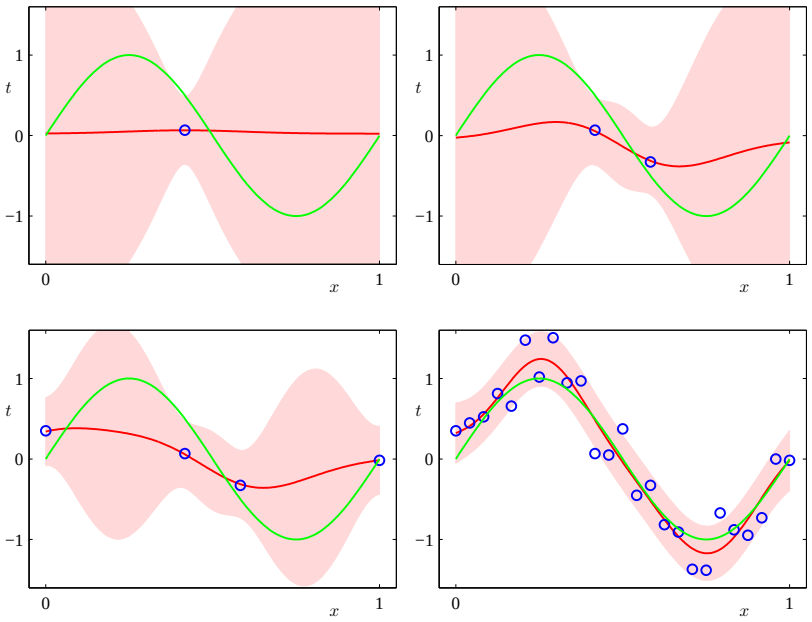
$$\text{where } \sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \phi(\mathbf{x})^\top \mathbf{S}_N \phi(\mathbf{x})$$

- Additive variances $\sigma_N^2(\mathbf{x})$ due to two independent Gaussians
  - $\frac{1}{\beta}$: noise on the data (noise process)
  - $\phi(\mathbf{x})^\top \mathbf{S}_N \phi(\mathbf{x})$: uncertainty associated with $\mathbf{w}$ (distribution of $\mathbf{w}$)
- As additional data points are observed, the posterior distribution becomes narrower

$$\begin{aligned} \sigma_{N+1}^2(\mathbf{x}) &\leq \sigma_N^2(\mathbf{x}) \\ \lim_{N\to\infty} \phi(\mathbf{x})^\top \mathbf{S}_N \phi(\mathbf{x}) &= 0 \end{aligned}$$

  - Variance arises solely from the additive noise governed by β

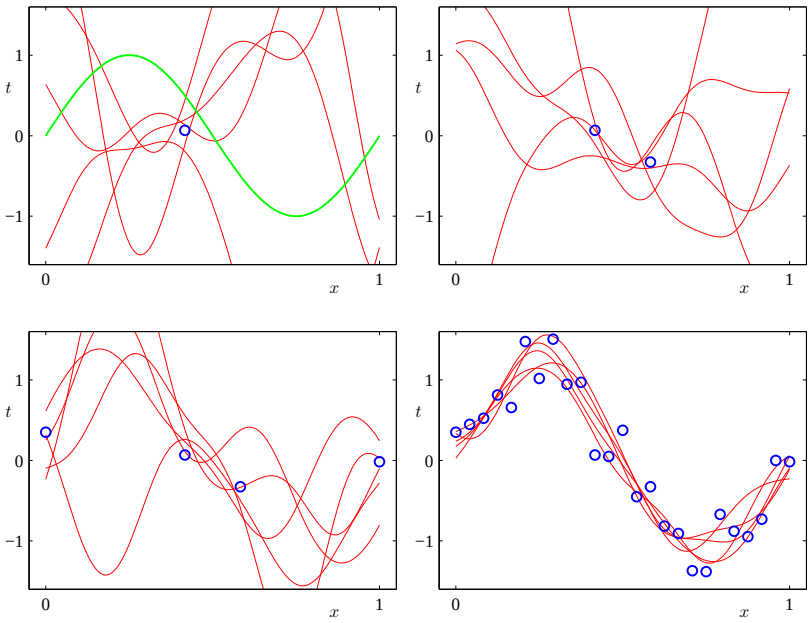
Point-wise predictive variance as a function of x



A model consisting of 9 Gaussian basis functions
Green: $\sin(2\pi x)$; Blue: data sets of N=1,2,4,25;
Red: mean of the corresponding Gaussian predictive distribution;
Shaded: one standard deviation either side of the mean

Predictive uncertainty depends on x and is smallest in the neighbourhood of the data points. (The level of uncertainty decreases as more data points are observed.)

- Insight into the covariance b/w predictions at different values of x
  - Draw samples from the posterior distribution over $\mathbf{w}$
  - Then plot the corresponding functions $y(x, \mathbf{w})$



These curves represent the distribution of the regression function.

Undesirable behaviour when using localized basis functions such as Gaussians

- In regions away from the basis function centers, the contribution from the term of $\phi(\mathbf{x})^{\top} \mathbf{S}_N \phi(\mathbf{x})$ will go to zero, leaving only the noise contribution β^{-1}.
- Thus, the model becomes very confident in its predictions when extrapolating outside the region occupied by the basis functions.
- Can be avoided by adopting an alternative Bayesian approach to regression known as a '**Gaussian Process**'.

If both $\mathbf{w}$ and β are treated as unknown

- Conjugate prior distribution $p(\mathbf{w}, \beta)$: Gaussian-gamma distribution

$$p(\mathbf{w}, \beta) = \mathcal{N}\left(\mathbf{w}|\mathbf{m}_0, \beta^{-1}\mathbf{S}_0\right) \text{Gam}\left(\beta|a_0, b_0\right)$$

- Predictive distribution: Student's t-distribution

$$p(t|\mathbf{x}, \mathbf{t}) = \text{St}\left(t|\mu, \lambda, \nu\right)$$

  - $\mu = \Phi^{\top}\mathbf{m}_N$, $\lambda = \frac{a_N}{b_N}\left\{1 + \Phi^{\top}\left(\mathbf{S}_0 + \Phi^{\top}\Phi\right)^{-1}\Phi\right\}^{-1}$, $\nu = 2\left(a_0 + \frac{N}{2}\right)$

Equivalent Kernel

Interpretation of the posterior mean solution

$$p(\mathbf{w}|\alpha) = \mathcal{N}\left(\mathbf{w}|0, \alpha^{-1}\mathbf{I}\right) \quad \Rightarrow \quad \begin{cases} \mathbf{m}_N = \beta \mathbf{S}_N \Phi^\top \mathbf{t} \\ \mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \Phi^\top \Phi \end{cases}$$

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

- Mean of the predictive distribution at a point $\mathbf{x}$ is given by a linear combination of the training set target variables t_n

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

$$k(\mathbf{x}, \mathbf{x}') = \beta \phi(\mathbf{x})^\top \mathbf{S}_N \phi(\mathbf{x}')$$

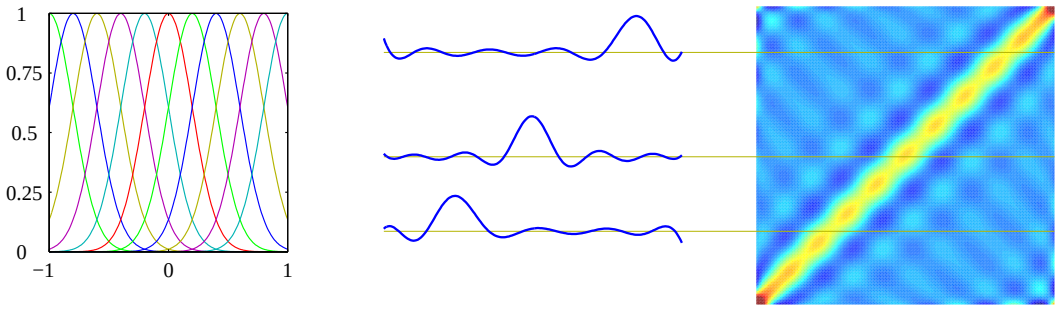
called equivalent kernel or smoother matrix

- *Linear smoothers*

  Regression functions that make predictions by taking linear combinations of the training set target values

- The equivalent kernel depends on the input values $\mathbf{x}_n$ from the data set because these appear in the definition of $\mathbf{S}_N$.

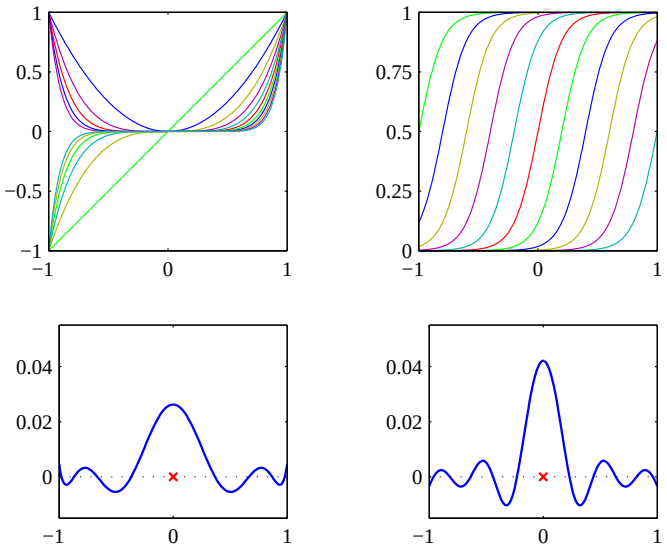
Equivalent kernel $k(x, x')$ for the Gaussian basis functions



- Data set used to generate this kernel comprised 200 values of x equally spaced over the interval $(-1, 1)$
- The mean of the predictive distribution at x, given by $y(x, \mathbf{m}_N)$, is obtained by forming a weighted combination of the target values.
  - data points close to x are given higher weight than points further removed from x

Role of an equivalent kernel: defining the weights by which the training set target values are combined in predicting a target value at a new value of $\mathbf{x}$.

- *Localization property*
  - Weight local evidence more strongly than distant evidence
  - Holds not only for the localized Gaussian basis functions but also for the nonlocal polynomial (left) and sigmoidal (right) basis functions



- Further Insight into the role of the equivalent kernel

$$\begin{aligned} \text{Cov}\left[y(\mathbf{x}), y(\mathbf{x}')\right] &= \text{Cov}\left[\phi(\mathbf{x})^\top \mathbf{w}, \mathbf{w}^\top \phi(\mathbf{x}')\right] \\ &= \phi(\mathbf{x})^\top \mathbf{S}_N \phi(\mathbf{x}') = \beta^{-1} k(\mathbf{x}, \mathbf{x}') \end{aligned}$$

  - Making use of $p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$ and $k(\mathbf{x}, \mathbf{x}') = \beta\phi(\mathbf{x})^\top \mathbf{S}_N \phi(\mathbf{x}')$
  - The predictive mean at nearby points will be highly correlated, whereas for more distant pairs of points the correlation will be smaller.

Alternative approach to regression by the formulation of linear regression in terms of a kernel function

- Define a localized kernel directly and use this to make predictions for new input vectors $\mathbf{x}$, given the observed training set
  - cf) introducing a set of basis functions (implicitly determines an equivalent kernel)
- Leads to a practical framework for regression (and classification)

Gaussian Processes

- The effective kernel defines the weights by which the training set target values are combined in order to make a prediction at a new value of $\mathbf{x}$.
- For all values of $\mathbf{x}$, sum of the equivalent kernel values, i.e., weights, equal to one.

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

- The equivalent kernel, $k(\mathbf{x}, \mathbf{x}') = \beta\phi(\mathbf{x})^\top \mathbf{S}_N \phi(\mathbf{x}')$, can be expressed in the form of an inner product.
  - Shared by kernel functions in general

$$k(\mathbf{x}, \mathbf{z}) = \psi(\mathbf{x})^\top \psi(\mathbf{z})$$

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

Bayesian Model Comparison

Bayesian Perspective

- Avoiding the overfitting associated with maximum likelihood
  - By marginalizing over the model parameters
- Models can be directly compared on the training data
  - No need for a validation set
  - Allowing all available data to be used for training
  - Avoiding multiple training runs for each model associated with cross-validation
  - Allowing multiple complexity parameters to be determined simultaneously as part of the training process

Bayesian View of Model Comparison

- Use of probabilities to represent uncertainty in the choice of model
- Comparing a set of L models $\{\mathcal{M}_i\}$
- Given a training set $\mathcal{D}$, wish to evaluate the posterior distribution

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

  - $p(\mathcal{M}_i)$: *prior*, preference for different models
  - $p(\mathcal{D}|\mathcal{M}_i)$: *model evidence*, preference shown by the data for different models
    - also called *marginal likelihood*: a likelihood function over the space of models, in which the parameters have been marginalized out
- Bayes factor [Kass and Faftery, 1995]
  - Ratio of model evidences for two models $\frac{p(\mathcal{D}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_j)}$

- Predictive distribution with the known posterior distribution

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

  - Mixture distribution: averaging the predictive distributions, weighted by $p(\mathcal{M}_i|\mathcal{D})$
  - Model selection: approximation to model averaging with the single most probable model alone to make predictions

Model Evidence

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

($\mathbf{w}$: a set of parameters for a model)

- From a sampling perspective,
  - marginal likelihood: probability of generating the data set $\mathcal{D}$ from a model $\mathcal{M}_i$ whose parameters are sampled from the prior
- Normalizing term (or denominator) in Bayes' theorem

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

Making a simple approximation to the integral over parameters

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

- Consider a model $\mathcal{M}_i$ with a single parameter w
- Posterior distribution $p(w|\mathcal{D}) \propto p(\mathcal{D}|w)\, p(w)$
  - Omit the dependence on the model $\mathcal{M}_i$ to keep notation uncluttered

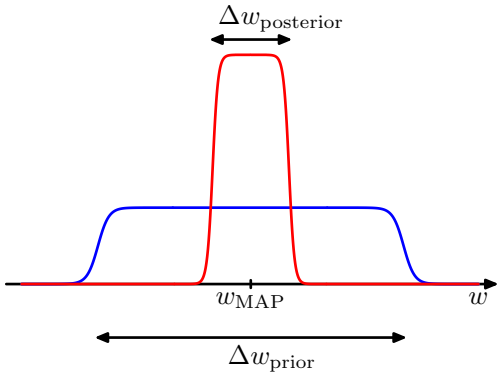
- If the posterior probability is sharply peaked around w_{MAP}
  - Integral $\simeq$ (peak value, *i.e.*, maximum probability) $\times$ (width of peak)
    - $\Delta w_{\text{posterior}}$: width of the peak
    - $p(w) = 1/\Delta w_{\text{prior}}$: flat prior

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

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

- $\ln p(\mathcal{D}|w_{\text{MAP}})$
  - Fit to the data given most probable parameter values
  - For a flat prior, corresponds to the log likelihood

- $\ln\left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}\right)$
  - Penalizes the model according to its complexity since $\Delta w_{\text{posterior}} < \Delta w_{\text{prior}}$ and terms is negative
  - If parameters are finely tuned to the data, this term is large.



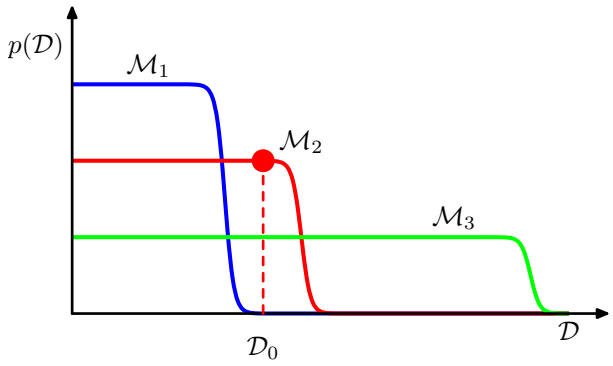
- For a model having a set of M parameters, assuming all parameters have the same ratio $\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}$

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

- The size of the complexity penalty increases linearly with the number M of adaptive parameters in the model.
- $\ln p(\mathcal{D}|\mathbf{w}_{\text{MAP}})$ will decrease with model complexity since it better fits the data (overfitting)
- $M \ln \left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}} \right)$ will increase due to dependence on M
- Trade-off between terms determines the optimal model complexity.

Insight into how the marginal likelihood can favour models of intermediate complexity

- Horizon-axis: one-dimensional representation of the space of possible data sets
- Complexity: $\mathcal{M}_1 < \mathcal{M}_2 < \mathcal{M}_3$
  - A simple model (*e.g.*, 1st-order polynomial): little variability in generated data sets
  - A complex model (*e.g.*, 9th-order polynomial): high variation of different data sets



Distribution of data sets for three models of different complexity

- For a specific dataset $\mathcal{D}_0$
  - Model of intermediate complexity, $\mathcal{M}_2$, has high evidence
  - Simple model $\mathcal{M}_1$ fails to fit the data well
  - Complex mode $\mathcal{M}_3$ assigns relatively small probability for any data set

- Implicit in the Bayesian model comparison framework is the assumption that the true distribution from which the data are generated is contained within the set of the true distribution.
- When $\mathcal{M}_1$ is the true model, averaging the Bayes factor between $\mathcal{M}_1$ and $\mathcal{M}_2$ over the distribution of data sets

$$\int p(\mathcal{D}|\mathcal{M}_1) \ln \frac{p(\mathcal{D}|\mathcal{M}_1)}{p(\mathcal{D}|\mathcal{M}_2)} d\mathcal{D} \geq 0 \quad \text{(Kullback-Leibler divergence)}$$

  - Equal to zero iff $p(\mathcal{D}|\mathcal{M}_1) = p(\mathcal{D}|\mathcal{M}_2)$, *i.e.*, $\mathcal{M}_1 = \mathcal{M}_2$
- Thus, Bayesian model comparison favours the correct model.