Notes 3-Linear Models for Regeression

Junda Du

2019-8-28

1 Linear Basis Function Models

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\boldsymbol{x}) \overset{\boldsymbol{\phi}_0(\boldsymbol{x})=1}{=\!=\!=} \sum_{j=0}^{M-1} w_j \boldsymbol{\phi}_j(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}),$$

where $\boldsymbol{w} = (w_0, \ldots, w_{M-1})^T, \boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^T$, $y(\boldsymbol{x}, \boldsymbol{w})$ is called linear models(linear for $\boldsymbol{w}$), but $\boldsymbol{\phi}(\boldsymbol{x})$ is the nolinear function, playing a role in preprocessing and feature extraction.
The basis function $\phi_j(x)$ have many choice, for example

- $\phi_j(x) = x^j$, which is used for polynomial regression in Chapter 1. Its limitation is that they are global functions of the input variables, so that changes in one region of input space affect all other regions.??
- 'Gaussian' basis functions $\phi_j(x) = exp\Big\{-\dfrac{(x-\mu_j)^2}{2s^2}\Big\}$. μ_j govern the locations of the basis functions in input space, s governs their spatial scale, and the normalization coefficient is unimportant because of w_j.
- Sigmoidal basis function $\phi_j(x) = \sigma\Big(\dfrac{x-\mu_j}{s}\Big)$, where $\sigma(a) = \dfrac{1}{1+exp(-a)}$ is the logistic sigmoid function. Equivalently, we can use 'tanh' function $tanh(a) = 2\sigma(a) - 1$
- Fourier basis: each function represents a specific frequency and has infinite spatial extent.It's interesting to consider basis functions that are localized in both space and frequency, leading to a class fo functions known as *wavelets*.

1.1 Maximum likelihood and least squares

Maximum likelihood steps(least squares appears in step 3):

1. Target varibale probablity distribution given parameters:

$$p(t|\boldsymbol{x}, \boldsymbol{w}, \beta) = N(t|y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}), y(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$$

2. Joint probablity given a series of observations $\boldsymbol{X} = x_1, \ldots, x_N$ with corresponding target values $\boldsymbol{t} = t_1, \ldots, t_N$:

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

3. Take the logarithm of the likelihood function:

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

where $E_D(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\}^2$, which corresponds to least squares.

4. Solving for parameters:

   - Derivate log likelihood function with respect to $\boldsymbol{w}$:

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

     Solving for $\boldsymbol{w}$ we obtain: $\boldsymbol{w}_{ML} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{t}$, which is known as the *normal equations* for the least squares problem.
     $\boldsymbol{\Phi}$ is and $N \times M$ matrix, called *design matrix*, $\Phi_{nj} = \phi_j(\boldsymbol{x}_n)$. And the quantity $(\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T$ is known as the *Moore-Penrose pseudo-inverse* of the matrix $\boldsymbol{\Phi}$.
   - Derivate $E_D(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - w_0 - \sum_{j=1}^{M-1} w_j\phi_j(\boldsymbol{x}_n)\}^2$ with respect to w_0 equal to zero, and solving for w_0, we obtain

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

   - Derivate log likelihood function with respect to β, giving

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

1.2 Geometry of least squares

1.3 Sequential learning

1.4 Regularized least squares

Add a regularization term to an error function $E_W(w) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$ in order to control over-fitting, so

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

Accordingly, solving for $\boldsymbol{w}$ as before, we obtain $\boldsymbol{w} = (\lambda \boldsymbol{I} + \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{t}$.
A more general regularization term is $E_W(\boldsymbol{w}) = \frac{1}{2} \sum_{j=1}^{M} |w_j|^q$, so

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

And minimizing it is equivalent to minimizing the unregularized sum-of-squares error subject to the constraint $\sum_{j=1}^{M} |w_j|^q \leq \eta$, which can be found by Lagrange multipiers.
The case of $q = 1$ leads to a sparse model, i.e. some of the coefficients w_j are driven to zero. The origin can be seen from the following picture.

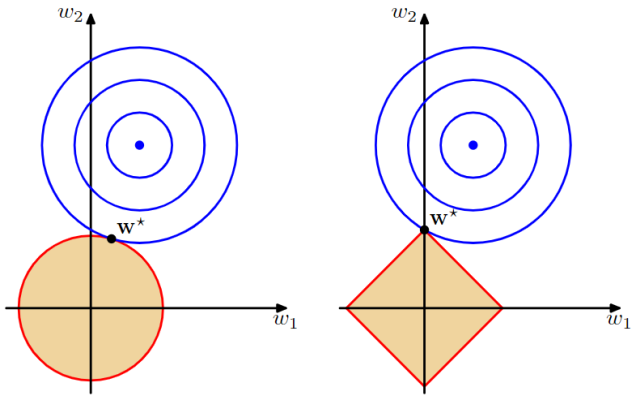

Figure 3.4 Plot of the contours of the unregularized error function (blue) along with the constraint region (3.30) for the quadratic regularizer $q = 2$ on the left and the lasso regularizer $q = 1$ on the right, in which the optimum value for the parameter vector $\mathbf{w}$ is denoted by $\mathbf{w}^\star$. The lasso gives a sparse solution in which $w_1^\star = 0$.

1.5 Multiple outputs

If target vector $\boldsymbol{t}$ is a K-dimensional column vector, the approach is to use the same set of basis function to model all of the components of the target vector so that $\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{W}^T \boldsymbol{\phi}(\boldsymbol{x})$, then the conditional distribution of the target vector is $p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{W}, \beta) = N(\boldsymbol{t}|\boldsymbol{W}^T \phi(\boldsymbol{x}), \beta^{-1} \boldsymbol{I})$ assuming that is an isotropic Gaussian.
The log likelihood function is

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

Solving for $\boldsymbol{W}$, giving $\boldsymbol{W}_{ML} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{T}$, examine this result for each target variable $\boldsymbol{t}_k$, we have $\boldsymbol{w}_k = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{t}_k$, which leads to a decoupling into K independent regression problems when sharing the pseudo-inverse matrix.

2 The Bias-Variance Decomposition

It's unreasonable to minimize the regularized error function with respect to λ, a good approach is marginalizing over parameters in a Baysian setting. Before doing so, it's instructive to consider a frequentist viewpoint of the model complexity issue, known as the *bias-variance*.
The expected squared loss can be written in the form

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

where $y(\boldsymbol{x})$ is which we seek a solution for to make first term a minimum, $h(\boldsymbol{x})$ is the optimal prediction given conditional distribution and the loss function, such as $h(\boldsymbol{x}) = E[t|\boldsymbol{x}] = \int tp(t|\boldsymbol{x})dt$ given $p(t|\boldsymbol{x})$ and the squared loss function.

Supposed we have a large number of data sets each of size N, for any given data set D we can run learning algorithm and obtain a prediction function $y(\boldsymbol{x}; D)$. For a given data set D, the second term is about noise, independent of $y(\boldsymbol{x})$. Consider the integrand of the first term $\{y(\boldsymbol{x}; D) - h(\boldsymbol{x})\}^2$, we introduce the quantity $E_D[y(\boldsymbol{x}; D)]$ because this term is dependent on the particular data set D.We obtain
$\{y(\boldsymbol{x}; D) - E_D[y(\boldsymbol{x}; D)] + E_D[y(\boldsymbol{x}; D)] - h(\boldsymbol{x})\}^2$
$= \{y(\boldsymbol{x}; D) - E_D[y(\boldsymbol{x}; D)]\}^2 + \{E_D[y(\boldsymbol{x}; D)] - h(\boldsymbol{x})\}^2$
$+ 2\{y(\boldsymbol{x}; D) - E_D[y(\boldsymbol{x}; D)]\}\{E_D[y(\boldsymbol{x}; D)] - h(\boldsymbol{x})\}$,
then $E_D[\{y(\boldsymbol{x}; D) - h(\boldsymbol{x})\}^2] = \underbrace{\{E_D[y(\boldsymbol{x}; D)] - h(\boldsymbol{x})\}^2}_{bias^2} + \underbrace{E_D[\{y(\boldsymbol{x}; D) - E_D[y(\boldsymbol{x}; D)]\}^2]}_{variance}$

$E[L] = bias^2 + variance + noise$, where $bias^2 = \int \{E_D[y(\boldsymbol{x}; D)] - h(\boldsymbol{x})\}^2 p(\boldsymbol{x}) d\boldsymbol{x}$, $variance = \int E_D[\{y(\boldsymbol{x}; D) - E_D[y(\boldsymbol{x}; D)]\}^2] p(\boldsymbol{x}) d\boldsymbol{x}$, $noise = \int \{h(\boldsymbol{x}) - t\}^2 p(\boldsymbol{x}, t) d\boldsymbol{x} dt$

As we can see, there is a trade-off between bias and variance with flexible models(small λ) having high bias and low variance, and relatively rigid models(large λ) having high bias and low variance.

3 Bayesian Linear Regression

We turn to a Baysesian treatment of linear regression, which will avoid the overfitting problem of maximum likelihood, and which will also lead to automatic methods of determining model complexity using the training data alone.

3.1 Parameter distribution

We want to obtain $p(\boldsymbol{w}|\boldsymbol{t})$. Noting the likelihood function $p(\boldsymbol{t}|\boldsymbol{w}) = \prod_{n=1}^{N} N(t_n|\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1})$ in 3.1.1, the corresponding conjugate prior is therefore given by the distribution of the form $p(\boldsymbol{w}) = N(\boldsymbol{w}|\boldsymbol{m}_0, \boldsymbol{S}_0)$ having the mean $\boldsymbol{m}_0$ and covariance $\boldsymbol{S}_0$. Making use of the result in 2.3.3, we can obtain

$$p(\boldsymbol{w}|\boldsymbol{t}) = N(\boldsymbol{w}|\boldsymbol{m}_N, \boldsymbol{S}_N), \boldsymbol{m}_N = \boldsymbol{S}_N(\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 + \beta \boldsymbol{\Phi}^T \boldsymbol{t}), \boldsymbol{S}_N^{-1} = \boldsymbol{S}_0^{-1} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

For the remainder of this chapter, we shall consider a zero-mean isotropic Gaussian $p(\boldsymbol{w}|\alpha) = N(\boldsymbol{w}|\boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$ to simplify the treament.And corresponding posterior distribution over $\boldsymbol{w}$ is given by

$$p(\boldsymbol{w}|\boldsymbol{t}) = N(\boldsymbol{w}|\boldsymbol{m}_N, \boldsymbol{S}_N), \boldsymbol{m}_N = \beta \boldsymbol{S}_N \boldsymbol{\Phi}^T \boldsymbol{t}, \boldsymbol{S}_N^{-1} = \alpha \boldsymbol{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}.$$

The log posterior distribution is given by the sum of the log likelihood and log prior,

$$ln\, p(\boldsymbol{w}|\boldsymbol{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\}^2 - \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} + const.$$

Of course, it applys to sequential learning, i.e. it's easy to update the result when a new observation arrives.

There is a general Gaussian form of prior over the parameter $\boldsymbol{w}$

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

In our discussion of $q = 2$, the mode of posterior distribution is equal to mean.

3.2 Predictive distribution

We want to obtain $p(t|\boldsymbol{t})$. Given $p(t|\boldsymbol{w}, \beta)$ and $p(\boldsymbol{w}|\boldsymbol{t}, \alpha, \beta)$, the *predictive distribution* is defined by

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

Making use of the result in 2.3.3, it takes the form

$$p(t|\boldsymbol{x}, \boldsymbol{t}, \alpha, \beta) = N(t|\boldsymbol{m}_N^T\boldsymbol{\phi}(\boldsymbol{x}), \sigma_N^2(\boldsymbol{x})), \sigma_N^2(\boldsymbol{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}),$$

where $\sigma_N^2(\boldsymbol{x})$ is associated with data noise and parameters $\boldsymbol{w}$ and they are independent so are additive. Also note that the level of uncertainty decreases as more data points are observed and $\sigma_{N+1}^2(\boldsymbol{x}) \leq \sigma_N^2(\boldsymbol{x})$. Problems:

- If we used localized basis functions such as Gaussians, then in region away from the basis function centre, the contribution from the second term in the predicive variance will go to zero. Thus,the model becomes very confident in its predictions when extrapolating outside theregion occupied by the basis functions, which is generally an undesirable behaviour.This problem can be avoided by adopting an alternative Bayesian approach to regression known as a Gaussian process.

- If both $\boldsymbol{w}$ and β are treated as unknown, then we can introduce a conjugate prior distribution $p(\boldsymbol{w}, \beta)$, which is given by Gaussian-gamma distribution discussed in 2.3.6.

3.3 Equivalent kernel

We want to introduce the equivalent kernel method by linear prediction. It's intuitive, jianjiede?? and have some properties.

We using $p(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x})$ to predict the target given the input, and we can estimate $\boldsymbol{w}$ using $\boldsymbol{m}_N$ in 3.3.1. Substitute it into the predictive function we can get

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

$$\xrightarrow[equivalent\ kernel]{k(\boldsymbol{x},\boldsymbol{x}')=\beta\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}')} \sum_{n=1}^{N} k(\boldsymbol{x}, \boldsymbol{x}_n)t_n, [Property\,1 : \sum_{n=1}^{N} k(\boldsymbol{x}, \boldsymbol{x}_n) = 1]$$

$$\xrightarrow[inner\ product\ transform]{\boldsymbol{\psi}(\boldsymbol{x})=\beta^{1/2}\boldsymbol{S}_N^{1/2}\boldsymbol{\phi}(\boldsymbol{x})} \boldsymbol{\psi}(\boldsymbol{x})^T\boldsymbol{\psi}(\boldsymbol{x}), [Property\,2 : k(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{\psi}(\boldsymbol{x})^T\boldsymbol{\psi}(\boldsymbol{z})]$$

We can obtain some insight from equivalent kernel:

- Regression functions, such as this, which makes predictions by taking combinations of the training set target values are known as *linear smoothers*. And the equivalent kernel is called smoother matrix.
- *Localization property??*: the weight of target is dependent on the defined distance[i.e. $\boldsymbol{\psi}(\boldsymbol{x})$] betweet the data points and x with closer data points has higher weight.
- The localization property holds not only for the localized Gaussian basis functions but also for the nonlocal polynomial and sigmoidal basis functions.??
- Instead of introducing a set of basis functions, which implicitly determines an equivalent kernel, we can instead define a localized kernel directly and use this to make predictions for new input given the observed training set.

4 Bayesian Model Comparison

There are many models to predict, we can choose the most accurate, known as *model selection*; we can also make use of the sum and product rules by $p(t|\boldsymbol{x}, D) = \sum_{i=1}^{L} p(t|\boldsymbol{x}, M_i, D)p(M_i|D)$, which is an example of a *mixture distribution* weighted by $p(M_i|D)$.

We need to obtain $p(M_i|D)$ by $p(M_i|D) \propto p(M_i)p(D|M_i)$, where $p(D|M_i)$

is important and known as *model evidence*. Then we try to obtain $p(D|M_i)$.

$$p(D|M_i) = \int p(D|\boldsymbol{w}, M_i)p(\boldsymbol{w}|M_i)d\boldsymbol{w}$$

$$\Big[Ignore\ M_i\Big]p(D) = \int p(D|w)p(w)dw$$

$$\xrightarrow{\substack{p(w)\ is\ peaked\ around\ w_{MAP}\ with\ width\ \Delta w_{prior} \\ p(D|w)\ is\ peaked\ around\ w_{MAP}\ with\ width\ \Delta w_{posterior}}} p(D|w_{MAP})\frac{\Delta w_{posterior}}{\Delta w_{prior}}$$

$$\Big[Taking\ logs\Big]ln\, p(D) = ln\, p(D|w_{MAP}) + ln\Big(\frac{\Delta w_{posterior}}{\Delta w_{prior}}\Big)$$

$$\Big[The\ model\ has\ M\ parameters\Big]ln\, p(D) = ln\, p(D|\boldsymbol{w}_{MAP}) + M\, ln\Big(\frac{\Delta w_{posterior}}{\Delta w_{prior}}\Big)$$

Here, we can see that the maximum evidence is given by the trade-off between these two competing terms. This is why Bayesian framework avoids the over-fitting problems.

5 The Evidence Approximation

In a fully Bayesian treatment, we don't select any model, but consider all models, i.e. integrate over all parameters. So

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

We have obtained $p(t|\boldsymbol{w}, \beta), p(\boldsymbol{w}|\boldsymbol{t}, \alpha, \beta)$, so we need to obtain $p(\alpha, \beta|\boldsymbol{t})$ and $p(\alpha, \beta|\boldsymbol{t}) \propto p(\boldsymbol{t}|\alpha, \beta)p(\alpha, \beta)$ from Bayes' theorem. The important term is $p(\boldsymbol{t}|\alpha, \beta)$ because chances are that the prior is relatively flat or the training data set is large.

5.1 Evaluation of the evidence function

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

$$\xrightarrow{3.1.1,\ 3.3.1,\ 2.3.3} \Big(\frac{\beta}{2\pi}\Big)^{N/2}\Big(\frac{\alpha}{2\pi}\Big)^{M/2}\int exp\{-E(\boldsymbol{w})\}d\boldsymbol{w}$$

where

$$\begin{aligned} E(\boldsymbol{w}) =& \beta E_D(\boldsymbol{w}) + \alpha E_W(\boldsymbol{w}) \\ =& \frac{\beta}{2}\|\boldsymbol{t} - \boldsymbol{\Phi w}\|^2 + \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} \\ \xrightarrow{complete\ square}& E(\boldsymbol{m}_N) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{m}_N)^T\boldsymbol{A}(\boldsymbol{w} - \boldsymbol{m}_N), \\ & \boldsymbol{A} = \alpha\boldsymbol{I} + \beta\boldsymbol{\Phi}^T\Phi = \nabla\nabla E(\boldsymbol{w}) = \boldsymbol{S}_N^{-1}, \\ & E(\boldsymbol{m}_N) = \frac{\beta}{2}\|\boldsymbol{t} - \boldsymbol{\Phi m}_N\|^2 + \frac{\alpha}{2}\boldsymbol{m}_N^T\boldsymbol{m}_N, \boldsymbol{m}_N = \beta\boldsymbol{A}^{-1}\boldsymbol{\Phi}^T\boldsymbol{t} \end{aligned}$$

So

$$\begin{aligned}\int exp\{-E(\boldsymbol{w})\}d\boldsymbol{w} =& exp\{-E(\boldsymbol{m}_N)\}\int exp\Big\{-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{m}_N)^T\boldsymbol{A}(\boldsymbol{w}-\boldsymbol{m}_N)\Big\}\\ =& exp\{-E(\boldsymbol{m}_N)\}(2\pi)^{M/2}|\boldsymbol{A}|^{-1/2}\end{aligned}$$

Finally,

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

For the example in chapter 1, fix α, change M, the evidence favours the model with $M = 3$.

5.2 Maximizing the evidence function

First consider the maximization of $p(\boldsymbol{t}|\alpha,\beta)$ with respect to α. Obtaining $\frac{d}{d\,\alpha}ln\, p(\boldsymbol{t}|\alpha,\beta)$, needs $\frac{d}{d\,\alpha}ln\,|\boldsymbol{A}|$, then needs to define the following eigenvector equation $(\beta\boldsymbol{\Phi}^T\boldsymbol{\Phi})\boldsymbol{u}_i = \lambda_i\boldsymbol{u}_i$, so $\boldsymbol{A}$ has eigenvalues $\alpha + \lambda_i$, so

$$\frac{d}{d\,\alpha}ln\,|\boldsymbol{A}| = \frac{d}{d\,\alpha}ln\prod_i(\lambda_i+\alpha) = \frac{d}{d\,\alpha}\sum_i ln(\lambda_i+\alpha) = \sum_i\frac{1}{\lambda_i+\alpha}$$

So

$$\begin{aligned}&\frac{d}{d\,\alpha}ln\, p(\boldsymbol{t}|\alpha,\beta) = \frac{M}{2\alpha} - \frac{1}{2}\boldsymbol{m}_N^T\boldsymbol{m}_N - \frac{1}{2}\sum_i\frac{1}{\lambda_i+\alpha} = 0\\ \Leftrightarrow&\alpha\boldsymbol{m}_N^T\boldsymbol{m}_N = M - \alpha\sum_i\frac{1}{\lambda_i+\alpha} = \gamma,\\ \Leftrightarrow&\alpha = \frac{\gamma}{\boldsymbol{m}_N^T\boldsymbol{m}_N}, \gamma = M - \alpha\sum_i\frac{1}{\lambda_i+\alpha} = \sum_i\frac{\lambda_i}{\alpha+\lambda_i}\end{aligned}$$

Similarly, we obtain

$$\frac{1}{\beta} = \frac{1}{N-\gamma}\sum_{n=1}^{N}\{t_n - \boldsymbol{m}_N^T\boldsymbol{\phi}(\boldsymbol{x}_n)\}^2$$

5.3 Effective number of parameters

γ represent the number of the effective number because for $\lambda \gg \alpha$, $\frac{\lambda_i}{\alpha+\lambda_i}$ is 1, which means λ_i is effective, for $\lambda \ll \alpha$, $\frac{\lambda_i}{\alpha+\lambda_i}$ is 0, which means λ_i is non-effective.

For the case of $N \gg M$, $\gamma = M$ because $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ involves an implicit sum over data points and so the eigenvalues λ_i increase with the size of the data set. So the re-estimation equations for α, β becomes

$$\alpha = \frac{M}{2E_W(\boldsymbol{m}_N)}, \beta = \frac{N}{2E_D(\boldsymbol{m}_N)}$$

6 Limitations of Fixed Basis Functions

The basis functions $\phi_j(\boldsymbol{x})$ are fixed before the training data set is observed and is a manifestation of the curse of dimensionality. As a consequence, the number of basis functions needs to grow repidly.

Fortunately, there are two properties of real data sets that we can exploit to help alleviate this problem. First, the real dimension of $\boldsymbol{x}$ is smaller than input space because of the strong correlations between the input variables. The second property is that the target may have significant dependence on only a small number of possible directions.