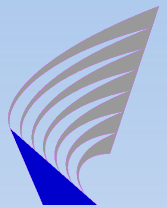


HELSINKI UNIVERSITY OF TECHNOLOGY
LABORATORY OF COMPUTER AND INFORMATION SCIENCE

T-61.6020 Special Course in Computer and Information Science II
Machine Learning: Basic Principles

Linear Models for Regression

Nicolau Gonçalves
68121H

HELSINKI UNIVERSITY OF TECHNOLOGY
LABORATORY OF COMPUTER AND INFORMATION SCIENCE

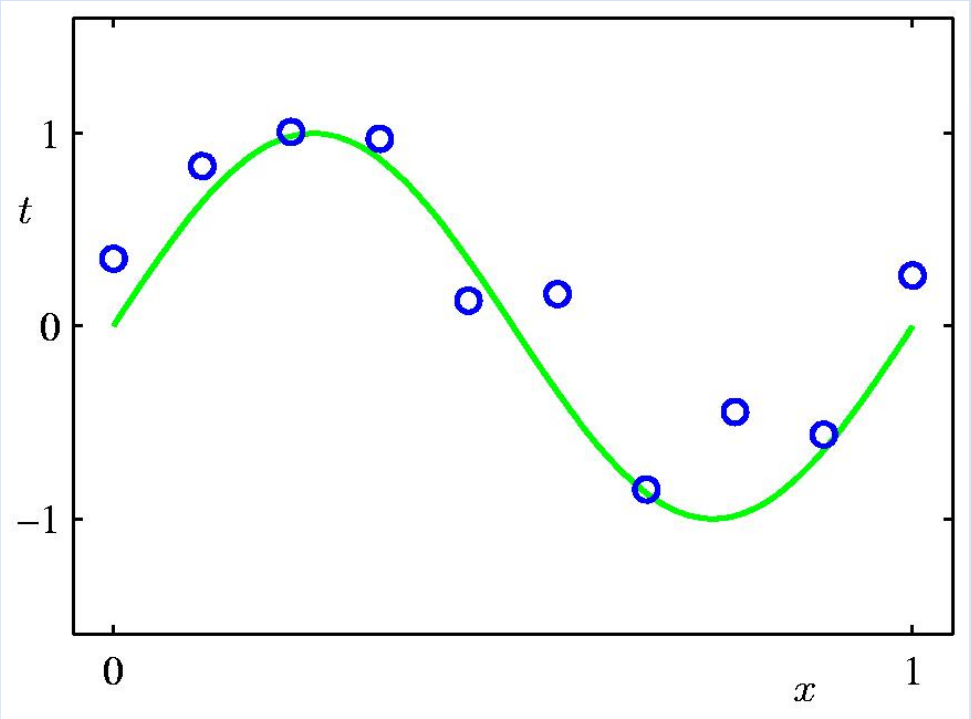
T-61.6020 Special Course in Computer and Information Science II
Machine Learning: Basic Principles

Presentation based on the book:

Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006. ISBN 0387310738

Chapter 3 overview:

- Linear basis function models
- The bias-variance decomposition
- Bayesian Linear Regression
- Bayesian Model Comparison
- The Evidence approximation
- Limitations of Fixed Basis Functions



Espoo, 27/01/07

Chapter Introduction

- Regression: predict the value of one or more continuous *target* variables t given the value of a D-dimensional vector x of input variables.
- Linear regression models is one class of models that shares the property of being linear functions of the adjustable parameters.
- From a probabilistic point of view, we aim to model the predictive distribution $p(t \mid x)$, thus expressing the uncertainty of the prediction
- Linear models have nice analytical properties, despite their limitations, particularly for high dimensional input spaces. Also, they form the foundation for more sophisticated models (later chapters)

T-61.6020 Special Course in Computer and Information Science II
Machine Learning: Basic Principles

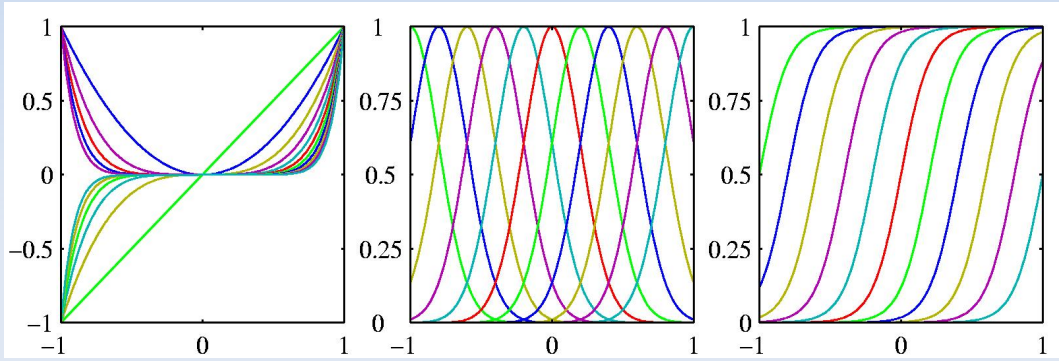
Linear Basis Function Models

A linear combination of fixed functions of input $\mathbf{x}=(x_1,\ldots,x_N)^T$ is of the form:

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

where $\phi_j(x)$ are the basis functions, and the number of parameters is *M*.

Using non-linear basis functions, $y(x,w)$ is a non-linear function of **x**.



Maximum likelihood and least squares

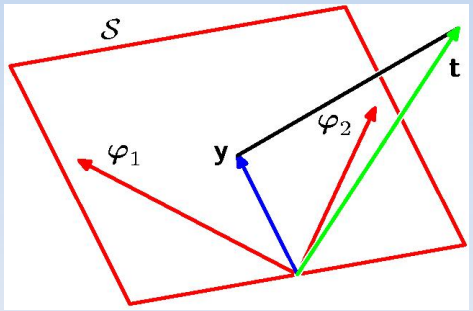
The target variable t is given by $\boldsymbol{t}=y(\boldsymbol{X},\boldsymbol{w})+\epsilon$, and with Gaussian noise we get the likelihood function $p(\mathbf{t}|\boldsymbol{X},w,\beta)=\prod_{n=1}^{N} N(t_n|w^T\phi(x_n),\beta^{-1})$

Maximizing the likelihood function for w, we get the normal equations for the least squares problem $w_{ML}=(\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{t}$, where $\boldsymbol{\Phi}$ is an *N* x *M* matrix with $\Phi_{nj}=\phi_j(x_n)$

T-61.6020 Special Course in Computer and Information Science II
Machine Learning: Basic Principles

Linear Basis Function Models

Geometry of least squares



The least squares solution for **w** corresponds to the choice of of **y** that lies in the subspace *S* and that is closest to **t**, which is spanned by the basis function.

Sequential learning and regularized LS

If the data is large enough, then we can try to use sequential algorithms.

- Applying the stochastic gradient descent, we get the *LMS* algorithm for the sum-of-squares error: $w^{(\tau+1)} = w^{(\tau)} + \eta(t_n - w^{(\tau)T}\phi_n)\phi_n$

By adding a regularization term to the error function we can control over-fit.

- For instance, using the so-called *weight decay*, the error function becomes

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

- With this regularization term, we obtain a simple extension for the LS solution:

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

The Bias-Variance Decomposition

Up until now, we assumed that the form and number of basis functions are fixed. But using the LS method can lead to over-fit in complex models. Also, limiting the number of basis function limits the flexibility of the model.

Using regularization leads to the question on how to find the optimum λ.

So, in order to avoid these problems, we should consider a Bayesian treatment.

But first, let us examine a frequentist view-point of the model complexity: the *bias-variance* trade-off.

The expected squared loss can be written in the form:

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

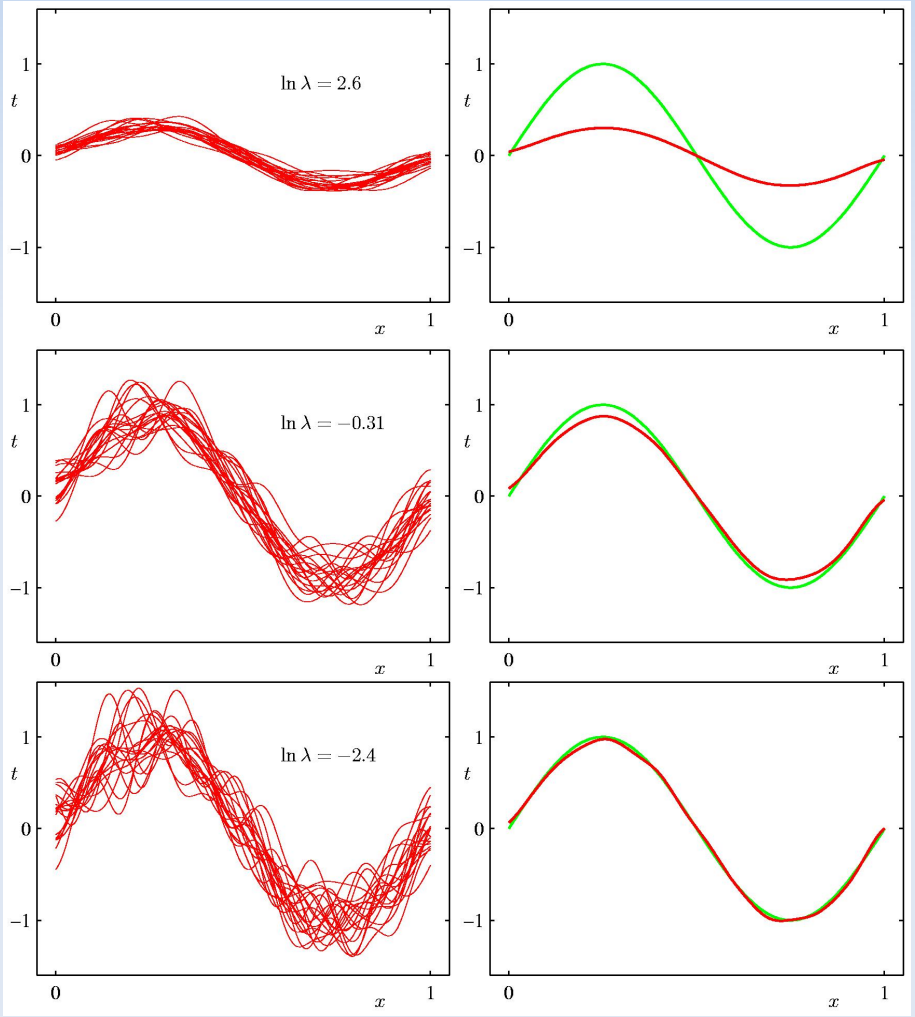
With some manipulation the expected loss becomes:

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

which is of the form: $\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$

The Bias-Variance Decomposition

- There is a trade-off between bias and variance.
- The best model is the one that leads to an optimal balance between bias and variance.
- Small values of λ allow the model to tune to the noise on each individual data set leading to high variance. Conversely, a large λ pulls the weight towards zero, leading to a large bias.
- The bias-variance provides some insight into model complexity, but requires several data sets.
- Therefore, let us proceed to the Bayesian approach



Bayesian Linear Regression
Parameter distribution

How to decide the appropriate model complexity?

R: Maximizing the likelihood: it would lead to complex models and over-fit.

R: Bayesian treatment: avoids the over-fit and leads to an automatic way of determining the model complexity using only the training data.

We start by defining a simple likelihood conjugate prior, a zero-mean Gaussian prior governed by a precision parameter:

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

(we could use more general prior but this simplifies the Bayesian treatment)

This prior, according to Bayes law, leads to the posterior distribution:

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

The log of the posterior distribution is then $\ln p(w|\mathbf{t})=-\frac{\beta}{2}\sum_{n=1}^{N}\{t_n-\boldsymbol{w}^T\phi(x_n)\}^2-\frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w}$

which has the regularization term $\lambda=\frac{\alpha}{\beta}$, in least square sense.

T-61.6020 Special Course in Computer and Information Science II
Machine Learning: Basic Principles

Bayesian Linear Regression
Parameter distribution

- Using $y(x, \boldsymbol{w}) = w_0 + w_1 x$

as an example model, we can observe several important aspects of Bayesian inference.

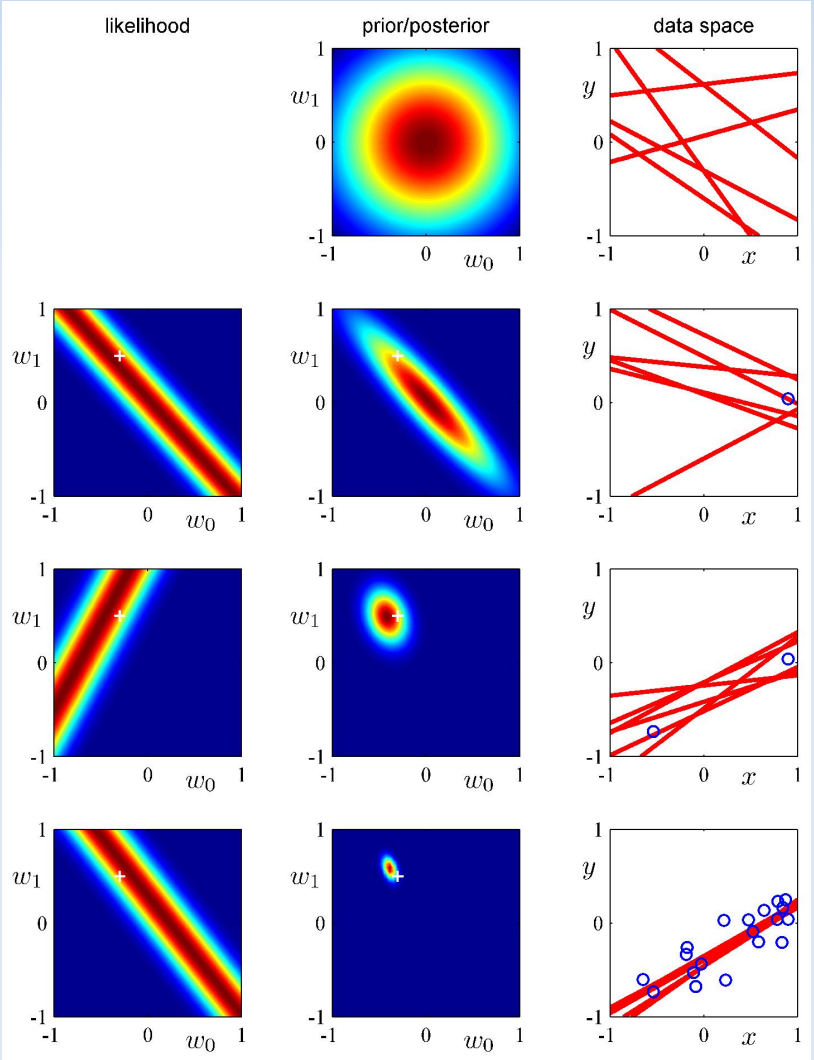
- The data was generated from the function:

$$f(x(n), \boldsymbol{a}) = -0.3 + 0.5\, x(n) + 0.2\mathrm{v}(n)$$

- The likelihood provides a soft constrain that the line must be close to the data point.

- After two data points, because they are sufficient to define a line, the posterior already has a very compact shape.

- With an infinite number of data points, the posterior distribution would be centred on the true parameter values.



T-61.6020 Special Course in Computer and Information Science II
Machine Learning: Basic Principles

Bayesian Linear Regression

Predictive distribution

- Usually we want to evaluate the predictive distribution:

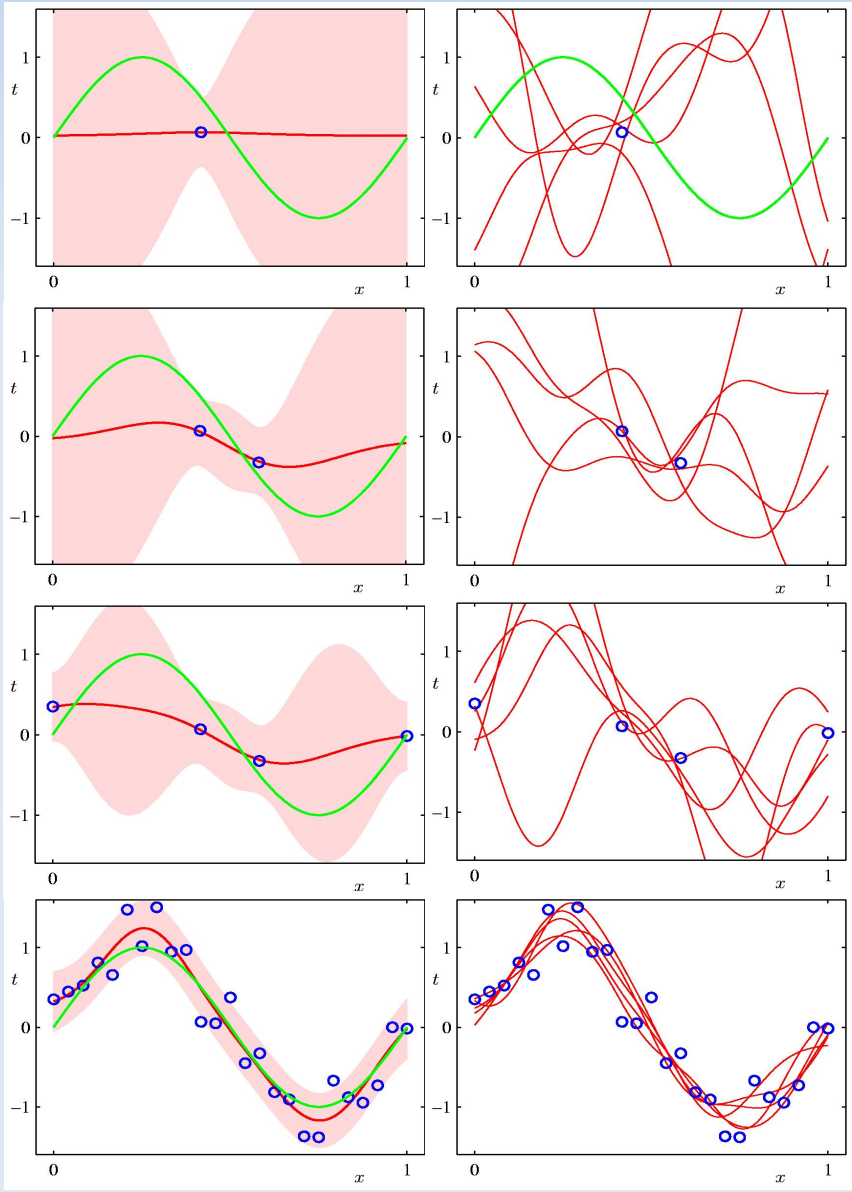
$$p(t|\mathbf{t},\alpha,\beta)=\int p(t|\boldsymbol{w},\beta)\,p(\boldsymbol{w}|\mathbf{t},\alpha,\beta)\,d\,\boldsymbol{w}$$

- This is the convolution of the conditional distribution of the target variable and the posterior weight distribution. This results in:

$$p(t|\boldsymbol{x},\mathrm{t},\alpha,\beta)=N\left(t|\boldsymbol{m}_n^T\phi(\boldsymbol{x}),\sigma_N^2(\boldsymbol{x})\right)$$

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

- The first term represents the data noise whereas the second is associated with the uncertainty of parameters **w.**



Bayesian Linear Regression

Equivalent kernel

- The posterior mean solution has an interesting interpretation, that sets the stage for kernel methods, including Gaussian processes.
- The predictive mean can be written in the form:

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

- And we can also write it as: $y(\boldsymbol{x}, \boldsymbol{m}_N) = \sum_{n=1}^{N} k(\boldsymbol{x}, \boldsymbol{x}_n) t_n$

  where $k(\boldsymbol{x}, \boldsymbol{x}') = \beta \phi(\boldsymbol{x})^T \boldsymbol{S}_N \phi(\boldsymbol{x}')$ is known as the *smoother matrix* or *equivalent kernel*.

- Regression functions which make predictions by taking linear combinations of the training set target values are know as *linear smoothers*.
- The formulation of linear regression in terms of kernel functions suggests that we can define a localized kernel directly and use this to make predictions for the new inputs, given the observed set. This leads to a framework called Gaussian processes (to be reviewed later).

Bayesian Model Comparison

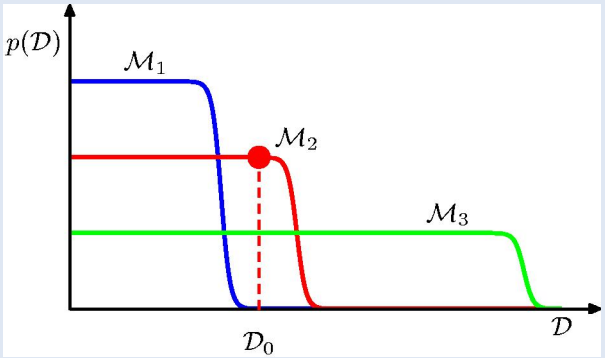
The Bayesian model comparison involves the use of probabilities to represent uncertainty in the choice of model. Suppose we wish to compare a set of L models $\{M_i\}$. The posterior is given by $p(M_i|D) \propto p(M_i) P(D|M_i)$

The prior allows to express a preference (or not) for the different models. The term $P(D|M_i)$ is called *model evidence* or marginal likelihood, and provides the basis for model selection.

The *model evidence* expresses the preference shown from the data to different models. For a model governed by a set of parameters **w**, we have

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

By analysing the marginal likelihood, we can see that it favours models with intermediate complexity.



The Evidence Approximation

Since it would be analytically intractable to marginalize over the hyperparameters (α and β) or **w**, we'll discuss an approximation in which we set the hyperparameters to specific values by maximing the marginal likelihood function by first integrating over the parameters **w**. This is called *evidence approximation*.

The predictive distribution obtained by marginalizing over α, β and **w** is:

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

To determine values for the hyperparameters from the training data alone, we proceed to evaluate the marginal likelihood for the linear basis function model and then to find its maxima.

There are two approaches to do this. We can evaluate the evidence function analytically and then set its derivative to zero to obtain a re-estimation for the hyperparameter's equation (next slides). Or use a technique called expectation maximization (EM) algorithm (to be covered later).

The Evidence Approximation

Evaluation of the EF

- The marginal likelihood is obtained by integrating over the weight parameters w, and we can write the evidence function as:

$$p(\mathbf{t}|\alpha,\beta)=\left(\frac{\beta}{2\pi}\right)^{N/2}\left(\frac{\alpha}{2\pi}\right)^{M/2}\int exp\{-E(\boldsymbol{w})\}d\,\boldsymbol{w}$$

- where

$$E(\boldsymbol{w})=\frac{\beta}{2}\|\mathbf{t}-\boldsymbol{\Phi}\,\boldsymbol{w}\|^2+\frac{\alpha}{2}\boldsymbol{m}_N^T\boldsymbol{m}_N+\frac{1}{2}(\boldsymbol{w}-\boldsymbol{m}_N)^T\boldsymbol{A}(\boldsymbol{w}-\boldsymbol{m}_N)$$

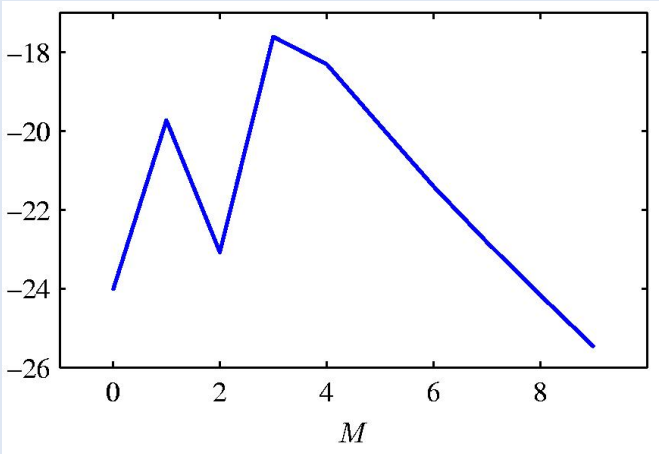
$$A=\nabla\nabla E(\boldsymbol{w})=\alpha\boldsymbol{I}+\beta\boldsymbol{\Phi}^T\boldsymbol{\Phi}$$

$$\boldsymbol{m}_N=\beta\boldsymbol{A}^{-1}\boldsymbol{\Phi}^T\mathbf{t}$$

And the log marginal likelihood is:

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

By maximizing it, we can make a model comparison and get the hyper-parameters.



The Evidence Approximation

Maximizing the EF and effective number of parameters

- First, let us consider maximization with respect to α. Defining $(\beta \Phi^T \Phi)\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i$
- We see that the maximizing value for α is: $\alpha = \frac{\gamma}{\boldsymbol{m}_N^T \boldsymbol{m}_N}$, where $\gamma = \sum_i \frac{\lambda_i}{\alpha + lambda_i}$
- The maximization with respect to β gives: $\frac{1}{\beta} = \frac{1}{N-\gamma} \sum_{n=1}^{N} \{t_n - \boldsymbol{m}_N^T \phi(\boldsymbol{x}_n)\}^2$
- These solutions are implicit, because both γ and $\mathbf{m}_N$ depend on these values.

An iterative method can be used to find the hyper-parameter's optimal values.

Because $\beta \Phi^T \Phi$ is a positive definite matrix, it will have positive eigenvalues and, consequently, γ will vary between 0 and M. For direction where $\gamma >> \alpha$, the corresponding w_i will be close to its maximum likelihood value. Such parameters are called well defined, due to the fact that they are tightly constrained by the data.

- Therefore, γ measures the effective number of well determined parameters

Summary of chapter 3

Linear basis functions can create a model where the regression is a non-linear function of the input vector **x**.

They have nice analytical properties and form the foundation for more sophisticated models.

The Bayesian linear regression method allow the full use of the basis functions avoiding the over-fit to the data and leads to automatic methods of determining the model complexity using only the training data.

Despite all the advantages from using linear models, they have the significant limitations, especially in high-dimensional input spaces. The difficulty stems stems from the assumption are fixed before the training data set is observed. As a consequence, the number of basis functions needs to grow rapidly with the dimensionality *D* of the input space.

In later chapters, more sophisticated models that overcome this limitation, like neural networks or support vector machines, will be presented.

I hope you have enjoyed this review of *linear models for regression.*

Espoo, 27/01/07