Linear Models for Regression

Chia-Ping Chen

Professor
National Sun Yat-sen University
Department of Computer Science and Engineering

Machine Learning

Topics

- Linear Regression Models
- Bias-Variance Decomposition
- Bayesian Linear Regression
- Evidence Approximation
- Limitation of Fixed Basis Functions

Regression

Definition. In **regression** we have **input variables** $\mathbf{x}$ and **target variable** t, where t is continuous and $\mathbf{x}$ may be discrete or continuous. The goal of regression is to predict t given $\mathbf{x}$ via a **regression function** or **prediction function** $y(\mathbf{x})$

$$\mathbf{x} \longrightarrow y(\mathbf{x}) \approx t$$

- polynomial curve fitting
- predict the deal value of a real estate
- predict future price of a stock
- in a game of Go, predict the probability of black winning

Squared Loss

Definition. The **squared loss** of $y(\mathbf{x})$ and t is

$$L(\mathbf{x}, t) = (y(\mathbf{x}) - t)^2$$

Given $\mathbf{x}$, the **expected squared loss** is

$$\begin{aligned}\mathbb{E}[L(\mathbf{x}, t)|\mathbf{x}] &= \mathbb{E}[(y(\mathbf{x}) - t)^2|\mathbf{x}] \\ &= \int p(t|\mathbf{x})(y(\mathbf{x}) - t)^2\, dt\end{aligned}$$

It follows that the regression function that minimizes the expected squared loss is the conditional mean of t

$$y^*(\mathbf{x}) = \int t\, p(t|\mathbf{x})\, dt = \mathbb{E}[t|\mathbf{x}]$$

Approaches to Regression

There are 2 approaches to learning regression function with data. Let $\mathcal{D} = \{(\boldsymbol{x}_n, t_n)\}_{n=1}^N$ be a data set of a regression problem.

- **Deterministic regression.** Assume a regression function $y(\mathbf{x})$ that maps $\mathbf{x}$ to t, and then learn $y(\mathbf{x})$ with $\mathcal{D}$.
- **Probabilistic regression.** Assume a conditional probability model of $p(t|\boldsymbol{x})$ of t given $\mathbf{x}$, and then learn $p(t|\boldsymbol{x})$ with $\mathcal{D}$. Finally, derive a regression function $y(\mathbf{x})$ from the learned $p(t|\boldsymbol{x})$.

Here we emphasize the probabilistic approaches.

Probabilistic Regression

We can learn a regression function from a data set with a probability model.

1. Assume a **parametric conditional model**

   $$p(t|\boldsymbol{x}, \boldsymbol{w})$$

   Here $\boldsymbol{w}$ denotes the set of learnable parameters.

2. Learn $\boldsymbol{w}$ (MLE or Bayesian learning) with $\mathcal{D}$.

3. Derive a regression function by substitution of point estimate of $\boldsymbol{w}$ (MLE, MAP) or integration over distribution of $\boldsymbol{w}$ (Bayesian).

Gaussian Noise Model

Definition. In **Gaussian noise model**, we assume that t is the sum of a function of $\mathbf{x}$ and a Gaussian noise with zero mean.

That is

$$t = u(\mathbf{x}) + \epsilon, \ \epsilon \sim \mathcal{N}(\epsilon | 0, \beta^{-1})$$

It follows that

$$p(t|\boldsymbol{x}) = \mathcal{N}(t|u(\boldsymbol{x}), \beta^{-1})$$

Linear Regression Model

Definition. In **linear regression model**, we assume a Gaussian noise model

$$t = u(\mathbf{x}) + \epsilon$$

and $u(\mathbf{x})$ is approximated by a linear combination of fixed **basis functions**

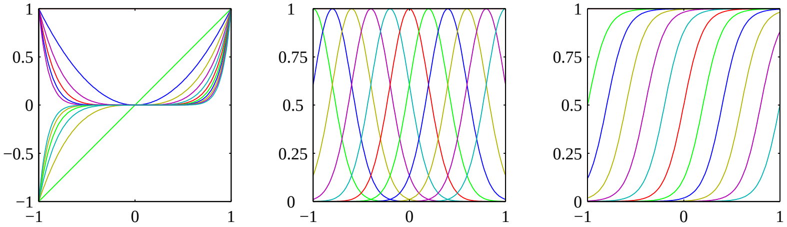
$$u(\mathbf{x}) = \sum_{i=1}^{M} w_i \phi_i(\mathbf{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\mathbf{x})$$

$\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x})]^T$ is the **feature vector** of $\mathbf{x}$.

It follows that

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

Basis Functions



Examples of polynomial, Gaussian, and Sigmoidal basis functions.

Learning Parameters

In a linear regression model, we have a Gaussian conditional model

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

The basis functions $\boldsymbol{\phi}(\boldsymbol{x})$ are given. The parameters $\boldsymbol{w}$ and β are to be learned from data.

Data Likelihood

Let $\mathcal{D} = \{(\boldsymbol{x}_n, t_n)\}_{n=1}^N$ be a data set. The likelihood of a data point $(\boldsymbol{x}_n, t_n)$ is

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

The data likelihood of $\mathcal{D}$ is

$$p(\mathcal{D}|\boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\boldsymbol{w}^T\boldsymbol{\phi}_n, \beta^{-1})$$

Maximum Likelihood Estimate

The log likelihood of $\mathcal{D}$ is

$$\log p(\mathcal{D}|\boldsymbol{w}, \beta) = \sum_{n=1}^{N} \log \mathcal{N}(t_n|\boldsymbol{w}^T \boldsymbol{\phi}_n, \beta^{-1})$$
$$= \frac{N}{2} \log \beta - \frac{N}{2} \log(2\pi) - \frac{\beta}{2} \sum_{n=1}^{N} [t_n - \boldsymbol{w}^T \boldsymbol{\phi}_n]^2$$

At $\boldsymbol{w}_{\mathsf{ML}}$

$$\boldsymbol{\nabla}_{\boldsymbol{w}} p(\mathcal{D}|\boldsymbol{w}, \beta) = \beta \sum_{n=1}^{N} \left\{ t_n - \boldsymbol{w}_{\mathsf{ML}}^T \boldsymbol{\phi}_n \right\} \boldsymbol{\phi}_n = \mathbf{0}$$

$$\Rightarrow \quad \sum_{n=1}^{N} \boldsymbol{w}_{\mathsf{ML}}^T \boldsymbol{\phi}_n \boldsymbol{\phi}_n = \sum_{n=1}^{N} t_n \boldsymbol{\phi}_n$$

Design Matrix

Definition. The **design matrix** of $\mathcal{D}$ is

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \dots & \phi_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{x}_N) & \dots & \phi_M(\boldsymbol{x}_N) \end{bmatrix}$$

The row vectors and column vectors are

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1^T \\ \vdots \\ \boldsymbol{\phi}_N^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}_1 & \dots & \boldsymbol{\varphi}_M \end{bmatrix}$$

The transpose is

$$\boldsymbol{\Phi}^T = \begin{bmatrix} \boldsymbol{\phi}_1 & \dots & \boldsymbol{\phi}_N \end{bmatrix}$$

Matrix Equation of $\boldsymbol{w}_{\mathrm{ML}}$

A maximum likelihood estimate of $\boldsymbol{w}$ satisfies

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

Since

$$\begin{aligned}\sum_{n=1}^{N}\boldsymbol{w}_{\mathsf{ML}}^T\boldsymbol{\phi}_n\boldsymbol{\phi}_n = \sum_{n=1}^{N}\boldsymbol{\phi}_n\boldsymbol{\phi}_n^T\boldsymbol{w}_{\mathsf{ML}} &= \left(\sum_{n=1}^{N}\boldsymbol{\phi}_n\boldsymbol{\phi}_n^T\right)\boldsymbol{w}_{\mathsf{ML}} \\ &= \mathbf{\Phi}^T\mathbf{\Phi}\boldsymbol{w}_{\mathsf{ML}}\end{aligned}$$

and

$$\sum_{n=1}^{N} t_n\boldsymbol{\phi}_n = \mathbf{\Phi}^T\mathbf{t}, \text{ where } \mathbf{t} = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}$$

The equation to be satisfied by $\boldsymbol{w}_{\mathsf{ML}}$ can be re-written as

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

Geometry of Least Squares

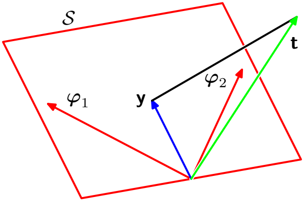
From linear algebra

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

is the normal equation of the system of linear equations

$$\boldsymbol{\Phi}\boldsymbol{w} = \mathbf{t}$$

and $\boldsymbol{w}_{\mathsf{ML}}$ is a least-squares solution. Furthermore, $\boldsymbol{\Phi}\boldsymbol{w}_{\mathsf{ML}} = \mathbf{y}$ is the projection of $\mathbf{t}$ to the space spanned by $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M$ where $\boldsymbol{\varphi}_i$ is the ith column of $\boldsymbol{\Phi}$.



Sequential Gradient Descent

Computing gradient using the entire set may be expensive.

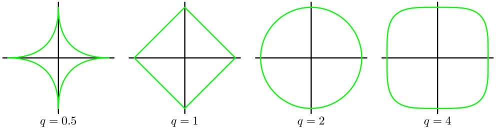
Sequential learning. One can estimate the gradient of the loss function with a random example, and then update parameters by

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} - \eta \nabla_{\boldsymbol{w}} E_n(\boldsymbol{w}^{(\tau)})$$

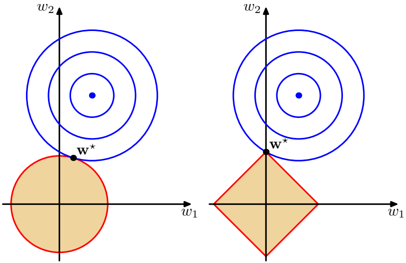
With squared loss $E_n = \frac{1}{2}(t_n - \boldsymbol{w}^T \boldsymbol{\phi}_n)^2$, we have

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

Norm Penalty



Contours of L^q-norm in 2-D weight space.



Regularization with norm-penalty using L^2-norm and L^1-norm.

Bias-Variance Decomposition

Optimal Regression Function

Let t and $\mathbf{x}$ be the target variable and the input variables. Given $\mathbf{x}$, the optimal regression function that minimizes the expected squared loss between the prediction and the target is the conditional mean

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}]$$

Random Learning Outcome

Let the regression function learned by data set $\mathcal{D}$ be denoted by $y(\boldsymbol{x};\mathcal{D})$. Learning from different data sets

$$\mathcal{D}_1, \mathcal{D}_2, \ldots$$

leads to different regression functions

$$y(\boldsymbol{x};\mathcal{D}_1), y(\boldsymbol{x};\mathcal{D}_2), \ldots$$

Expected Squared Loss

Conditioning on data set $\mathcal{D}$, the **expected squared loss** of the learned regression function is

$$\begin{aligned}\mathbb{E}[L|\mathcal{D}] &= \iint \{y(\boldsymbol{x};\mathcal{D}) - t\}^2 p(\boldsymbol{x},t)\,dt\,d\boldsymbol{x} \\ &= \int \{y(\boldsymbol{x};\mathcal{D}) - h(\boldsymbol{x})\}^2 p(\boldsymbol{x})\,d\boldsymbol{x} + \iint \{h(\boldsymbol{x}) - t\}^2 p(\boldsymbol{x},t)\,d\boldsymbol{x}\,dt \\ &= \int \{y(\boldsymbol{x};\mathcal{D}) - h(\boldsymbol{x})\}^2 p(\boldsymbol{x})\,d\boldsymbol{x} + \text{noise}\end{aligned}$$

where

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

Note that the noise term is invariant with respect to $\mathcal{D}$.

Total Expected Squared Loss

The **total expected squared loss** is

$$\begin{aligned}
\mathbb{E}[L] &= \mathbb{E}[\mathbb{E}[L|\mathcal{D}]] \\
&= \mathbb{E}_{\mathcal{D}}\left[\int \{y(\boldsymbol{x};\mathcal{D}) - h(\boldsymbol{x})\}^2 p(\boldsymbol{x}) d\boldsymbol{x}\right] + \text{noise} \\
&= \mathbb{E}_{\mathcal{D}}\left[\int \left\{y(\boldsymbol{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\boldsymbol{x};\mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\boldsymbol{x};\mathcal{D})] - h(\boldsymbol{x})\right\}^2 p(\boldsymbol{x}) d\boldsymbol{x}\right] \\
&\qquad + \text{noise} \\
&= \mathbb{E}_{\mathcal{D}}\left[\int \left\{y(\boldsymbol{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\boldsymbol{x};\mathcal{D})]\right\}^2 p(\boldsymbol{x}) d\boldsymbol{x}\right] \\
&\qquad + \int \left\{\mathbb{E}_{\mathcal{D}}[y(\boldsymbol{x};\mathcal{D})] - h(\boldsymbol{x})\right\}^2 p(\boldsymbol{x}) d\boldsymbol{x} + \text{noise} \\
&= \text{variance} + (\text{bias})^2 + \text{noise}
\end{aligned}$$

Bias and Variance

- **Bias.** The degree that $y(\boldsymbol{x};\mathcal{D})$ is different from the optimum regression function $h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}]$ on average

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

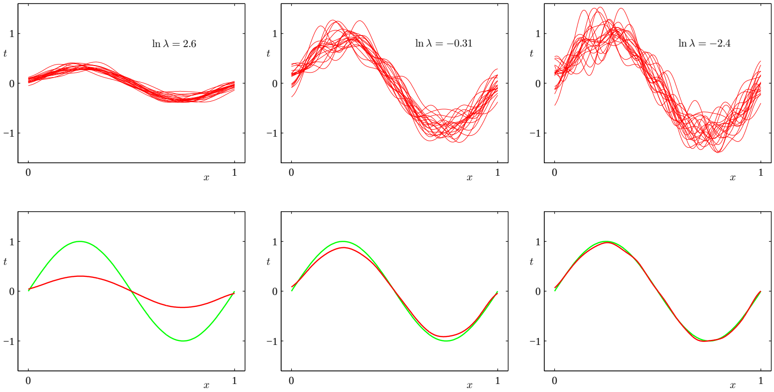
- **Variance.** The degree that one instance of $y(\boldsymbol{x};\mathcal{D})$ is different from its mean on average

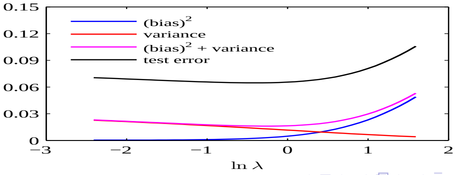
$$\text{variance} = \mathbb{E}_{\mathcal{D}}\left[\int \{y(\boldsymbol{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\boldsymbol{x};\mathcal{D})]\}^2 p(\boldsymbol{x}) d\boldsymbol{x}\right]$$

Trade-off

- Simple model: large squared bias and small variance
- Complex model: small squared bias and large variance

Sinusoidal Data, $N = 25, L = 100$





Bayesian Linear Regression

Conjugate Prior

For a linear regression model (with Gaussian noise), a conjugate prior of the parameters is Gaussian.

Let $\mathcal{D} = \{(\boldsymbol{x}_n, t_n)\}_{n=1}^N$ be a data set of a regression problem. The conditional likelihood of $\mathcal{D}$ is

$$p(\mathcal{D}|\boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\boldsymbol{w}^T\boldsymbol{\phi}_n, \beta^{-1})$$

$$\Rightarrow \quad \log p(\mathcal{D}|\boldsymbol{w}, \beta) = \frac{N}{2}\log\beta - \frac{N}{2}\log(2\pi) - \frac{\beta}{2}\sum_{n=1}^{N}[t_n - \boldsymbol{w}^T\boldsymbol{\phi}_n]^2$$

Since $p(\mathcal{D}|\boldsymbol{w}, \beta)$ is log quadratic in $\boldsymbol{w}$, a conjugate prior of $\boldsymbol{w}$ is Gaussian.

Posterior Distribution

Let the prior of the parameters $\boldsymbol{w}$ be Gaussian

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

Then the posterior distribution of $\boldsymbol{w}$ is also Gaussian

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

It can be shown, by the Bayes' rule and completing squares in the posterior, that

$$\begin{aligned} \boldsymbol{S}_N^{-1} &= \boldsymbol{S}_0^{-1} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi} \\ \boldsymbol{m}_N &= \boldsymbol{S}_N (\boldsymbol{S}_0^{-1} \boldsymbol{m}_0 + \beta \boldsymbol{\Phi}^T \mathbf{t}) \end{aligned}$$

Special Case

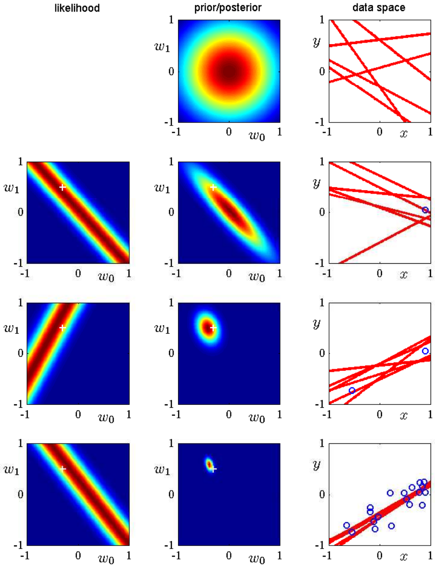
In the following discussion, we assume a zero-mean isotropic Gaussian prior distribution of $\boldsymbol{w}$

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

In this case, the Gaussian posterior has the following mean vector and covariance matrix

$$\begin{aligned} \boldsymbol{S}_N^{-1} &= \alpha\boldsymbol{I} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi} \\ \boldsymbol{m}_N &= \beta\boldsymbol{S}_N\boldsymbol{\Phi}^T\mathbf{t} \end{aligned}$$

Sequential Posterior Distribution



Predictive Distribution

In Bayesian learning, the predictive distribution is obtained by marginalization over the distribution of the parameters.

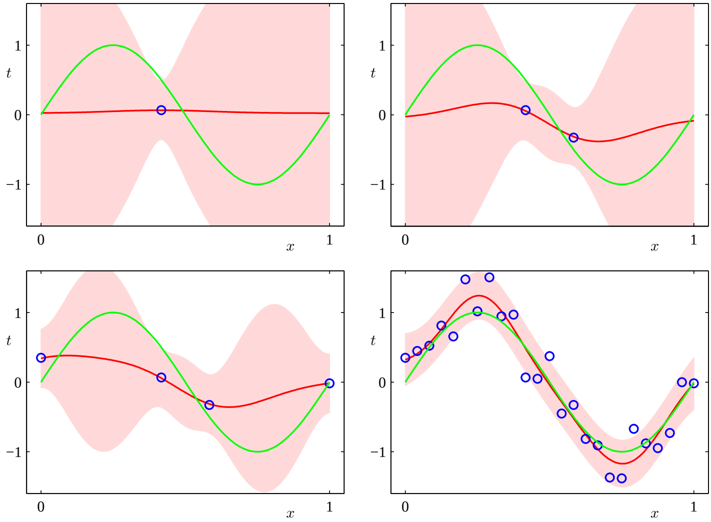
In this case, the predictive distribution of t is

$$\begin{aligned} p(t|\boldsymbol{x}, \mathcal{D}, \alpha, \beta) &= \int p(t, \boldsymbol{w}|\boldsymbol{x}, \mathcal{D}, \alpha, \beta) d\boldsymbol{w} \\ &= \int \underbrace{p(t|\boldsymbol{x}, \boldsymbol{w}, \beta)}_{\mathcal{N}(t|\boldsymbol{w}^T\boldsymbol{\phi}, \beta^{-1})} \underbrace{p(\boldsymbol{w}|\mathcal{D}, \alpha, \beta)}_{\mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_N, \boldsymbol{S}_N)} d\boldsymbol{w} \\ &= \mathcal{N}(t|\boldsymbol{m}_N^T\boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}) + \beta^{-1}) \end{aligned}$$

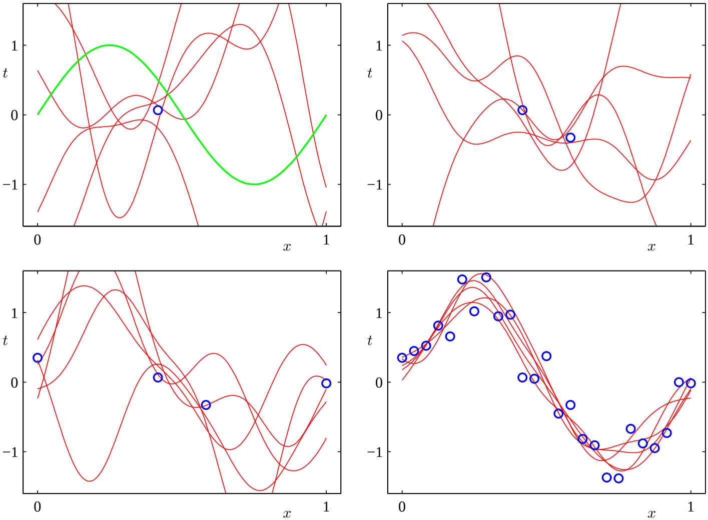
It follows that the optimal prediction is

$$y(\boldsymbol{x}) = \mathbb{E}[t|\boldsymbol{x}] = \boldsymbol{m}_N^T\boldsymbol{\phi}(\boldsymbol{x})$$

Predictive Distribution: Sequential Updates



Sample Output Functions



Equivalent Kernel

In linear regression model with Gaussian prior and Gaussian noise, the optimal prediction function can be re-written by a **kernel function**.

That is

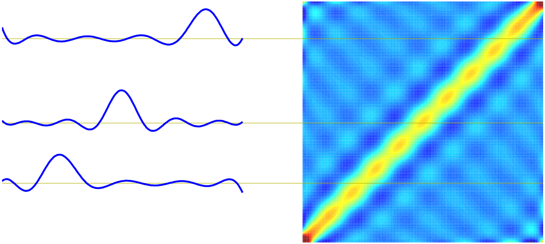
$$\begin{aligned} y(\boldsymbol{x}) &= \boldsymbol{m}_N^T \boldsymbol{\phi}(\boldsymbol{x}) \\ &= \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{m}_N \\ &= \beta \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\Phi}^T \mathbf{t} \\ &= \sum_{n=1}^{N} \beta \boldsymbol{\phi}(\boldsymbol{x})^T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

where $k(\boldsymbol{x}, \boldsymbol{x}') = \beta \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}')$ is a kernel function.

Basis Function and Kernel Function

$k(\boldsymbol{x}, \boldsymbol{x}')$ depends the basis functions $\boldsymbol{\phi}(\boldsymbol{x})$ and the design matrix $\boldsymbol{\Phi}$.

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{x}') &= \beta \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}') \\ &= \beta \boldsymbol{\phi}(\boldsymbol{x})^T \left(\alpha \boldsymbol{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\phi}(\boldsymbol{x}') \end{aligned}$$



A kernel function based on Gaussian basis functions.

Properties of Kernel Functions

- $k(\boldsymbol{x}, \boldsymbol{x}')$ is **symmetric**
- $k(\boldsymbol{x}, \boldsymbol{x}')$ is **localized**
- The **covariance** of the prediction values at two points $\boldsymbol{x}$ and $\boldsymbol{x}'$ is related to $k(\boldsymbol{x}, \boldsymbol{x}')$

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

- $k(\boldsymbol{x}, \boldsymbol{x}')$ can be expressed as an **inner product**

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

Bayesian Model Comparison

Model Comparison

Definition. In **model comparison**, we compare a set of candidate models

$$\mathcal{M}_1, \ldots, \mathcal{M}_L$$

based on a data set $\mathcal{D}$.

- For example, we may want to compare the models of different orders in the polynomial curve-fitting problem.
- We did this with a data set different from the training set.

Model Prior and Model Evidence

Definition. Our preference, if any, can be quantified through **model prior** $p(\mathcal{M}_i)$. The preference by data is quantified through **model evidence** $p(\mathcal{D}|\mathcal{M}_i)$.

By Bayes' rule, the **model posterior** is related to the model prior and the model evidence by

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

To make prediction, one can use model averaging or the single most probable model.

Marginal Likelihood

Definition. In Bayesian framework, where the parameters are treated as random variables, the model evidence $p(\mathcal{D}|\mathcal{M}_i)$ is obtained through **marginalization** over $\boldsymbol{w}_i$

$$\begin{aligned} p(\mathcal{D}|\mathcal{M}_i) &= \int p(\mathcal{D}, \boldsymbol{w}_i|\mathcal{M}_i)\, d\boldsymbol{w}_i \\ &= \int p(\mathcal{D}|\boldsymbol{w}_i, \mathcal{M}_i) p(\boldsymbol{w}_i|\mathcal{M}_i)\, d\boldsymbol{w}_i \end{aligned}$$

In Bayesian learning framework, the model evidence $p(\mathcal{D}|\mathcal{M}_i)$ is also called the **marginal likelihood**.

Analysis of Marginal Likelihood

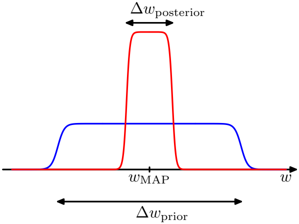
Consider a model with a parameter w. The marginal likelihood is

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w)dw$$

Assuming flat prior and posterior, we have

$$p(\mathcal{D}) \approx p(\mathcal{D}|w_{\mathsf{MAP}}) \left(\frac{1}{\Delta w_{\mathsf{prior}}} \right) \Delta w_{\mathsf{posterior}}$$

$$\Rightarrow \quad \log p(\mathcal{D}) \approx \log p(\mathcal{D}|w_{\mathsf{MAP}}) + \log \left(\frac{\Delta w_{\mathsf{posterior}}}{\Delta w_{\mathsf{prior}}} \right)$$



Trade-Off

For a model with M parameters, the log model evidence is

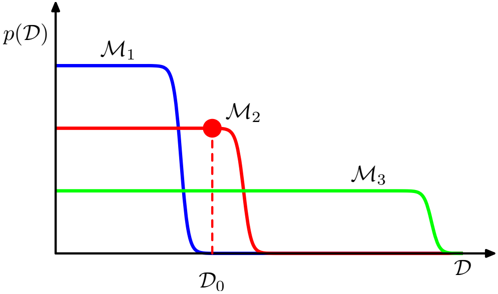
$$\log p(\mathcal{D}) \approx \log p(\mathcal{D}|w_{\mathsf{MAP}}) + M \log \left(\frac{\Delta w_{\mathsf{posterior}}}{\Delta w_{\mathsf{prior}}} \right)$$

An optimal model achieves the best trade-off of two terms.

- The first term favors data likelihood.
- The second term penalizes fine-tuning the parameters to the model.

Comparison of Model Evidence

Marginal likelihood favors models of intermediate complexity.



The model complexity $\mathcal{M}_1 < \mathcal{M}_2 < \mathcal{M}_3$.

Evidence Approximation

Fully Bayesian Linear Regression

Idea. The hyperparameters, like the parameters, are unknown or uncertain to us. Thus, we can introduce **hyperpriors** for the hyperparameters.

The predictive distribution is obtained by marginalization over the distribution of the hyperparameters and the parameters. That is

$$\begin{aligned} p(t|\boldsymbol{x},\mathcal{D}) &= \iiint p(t,\boldsymbol{w},\alpha,\beta|\boldsymbol{x},\mathcal{D})\,d\boldsymbol{w}\,d\alpha\,d\beta \\ &= \iiint p(t|\boldsymbol{x},\boldsymbol{w},\beta)p(\boldsymbol{w}|\mathcal{D},\alpha,\beta)p(\alpha,\beta|\mathcal{D})\,d\boldsymbol{w}\,d\alpha\,d\beta \end{aligned}$$

Marginal Likelihood

An alternative is to use a point estimate of the hyperparameters. One point estimate comes from maximizing the **marginal likelihood** (a.k.a. **model evidence**) as a function of the hyperparameters.

Recall that marginal likelihood is obtained by marginalization over the model parameters

$$\begin{aligned} p(\mathcal{D}|\alpha, \beta) &= \int p(\mathcal{D}, \boldsymbol{w}|\alpha, \beta)\, d\boldsymbol{w} \\ &= \int p(\mathcal{D}|\boldsymbol{w}, \beta) p(\boldsymbol{w}|\alpha)\, d\boldsymbol{w} \end{aligned}$$

Evidence of a Linear Regression Model

For linear regression with Gaussian noise and Gaussian prior, we have

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

where $E(\boldsymbol{w}) = \frac{\beta}{2}\|\mathbf{t}-\boldsymbol{\Phi}\boldsymbol{w}\|^2 + \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w}$ is a regularized error function. It can be written as

$$E(\boldsymbol{w}) = E(\boldsymbol{m}_N) + \frac{1}{2}(\boldsymbol{w}-\boldsymbol{m}_N)^T\boldsymbol{A}(\boldsymbol{w}-\boldsymbol{m}_N)$$

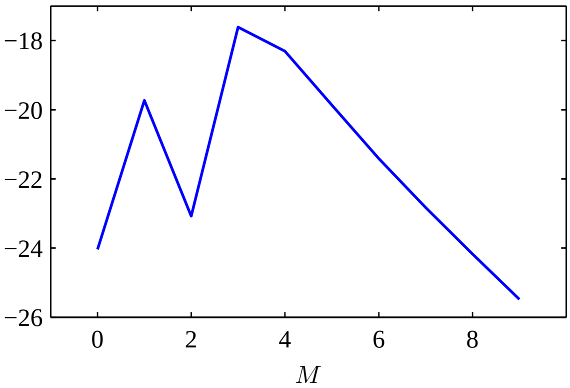
where $\boldsymbol{A} = \alpha\boldsymbol{I} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ and $\boldsymbol{m}_N = \beta\boldsymbol{A}^{-1}\boldsymbol{\Phi}^T\mathbf{t}$. It follows that

$$\int \exp\left\{-E(\boldsymbol{w})\right\} d\boldsymbol{w} = \exp\left\{-E(\boldsymbol{m}_N)\right\}(2\pi)^{M/2}|\boldsymbol{A}|^{-1/2}$$

Thus

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

Dependence of Model Evidence on M



Plot of the log model evidence $\log p(\mathcal{D}|\alpha, \beta)$ given α and β.
$\mathcal{D}$ is the sinusoidal data and M is the polynomial order.

Maximization of Evidence

The log evidence $\log p(\mathcal{D}|\alpha, \beta)$ of a linear model is

$$\frac{M}{2}\log\alpha + \frac{N}{2}\log\beta - E(\boldsymbol{m}_N) - \frac{1}{2}\log|\boldsymbol{A}| - \frac{N}{2}\log(2\pi)$$

We want to find α and β that maximizes it.

The determinant $|\boldsymbol{A}|$ can be expressed by the eigenvalues of $\boldsymbol{A}$. Recall $\boldsymbol{A} = \alpha\boldsymbol{I} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi}$. Let $\lambda_1, \ldots, \lambda_M$ be the eigenvalues of $\beta\boldsymbol{\Phi}^T\boldsymbol{\Phi}$. Then $\lambda_1 + \alpha, \ldots, \lambda_M + \alpha$ are the eigenvalues of $\boldsymbol{A}$ and

$$|\boldsymbol{A}| = \prod_{i=1}^{M}(\lambda_i + \alpha)$$

So

$$\log|\boldsymbol{A}| = \sum_{i=1}^{M}\log(\lambda_i + \alpha)$$

Stationary Points of α

Noting that λ_i is independent of α, we have

$$\frac{d}{d\alpha}\log|\boldsymbol{A}| = \frac{d}{d\alpha}\left(\sum_{i=1}^{M}\log(\lambda_i+\alpha)\right) = \sum_{i=1}^{M}\frac{1}{\lambda_i+\alpha}$$

For the stationary points of α, we have

$$0 = \frac{M}{2\alpha} - \frac{1}{2}\boldsymbol{m}_N^T\boldsymbol{m}_N - \frac{1}{2}\sum_{i=1}^{M}\frac{1}{\lambda_i+\alpha}$$

Thus

$$\alpha\,\boldsymbol{m}_N^T\boldsymbol{m}_N = M - \alpha\sum_{i=1}^{M}\frac{1}{\lambda_i+\alpha} = \sum_{i=1}^{M}\frac{\lambda_i}{\lambda_i+\alpha} = \gamma$$

or

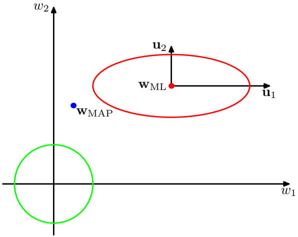
$$\alpha = \frac{\gamma}{\boldsymbol{m}_N^T\boldsymbol{m}_N},\ \text{where } \gamma = \sum_{i=1}^{M}\frac{\lambda_i}{\lambda_i+\alpha}$$

Effective Number of Parameters

γ can be interpreted as the **effective number of parameters**

$$\gamma = \sum_{i=1}^{M} \frac{\lambda_i}{\lambda_i + \alpha} = \sum_{i=1}^{M} n_i, \text{ where } 0 \leq n_i \leq 1$$

n_i is a measure of the degree that parameter i is influenced by data.



As $\lambda_1 < \alpha < \lambda_2$, w_1 is less influenced by data then w_2.

Stationary Points of β

Noting that λ_i is proportional to β, we have

$$\frac{d}{d\beta}\log|\boldsymbol{A}| = \frac{d}{d\beta}\left(\sum_{i=1}^{M}\log(\lambda_i+\alpha)\right) = \sum_{i=1}^{M}\frac{\frac{\lambda_i}{\beta}}{\lambda_i+\alpha}$$

$$= \frac{1}{\beta}\sum_{i=1}^{M}\frac{\lambda_i}{\lambda_i+\alpha} = \frac{\gamma}{\beta}$$

So the stationary points of β satisfy

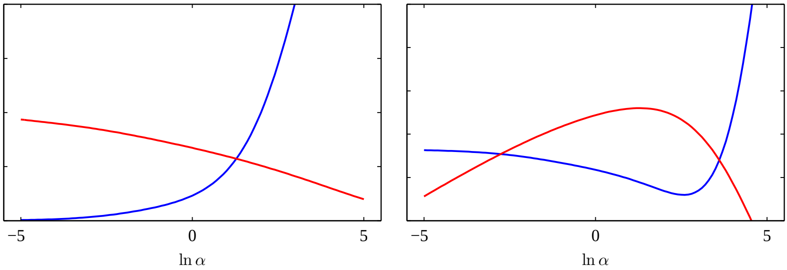
$$0 = \frac{N}{2\beta} - \frac{1}{2}\sum_{n=1}^{N}\{t_n - \boldsymbol{m}_N^T\boldsymbol{\phi}_n\}^2 - \frac{\gamma}{2\beta}$$

Thus

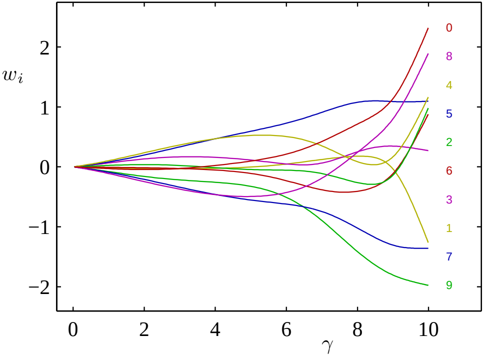
$$\frac{1}{\beta} = \frac{1}{N-\gamma}\sum_{n=1}^{N}\{t_n - \boldsymbol{m}_N^T\boldsymbol{\phi}_n\}^2$$

Example: Sinusoidal Synthetic Data

There are 9 Gaussian basis functions, so $M = 10$. $\beta = 11.1$.



Left: plot of γ (red) and $2\alpha E_W(\boldsymbol{m}_N)$ (blue)
Right: plot of $\log p(\mathcal{D}|\alpha, \beta)$ (red) and test set error (blue)



Plot of the learned values of 10 parameters ($\boldsymbol{m}_N$).

$N \gg M$

When the number of data points is much larger than the number of parameters, all the parameters are effectively determined by data, so $\gamma = M$. It follows that the optimal hyperparameters satisfy

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

where

$$E_W(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \text{ and } E_D(\boldsymbol{w}) = \frac{1}{2}\|\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{w}\|^2$$

They are iterative equations.

Limitation of Fixed Basis Functions

With fixed basis functions, the number of basis functions grows rapidly with the dimension of the input space.

- The intrinsic dimensionality of data is often small.
- The target values may have significant dependence only on a small number of directions within the data space.

Neural networks can adapt the parameters of the basis functions according to data.