

---

# Probabilistic Deep Learning:

## 3. Linear Model for Regression

ESC 2024 Winter Session 3주차  
김상민, 김채영, 김효은, 조준태



# Contents

---

1. Linear Basis Function Models
2. Frequentist and Bayesian Perspective on Model Complexity
3. Bayesian Model Comparison
4. Evidence Approximation
5. Bayesian Framework

# 1

## Linear Basis Function Models

---

Linear Basis Function Models, Maximum likelihood and least squares, Regularization

# Linear Basis Function Models

---

regression에서 가장 간단한 선형 모델은 input variables의 linear combination 형태

즉, linear regression model

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \dots + w_D x_D \quad (3.1), \text{ where } \mathbf{x} = (x_1, \dots, x_D)^T$$

핵심 매개 변수  $w_0, \dots, w_D$ 의 선형 함수 형태라는 것

but 입력 변수  $x_i$ 의 선형 함수라는 점으로 인한 한계점도 존재

ex) 선형성, 독립성, 동분산성 등

- 선형성을 가정하지만, 실제 데이터에서 항상 선형관계가 존재하는 것은  $X \rightarrow$  변수 간의 관계를 적절하게 파악 못 함
- Data의 독립성을 가정하지만, 실제로는 data points 사이에 상관 관계가 존재하는 경우가 많음  $\rightarrow$  오차를 과대 추정하거나 과소 추정

$\Rightarrow$  input variables의 non linear function인 basis function(기저 함수)의 linear combination을 활용해보자!

# Linear Basis Function Models

---

linear regression model의 한계점을 해결하기 위해

input variables의 non linear function 즉 basis function의 linear combination을 활용한 것이 아래의 함수

$$y(\mathbf{x}, \mathbf{w}) = w_o + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) \quad (3.2), \text{ where } \phi_j(\mathbf{x}) = \text{basis function}$$

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) \quad (3.3), \text{ where } \mathbf{x} = (w_0, \dots, w_{M-1})^T \text{ and } \phi = (\phi_0, \dots, \phi_{M-1})^T$$

*offset*

-  $w_0$ 는 data의 offset을 표현할 수 있도록 해주는 bias parameter

\*시계열 data → 특정 시간 단위로 data를 이동시키는데 사용

- 위 식에서 볼 수 있듯, 원래의 변수가  $x$  벡터로 구성되어 있다면,

⇒ 분석 시작 시점을 결정

basis function  $\phi_j(\mathbf{x})$ 의 형태로 표현이 가능

\*회귀 분석

→ 종속변수 값이 독립변수에 의해 조절되어야 할 필요가 있을 때 사용

⇒ 해당 변수의 영향을 고정

⇒ data 분석 때 기준점을 설정하거나

특정 변수의 영향력을 고정시키는 기능!

# Linear Basis Function Models

$$y(\mathbf{x}, \mathbf{w}) = w_o + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) \quad (3.2), \text{ where } \phi_j(\mathbf{x}) = \text{basis function}$$

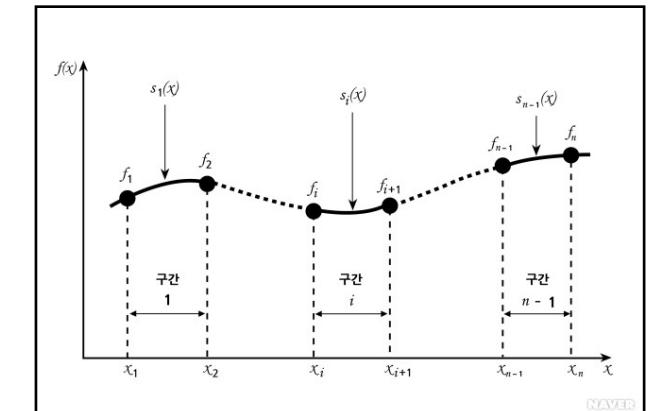
$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) \quad (3.3), \text{ where } \mathbf{x} = (w_0, \dots, w_{M-1})^T \text{ and } \phi = (\phi_0, \dots, \phi_{M-1})^T$$

- 입력 변수들의 비선형 함수 형태인 basis function을 활용하므로  
입력 벡터  $\mathbf{x}$ 의 관점에서는  $y$ 를 비선형 함수 형태로 정의 가능,  
 $\mathbf{w}$ 의 관점에서는 선형 함수 형태로 정의 가능

ex) week1 polynomial regression

→ 입력 변수:  $x$ , 기저함수  $\phi_j(x)$ :  $x^j$

- basis function이 입력 변수  $\mathbf{x}$ 에 대해 global한 함수이므로  
input space의 한 영역에서 발생하는 변화가  
다른 영역에도 영향을 미침  
→ 함수를 근사할 때 문제가 발생  
→ input space를 여러 영역으로 쪼개고,  
영역마다 다른 polynomial을 fitting하여 해결 가능  
⇒ "spline function"



# Linear Basis Function Models

---

## Possible basis function choices

### 1. Gaussian basis function

$$\phi_j(x) = \exp\left\{-\frac{(x - \mu_j)^2}{2s^2}\right\} \quad (3.4)$$

where  $\mu_j$ 는 입력 공간에서 기저 함수의 위치를 결정,

$s$ 는 spatial scale을 결정

$\mu_j$ 는 함수의 중심

→  $\mu_j$ 가 변하면 함수의 '위치'가 변화

즉, X축 상에서 어디에 그래프가 위치할지 결정

⇒  $\mu_j$  값이 다른 여러 개의 가우시안 기저 함수를 사용하면

입력공간을 잘 나타낼 수 있는 기저를 형성할 수 있음

$s$ 는 함수의 표준편차

→  $s$ 가 커지면 함수의 폭이 넓어지고, 작아지면 좁아짐

→ 함수의 '스케일'을 조절

⇒ data point 주변의 어느정도 공간에 대해

함수가 영향력을 미치는지 결정

반드시 확률적 해석이 필요한 것은 X

향후 모델에서 기저 함수에 adaptive parameter  $w_j$ 가

곱해질 것이므로, normalization coefficient는 중요 X

adaptive parameter는 data에 따라 그 값이 조절되는 변수이며,

이러한 변수에 의해 scaling되므로 중요하지 않은 것!

# Linear Basis Function Models

## Possible basis function choices

### 2. Sigmoidal basis function

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right) \quad (3.5)$$

where  $\sigma(a)$ 는 아래와 같이 정의되는 logistic sigmoid function

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \quad (3.6)$$

$\tanh(a) = 2\sigma(a) - 1$ 이므로,  $\tanh$  함수도 사용 가능

$\tanh$  함수와 logistic sigmoid function이 위와 같은 관계를 보이므로,

$\sigma(a)$ 의 선형결합 형태는  $\tanh$ 의 선형결합 형태로도 표현 가능한 것!

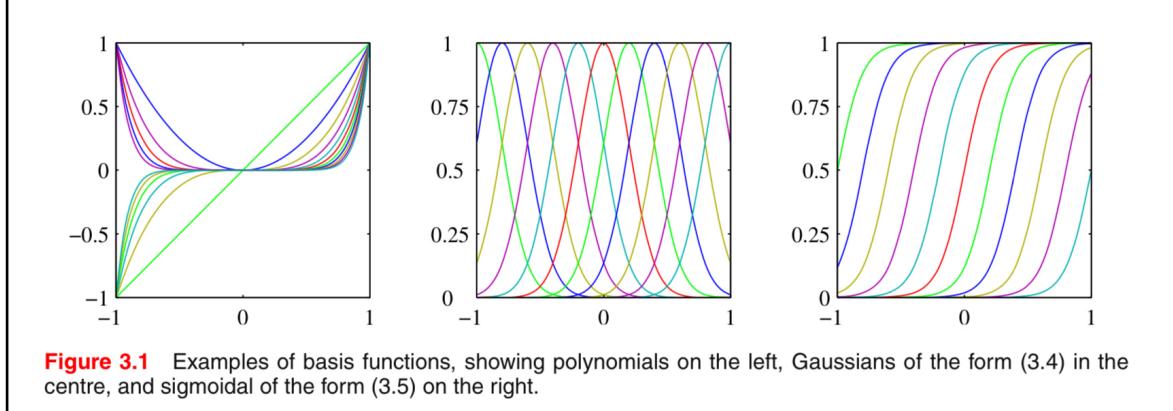
### 3. Fourier basis function

주기성을 가진 함수를 근사하는 데 사용하는 기저 함수

→ 각각의 주파수를 갖는 코사인과 사인 함수로 구성

→ 주기 함수를 푸리에 기저 함수들의 합으로 분해할 수 있음

⇒ 복잡한 신호나 데이터를 분석하고 이해하는데 매우 유용한 도구



# Maximum likelihood and least squares

---

1주차 세션에서 오차 제곱합(sum of squares error / SSE) 함수를 최소화하며 polynomial을 fitting했고, 이러한 접근은 maximum likelihood 해를 구하는 것과 같음을 확인했음

$$N(x | \mu, \sigma^2) \sim \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(x - \mu)^2}{2\sigma^2}\right\}, \beta = \frac{1}{\sigma^2}$$

$$p(\mathbf{t} | \mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^N N(t_n | y(x_n, \mathbf{w}), \beta^{-1}) = \prod_{n=1}^N \frac{1}{(2\pi)^{\frac{1}{2}}} \beta^{\frac{1}{2}} \exp\left\{-\frac{\beta}{2}(y(x_n, \mathbf{w}) - t_n)^2\right\}$$

$$\rightarrow \ln p(\mathbf{t} | \mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2} \sum_{n=1}^N \{t(x_n, \mathbf{w}) - t_n\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

$$\begin{aligned} \text{maximizing } \ln p(\mathbf{t} | \mathbf{x}, \mathbf{w}, \beta) &\Leftrightarrow \text{minimizing } \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 \\ &\Leftrightarrow \text{minimizing the sum of squares error function} \end{aligned}$$

→ 이러한 least squares approach(최소 제곱법)와 maximum likelihood(최대 가능도) 방법 간의 관계를 살펴볼 것!

# Maximum likelihood and least squares

---

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \quad (3.7), \text{ where } \epsilon \sim N(0, \beta^{-1})$$

$$p(t | \mathbf{x}, \mathbf{w}, \beta) = N(t | y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \quad (3.8)$$

새로운 변수  $\mathbf{x}$ 에 대한 최적의 예측값은 target variable의 조건부 평균 (1주차 세션 내용)

$$E[t | \mathbf{x}] = \int t p(t | \mathbf{x}) dt = y(\mathbf{x}, \mathbf{w}) \quad (3.9)$$

이제 input  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 과 그에 따른 target value  $t_1, \dots, t_N$ 로 이루어진 data set을 고려해보자

→ target value  $\{t_n\}$ 을 열벡터  $\mathbf{t}$ 로 묶을 수 있음 (볼드체로 표시하여 multivariate target의 single observation과 구별!)

# Maximum likelihood and least squares

---

likelihood function

$$p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^N N(t_n | y(\mathbf{x}, \mathbf{w}), \beta^{-1}) = \prod_{n=1}^N N(t_n | \mathbf{w}^T \phi(\mathbf{x}_n), \beta^{-1}) \quad (3.10)$$

likelihood function (3.10)에 log를 취하면

$$\begin{aligned} \ln p(\mathbf{t} | \mathbf{w}, \beta) &= \sum_{n=1}^N \ln N(t_n | \mathbf{w}^T \phi(\mathbf{x}_n), \beta^{-1}) \\ &= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(w) \quad (3.11) \end{aligned}$$

$$\text{where sum of squares error function } E_D(w) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 \quad (3.12)$$

⇒ 이제 위 식의  $\mathbf{w}, \beta$ 에 maximum likelihood 방법을 적용해보자!

# Maximum likelihood and least squares

---

$$\ln p(\mathbf{t} | \mathbf{w}, \beta) = \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \frac{\beta}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 \quad (3.11)$$

1)  $\mathbf{w}$ 에 대해 최대화해보자

Gaussian noise 분포 하에서 선형 모델에 대해

likelihood function 을 최대화하는 것은

sum of squares error function을 최소화하는 것과 동일

따라서, (3.11)의 gradient를 구해보면

$$\nabla \ln p(\mathbf{t} | \mathbf{w}, \beta) = \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\} \phi(\mathbf{x}_n)^T \quad (3.13)$$

이 gradient를 0으로 놓으면 아래와 같은 결과를 얻게 된다

$$\begin{aligned} 0 &= \sum_{n=1}^N t_n \phi(\mathbf{x}_n)^T - \mathbf{w}^T (\sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T) \quad (3.14) \\ \Rightarrow \mathbf{w}_{ML} &= (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t} \quad (3.15) \end{aligned}$$

- (3.15)는 normal equations for least squares problem  
(최소 제곱 문제의 정규 방정식)

-  $N \times M$  행렬  $\Phi$ 는 design matrix,  $\Phi_{nj} = \phi_j(\mathbf{x}_n)$

$$\Phi = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}. \quad (3.16)$$

-  $\Phi^+ \equiv (\Phi^T \Phi)^{-1} \Phi^T \quad (3.17)$

= Moore-Penrose pseudo-inverse of matrix  $\Phi$

= 행렬  $\Phi$ 의 무어-펜로즈 유사-역

= 역행렬 개념을 non square 행렬들에 대해 일반화한 것

# Maximum likelihood and least squares

---

$$\ln p(\mathbf{t} | \mathbf{w}, \beta) = \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \frac{\beta}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 \quad (3.11)$$

bias parameter  $w_0$ 에 대해 알아보자

$$E_D(w) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 \quad (3.12)$$

$$= \frac{1}{2} \sum_{n=1}^N \{t_n - w_0 - \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}_n)\}^2 \quad (3.18)$$

$w_0$ 에 대한 미분값을 0으로 두고  $w_0$ 에 대해 식을 풀면

$$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j \bar{\phi}_j \quad (3.19),$$

$$where \bar{t} = \frac{1}{N} \sum_{n=1}^N t_n, \bar{\phi}_j = \frac{1}{N} \sum_{n=1}^N \phi_j(\mathbf{x}_n) \quad (3.20)$$

⇒ bias parameter  $w_0$ 는 training set의 target values의 평균과

기저 함수 값의 평균들의 weighted sum 간의

차이를 보상하는 역할을 한다!

2)  $\beta$ 에 대해 최대화 해보자

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^N \{t_n - \mathbf{w}_{ML}^T \phi(\mathbf{x}_n)\}^2 \quad (3.21)$$

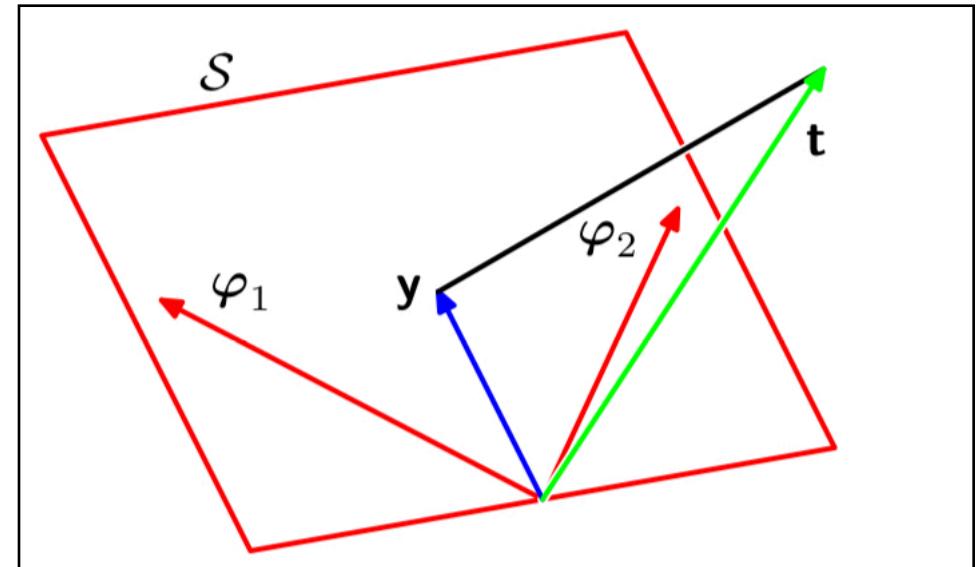
->  $\beta$ 의 역이 잔차의 분산이 된다.

# Geometry of least squares

- $t_1, \dots, t_N$ 를 축으로 가지는  $N$ 차원 공간을 가정  
( 즉,  $\mathbf{t} = (t_1, \dots, t_N)^T$ 는  $N$ 차원 공간의 벡터 )
- 부분공간  $S$ 는 기저 함수  $\phi_j(\mathbf{x})$ 에 의해 그려지는 공간
- 각각의 기저함수는  $\phi_j(\mathbf{x}_n)$ 를 원소로 가지는  $N$ 길이의 벡터  $\varphi_j$

sum of squares error function  $E_D(w)$

$$E_D(w) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 \quad (3.12)$$



→ 이 식을 보면, 오류 함수는  $\mathbf{y}$ 와  $\mathbf{t}$ 간의 거리로 볼 수 있다 (1/2은 무시)

즉,  $w$ 에 대한 최소 제곱해는 위의 그림처럼

$\mathbf{t}$ 를 부분공간  $S$ 에 정사영(orthogonal projection)한 것!

(부분공간  $S$ 의  $\mathbf{y}$ 와  $\mathbf{t}$ 가 가장 가까워야 하므로)

# Sequential learning

---

$$\mathbf{w}_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t} \quad (3.15)$$

같은 batch technique은 한 번에 전체 training set을 처리해야 해서 연산 비용이 높다는 단점이 존재

→ large data set의 경우에는

sequential 알고리즘(on-line 알고리즘)이 유용

→ 한 번에 하나의 data point를 고려하고,  
그 때마다 모델의 parameter를 update

Stochastic Gradient Descent (Sequential Gradient Descent)

- error function<sup>0|</sup> data points( $E = \sum_n E_n$ )의  
error function의 합과 같다면,

SGD를 활용하여 sequential learning을 진행할 수 있음

- 패턴  $n$ 이 등장한 후, parameter vector  $w$ 를 아래와 같이 update

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n \quad (3.22)$$

where  $\tau$ 는 iteration number,  $\eta$ 는 learning rate parameter

- sum of squares error function (SSE)에 대해서는,

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)T} \phi_n) \phi_n \quad (3.23), \text{ where } \phi_n = \phi(\mathbf{x}_n)$$

위의 식은 least means square (LMS) 알고리즘이며,

$\eta$ 값은 알고리즘이 수렴하도록 선택되어야 함

# Regularized least squares

---

Overfitting 문제를 해결하는 방법은 2가지 - training set의 data를 늘리거나 regularization을 활용하는 것

이 중, 우리에게 중요한 것은 regularization!

Overfitting을 막기 위해 error function에 regularization term을 더하면 error function은 아래와 같은 형태를 보임

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w}) \quad (3.24)$$

$\lambda$ 는 data에 종속적인 에러  $E_D(\mathbf{w})$ 와 regularization term(정규화항)  $E_W(\mathbf{w})$ 간의 상대적인 중요도를 조절하는 regularization coefficient(정규화 계수)

가장 단순한 형태의 정규화항은 sum of squares of weight vector elements  $E_W(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} \quad (3.25)$ ,

sum of squares error function  $E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 \quad (3.26)$

$\Rightarrow$  total error function =  $\frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \quad (3.27)$

# Regularized least squares

---

$$\text{total error function} = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \quad (3.27)$$

## 1) machine learning 관점

data에 의해 지지되지 않으면 가중치가 0으로 감소하므로,  
weight decay(가중치 감소)

위와 같은 형태의 정규화항의 장점

= error function이  $w$ 에 대한 이차함수 형태로 나타나  
error function을 최소화하는 값을 closed form으로 구할 수 있음

## weight decay

= overfitting은 weight 변수 값이 커서 발생하는 경우가  
많으므로, 큰 가중치에 큰 penalty를 부과하여  
overfitting을 억제하는 방식

식 (3.27)의 gradient를 0으로 두고,  $w$ 에 대해 풀어보면

$$\mathbf{w} = (\lambda I + \Phi^T \Phi)^{-1} \Phi^T \mathbf{t} \quad (3.28)$$

이는 (3.15) least squares solution이 확장된 형태

$$\mathbf{w}_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t} \quad (3.15)$$

## 2) statistics 관점

parameter value를 0으로 축소하므로,  
parameter shrinkage

# Regularized least squares

더 일반적인 형태의 regularizer를 사용하는 경우

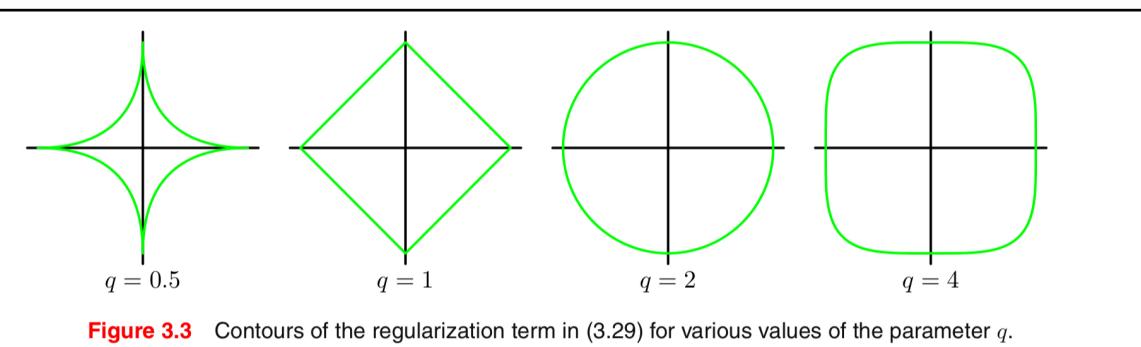
$$\text{total error function} = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \sum_{j=1}^M |w_j|^q \quad (3.29)$$

$q=2$ 일 때의 경우는 이전에 언급한 quadratic regularizer(3.17) /  $q=1$ 일 때의 경우는 lasso

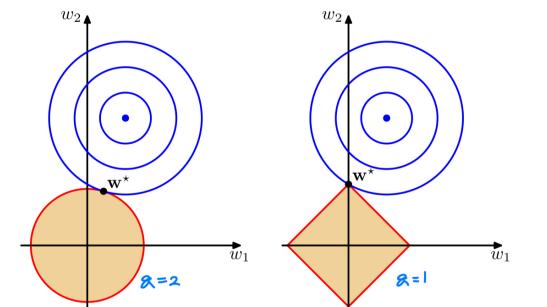
lasso =  $J(\theta) = \text{MSE}(\theta) + \alpha \sum_{i=1}^n |\theta_i| = \text{MSE} + \text{가중치들의 절대값의 합을 최소로 만들어야 한다는 제약}$

→  $\lambda$ 가 커지면 몇몇 계수  $w_j$ 가 0으로 가서

그에 상응하는 basis function이 아무 역할을 하지 못 하는 sparse model을 만드는 성질이 있음



**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  $w$  is denoted by  $w^*$ . The lasso gives a sparse solution in which  $w_1^* = 0$ .



# Regularized least squares

---

Regularization은 모델의 복잡도를 제한하여

복잡한 모델이 한정된 양의 데이터에서 overfitting 없이 만들어질 수 있도록 하는 역할!

이 때 최적의 모델 복잡도를 정하는 것은 적절한 정규화 계수  $\lambda$ 를 찾음으로써 해결

# Multiple outputs

---

지금까지는 single target variable  $t$ 만 다루었지만,  $K(K > 1)$ 개의 target variables을 예측해야 하는 경우도 존재

→ 같은 종류의 기저 함수를 target vector의 모든 성분에 동일하게 사용

$y(\mathbf{x}, \mathbf{w}) = \mathbf{W}^T \phi(\mathbf{x})$  (3.31), where  $y$ 는  $K$ 차원 열벡터,  $\mathbf{W}$ 는  $M \times K$  매개변수 행렬,  $\phi(\mathbf{x})$ 는  $\phi_j(\mathbf{x})$ 를 원소로 가지는  $M$ 차원 열벡터( $\phi_0(\mathbf{x})=1$ )

target vector의 조건부 분포를 isotropic Gaussian 형태로 나타내면, *isotropic Gaussian* = 공분산 행렬이 단위 행렬의 scalar배

$p(\mathbf{t} | \mathbf{x}, \mathbf{W}, \beta) = N(\mathbf{t} | \mathbf{W}^T \phi(\mathbf{x}), \beta^{-1} \mathbf{I})$  (3.32)

→ 모든 방향으로 동일한 분산, 표준 편차 → 공간에서 모든 방향이 동일하게 취급됨

관측값  $t_1, \dots, t_N$ 는  $n$ 번째 행이  $t_n^T$ 인  $N \times K$ 행렬  $T$ 로 묶고, 입력 벡터  $\mathbf{x}_1, \dots, \mathbf{x}_N$ 는  $\mathbf{X}$ 행렬로 묶으면, log likelihood 함수는

$$\ln p(\mathbf{T} | \mathbf{X}, \mathbf{W}, \beta) = \sum_{n=1}^N \ln N(t_n | \mathbf{W}^T \phi(\mathbf{x}_n), \beta^{-1} \mathbf{I}) = \frac{NK}{2} \ln\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2} \sum_{n=1}^N \|t_n - \mathbf{W}^T \phi(\mathbf{x}_n)\|^2 \quad (3.33)$$

이 식을  $\mathbf{W}$ 에 대해 최대화를 하면,  $\mathbf{W}_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{T}$  (3.34)

이 값을 각각의 타겟 변수  $t$ 에 적용하면,  $\mathbf{w}_k = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}_k = \Phi^+ \mathbf{t}_k$  (3.35)

→ regression solution은 서로 다른 타겟 변수들 간에 분리되고, 모든  $w_k$  사이에 공유되는 pseudo-inverse matrix  $\Phi^+$ 만 계산하면 구할 수 있다

2

# Frequentist and Bayesian Perspective on Model Complexity

---

The Bias-Variance Decomposition, Bayesian Linear  
Regression

# The Bias-Variance Decomposition

---

- 복잡한 모델을 근사할 경우 maximum likelihood 방법을 사용하면 overfitting이 발생할 수 있음
  - 이러한 overfitting을 피하기 위해 basis function의 수를 제한하면 트렌드를 잘 반영하지 못한 모델이 될 수 있음
  - overfitting을 피하기 위한 다른 방법으로 regularization term을 사용할 때에는  
 $\lambda$ 의 값을 적절하게 조절하는 것에 어려움이 있음
- model complexity에 대한 빈도주의적 관점에서 Bias - Variance trade off를 확인해보자

# The Bias-Variance Decomposition

---

Expected squared loss(기대 제곱 오류)

$$E[L] = \int [y(\mathbf{x}) - h(\mathbf{x})]^2 p(\mathbf{x}) d\mathbf{x} + \iint [h(\mathbf{x}) - t]^2 p(\mathbf{x}, t) d\mathbf{x} dt \quad (3.37)$$

- $h(\mathbf{x})$ = 최적의 예측치=  $E[t | \mathbf{x}]$
- 첫번째 항은 **reducible error**
- 두번째 항은  $y(\mathbf{x})$ 와 독립적인 항으로 data의 내재적인 노이즈, **irreducible error**
- 결국 첫번째 항을 줄이는 것을 목표로 해야 하는데,  
이를 위해서는  $y(\mathbf{x})$ 를  $h(\mathbf{x})$ 에 가깝게 fitting 해야 한다
- $N$ 이 무한이라면  $h(\mathbf{x})$ 를 찾아낼 수 있지만, data set  $D$ 는  
유한한  $N$ 개의 data points를 가지므로 정확하게 찾아낼 수 없다

빈도주의적 관점에서 불확실성 해석

- $p(t, \mathbf{x})$ 로 추출한 data set  $D$ 에서  
예측 함수  $y(\mathbf{x}; D)$ 와 그의 제곱 오류값들을 구한다.
- 이를 평균낸 것으로 알고리즘의 성능을 파악한다
- 이는 대수의 법칙에 의해 기대제곱오류로 근사할 것이다

# The Bias-Variance Decomposition

---

식 (3.37)의 첫번째 항의 피적분 함수

$$\{y(\mathbf{x}; D) - h(\mathbf{x})\}^2 \quad (3.38)$$

$$\begin{aligned} &= \{y(\mathbf{x}; D) - E_D[y(\mathbf{x}; D)] + E_D[y(\mathbf{x}; D)] - h(\mathbf{x})\}^2 \\ &= \{y(\mathbf{x}; D) - E_D[y(\mathbf{x}; D)]\}^2 + \{E_D[y(\mathbf{x}; D)] - h(\mathbf{x})\}^2 \\ &+ 2\{y(\mathbf{x}; D) - E_D[y(\mathbf{x}; D)]\}\{E_D[y(\mathbf{x}; D)] - h(\mathbf{x})\} \quad (3.39) \\ &\rightarrow D \text{에 대해 이 식의 기댓값을 구하고 마지막 항을 정리하면} \end{aligned}$$

$$\begin{aligned} &E[[y(\mathbf{x}) - h(\mathbf{x})]^2] \\ &= [E[y(\mathbf{x}|D) - h(\mathbf{x})]]^2 + E[\{y(\mathbf{x}|D) - E[y(\mathbf{x}|D)]\}^2] \quad (3.40) \end{aligned}$$

- 첫번째 항은  $(bias)^2$ , 두번째 항은 variance

- Bias: 전체 데이터 집합들에 대한 평균 예측이

회귀 함수와 얼마나 차이나는지 (편향)

- Variance: 각각의 데이터 집합에서의 해가

전체 평균에서 얼마나 차이나는지 (분산)

기대 오류 분해

$$\text{기대 오류} = (Bias)^2 + Var + Noise$$

$$(Bias)^2 = \int \{E_D[y(\mathbf{x}; D)] - h(\mathbf{x})\}^2 p(\mathbf{x}) d\mathbf{x} \quad (3.42)$$

$$Var = \int E_D[\{y(\mathbf{x}; D) - E_D[y(\mathbf{x}; D)]\}^2] p(\mathbf{x}) d\mathbf{x} \quad (3.43)$$

$$Noise = \int \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) d\mathbf{x} dt \quad (3.44)$$

# The Bias-Variance Decomposition

---

## Bias-Variance Tradeoff

- Flexibility가 높은 모델 (복잡한 모델) : 모델의 편향은 작고 모델의 분산은 크다

ex) 비모수 모델, 복잡한 regression

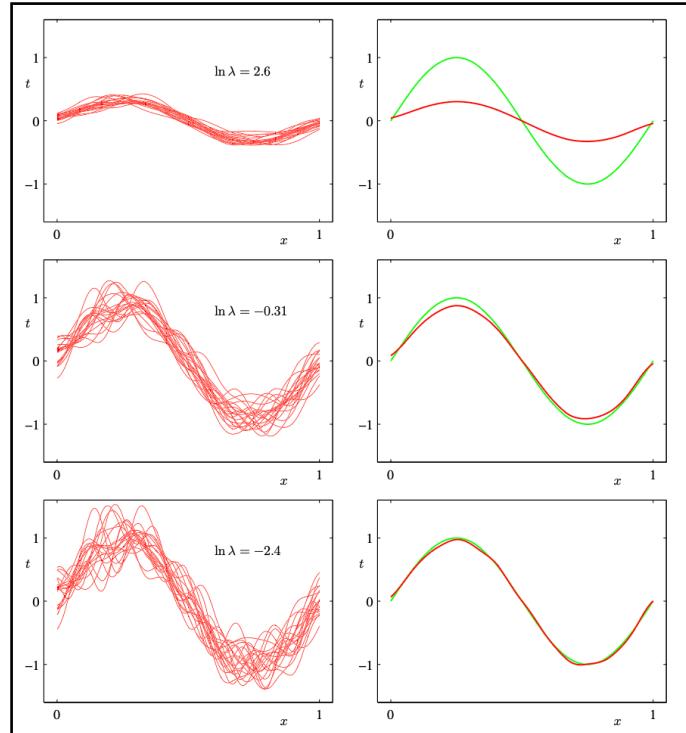
- Flexibility가 낮은 모델 (단순한 모델) : 모델의 편향은 크고 모델의 분산은 작다

ex) 단순 regression

→ 편향과 분산 사이의 가장 좋은 밸런스를 가지는 모델이 최적의 예측치를 내는 모델

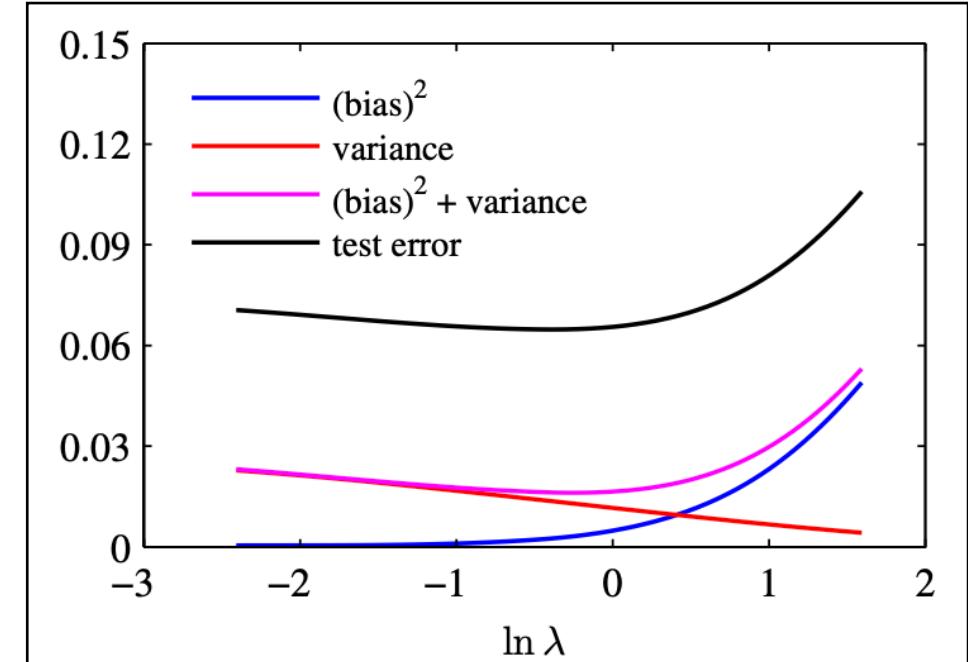
# The Bias-Variance Decomposition

$h(x) = \sin(2\pi x)$ 로부터  $N=25$ 인 100개의 데이터 집합을 생성하고 이를 통해 추정해보자



- regularization 계수  $\lambda$ 가 큰 1행에서는 높은 bias와 낮은 var를 보인다
- $\lambda$ 가 작은 3행에서는 낮은 bias와 높은 var를 보인다
- 이렇듯, regularization 계수  $\lambda$ 를 적절하게 사용해야 한다!

regularization 계수  $\lambda$ 에 따라 bias와 var이 어떻게 변해가는지 보자



- $\lambda$ 가 커짐에 따라 bias는 커지고 var은 감소하는 것을 볼 수 있다
- reducible error는 test error와 비례하여 커지고 있다
- 따라서, 적절하게 regularization 계수  $\lambda$ 를 설정해주어야 한다!

# The Bias-Variance Decomposition

---

Bias-Variance Decomposition은 여러 데이터 집합들의 모임에 대한 평균을 바탕으로 하므로,

모델 복잡도에 대한 빈도주의적 관점의 실제적인 가치는 제한적

현실 속 사례에서는 한 개의 observed data set만 주어지는 것이 일반적이므로!

⇒ overfitting과 모델 복잡도에 관련된 실용적인 technique을 제공하는 Bayesian 관점에 대해 살펴보자!

# Bayesian Linear Regression

---

## Maximum Likelihood 방법

- 모델의 복잡도: 기저 함수의 수로 결정되고, 데이터 집합 크기와 모델의 복잡도에 의해 조절된다

## Maximum Likelihood 방법의 문제

- overfitting하는 지나치게 복잡한 모델을 선택하게 된다
- training data와는 독립적인 검증 데이터 집합을 이용하는 Cross Validation 기법은 계산이 복잡하고, 소중한 data를 낭비하는 행위이다

⇒ 이에 대한 해결책으로 제시되는 것이 Bayesian 방법

- Maximum likelihood 방법에서 발생하는 overfitting 문제를 피할 수 있다
- Training data만으로 모델의 복잡도를 자동으로 결정할 수 있다

# Parameter distribution

## 1. Prior distribution

- noise precision parameter  $\beta$ 는 known constant라고 가정
- likelihood function  $p(\mathbf{t} | w)$ 는  $w$ 의 이차 함수의 지수 함수

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

- 이에 해당하는 conjugate prior는 Gaussian 분포를 사용

$$p(\mathbf{w}) = \mathbf{N}(\mathbf{w} | \mathbf{m}_0, \mathbf{S}_0) \quad (3.49),$$

where  $m_0$ 는 평균,  $S_0$ 는 공분산

Likelihood	Conjugate Prior Density	Posterior Density
Binomial	Beta	Beta
Negative binomial	Beta	Beta
Poisson	Gamma	Gamma
Normal, with unknown mean	Normal	Normal
Normal, with unknown variance	Inverse gamma	Inverse gamma
Normal, with unknown mean and variance	Normal-inverse gamma	Normal-inverse gamma

## 2. Posterior distribution

- Conjugate prior를 Gaussian으로 선택했으므로, Posterior도 Gaussian 분포

$$p(\mathbf{w} | \mathbf{t}) = \mathbf{N}(\mathbf{w} | \mathbf{m}_N, \mathbf{S}_N) \quad (3.49),$$

$$\text{where } \mathbf{m}_N = \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\mathbf{\Phi}^T\mathbf{t}) \quad (3.50)$$

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\mathbf{\Phi}^T\mathbf{\Phi} \quad (3.51)$$

- Posterior distribution이 Gaussian 분포이기에 최빈값과 평균값이 일치한다 ( $w_{MAP} = m_N$ )

$$- \mathbf{S}_0 = \alpha^{-1}\mathbf{I} \quad (\alpha \rightarrow 0)$$

즉, 무한대로 넓은 prior인 경우,  $w_{ML} = m_N$

- $N = 0$ 인 경우, posterior = prior

- data points가 순차적으로 입력될 경우,

각 단계에서의 posterior가 다음 단계의 prior에 해당

# Parameter distribution

---

## 3. Bayesian Linear Regression 예시

<단순화를 위한 가정>

- 예시를 보기에 앞서 단순화하기 위하여 특정한 가우시안 사전분포로 가정

[사전분포]

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

[사후분포]

$$p(\mathbf{w} | \mathbf{t}) = N(\mathbf{w} | \mathbf{m}_N, \mathbf{S}_N)$$

$$\text{where } \mathbf{m}_N = \beta \mathbf{S}_N \boldsymbol{\phi}^T \mathbf{t} \quad \mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \boldsymbol{\phi}^T \boldsymbol{\phi}$$

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

이때 사후 분포를 최대화시키는  $\mathbf{w}$

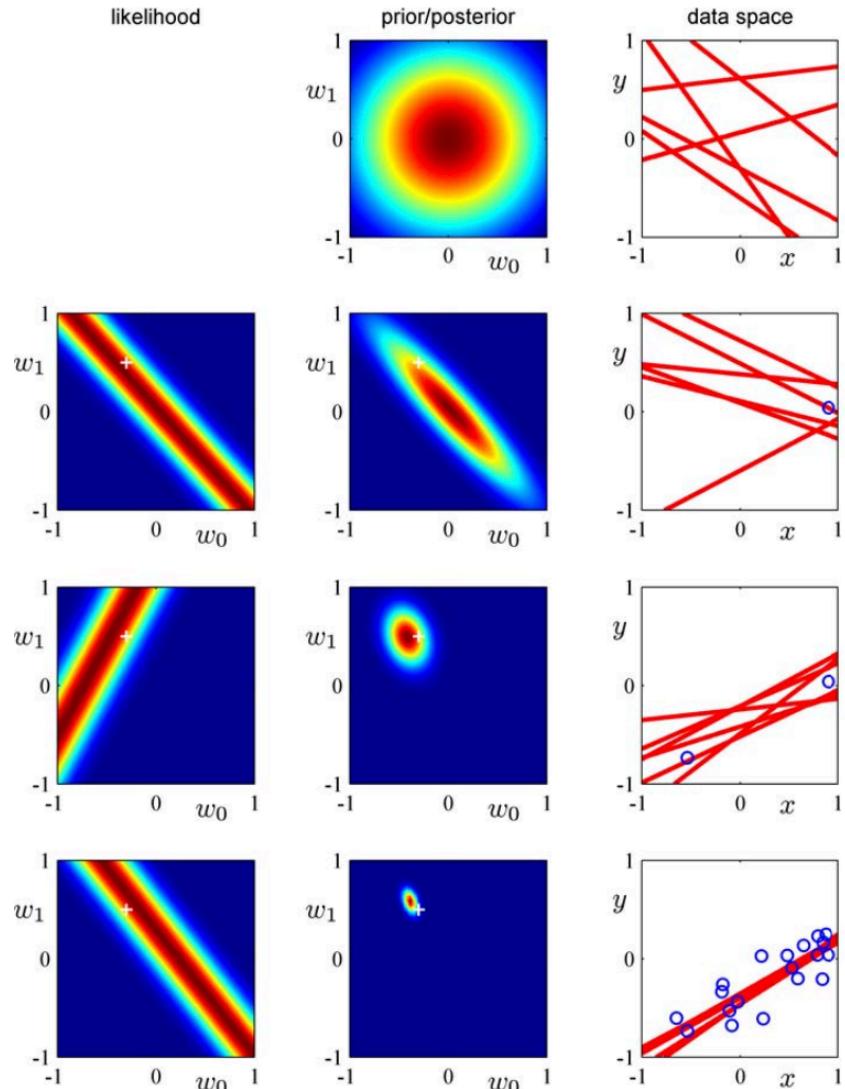
= 제곱 정규화항을 포함한 제곱합 오류함수 극소화시키는  $\mathbf{w}$

<target variable  $t_n$  생성>

- 첫번째,  $f(x, \mathbf{a}) = a_0 + a_1 x$ 에서  $a_0 = -0.3$ ,  $a_1 = 0.5$ 을 가정한다.
- 두번째,  $x_n$ 을  $U(0,1)$ 에서 난수를 생성한다.
- 세번째,  $f(x, \mathbf{a})$ 에 넣고 계산한 뒤,  
표준편차 0.2 가우시안 노이즈를 추가해  $t_n$ 을 만들어낸다.
- 이렇게 생성된 데이터를 통해서 우리는  $a_0$ 와  $a_1$ 을 추정해보겠다.

여기서  $\beta$ 는 25로 Known constant이다.

# Parameter distribution



그래프의 간략한 설명을 하자면

1행은  $N = 0$ , 2행은  $N = 1$ , 3행은  $N = 3$ , 4행은  $N = 20$ 이다.

1열은 Likelihood function, 2열은 posterior distribution, 3열은 data space 이다.

1열과 2열에서 + 점은 앞서 가정한 실제 모수이다. 또한 그래프는 등고선의 형태로 주어졌다.

3열은 파란색 원은 data point, 빨간색은 추정된 회귀직선들이다.

1행) 사후분포를 사전분포를 사용

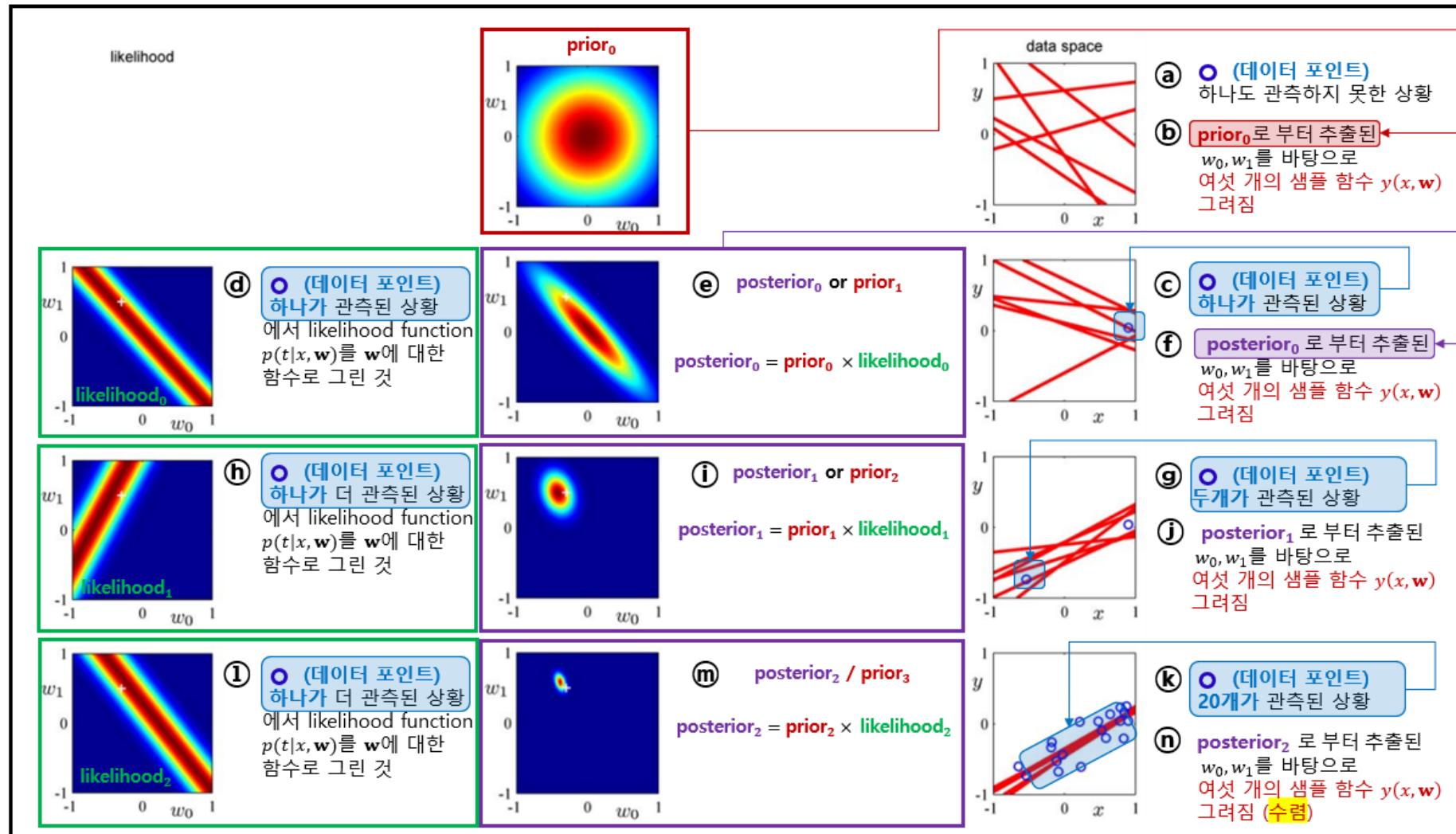
2행) 1행의 사후분포를 사전분포로 사용하여 사후분포를 도출

3행) 2행의 사후분포를 사전분포로 사용하여 사후분포를 도출

데이터가 추가될 때마다 순차적으로 사후분포를 업데이트한다.

이때 사후분포가 실제 parameter에 가까워짐을 볼 수 있다.

# Parameter distribution



# Predictive Distribution

- Predictive Distribution은 다음과 같이 정의된다.

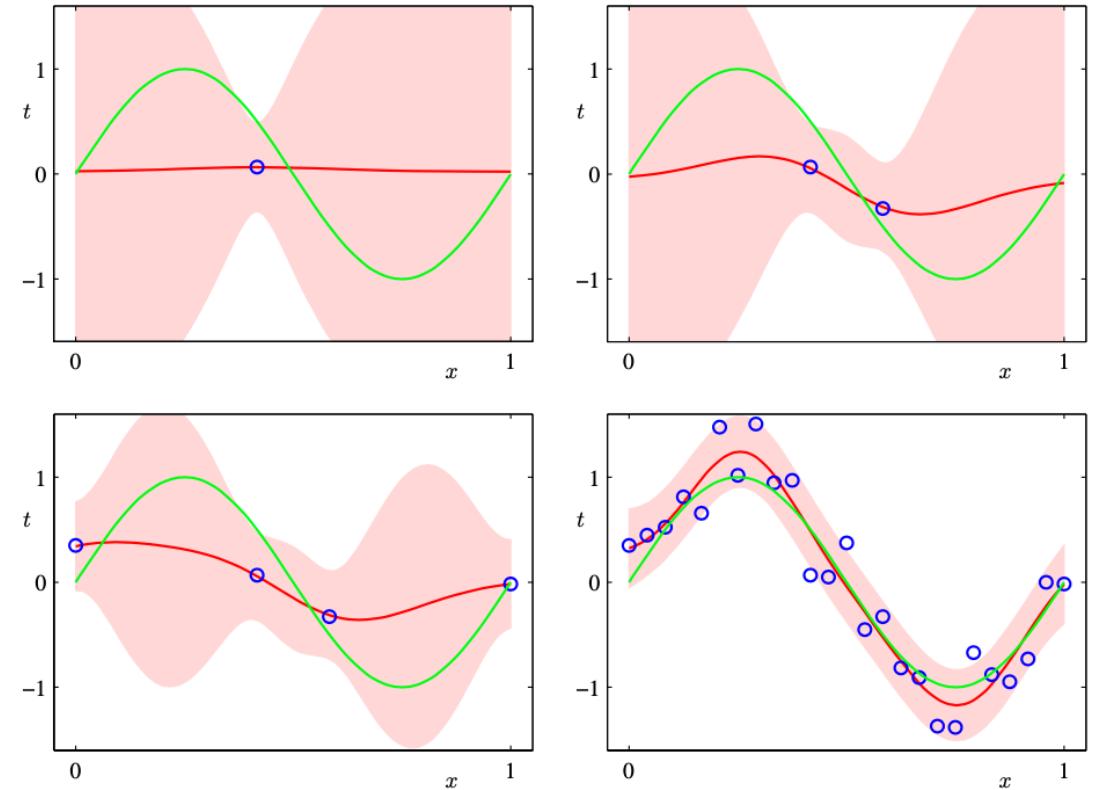
$$\begin{aligned} p(t | \mathbf{t}, \alpha, \beta) &= \int p(t | \mathbf{w}, \beta) p(\mathbf{w} | \mathbf{t}, \alpha, \beta) d\mathbf{w} \\ &= \mathcal{N}(t | \mathbf{m}_N^T \phi(\mathbf{x}), \sigma_N^2(\mathbf{x})) \end{aligned}$$

- 이때 예측 분포의 분산은 다음과 같다.

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

- 첫번째 항은 데이터의 noise를 표현하며 두번째 항은 매개변수  $w$ 에 대한 불확실성을 표현한다. 추가적인 데이터가 관측될수록 사후 분포는 좀 아지게 되며(narrower),  $N$ 이 무한대로 갈 경우, 두 번째 항이 0이 되면서 예측 분포의 분산은 데이터의 noise만을 포함하게 된다.

$$\lim_{N \rightarrow \infty} \sigma_N^2(\mathbf{x}) = \frac{1}{\beta}$$



- $N = 1, 2, 4, 25$
- $N$ 이 증가할수록 level of uncertainty가 감소한다는 점, data points 근처에서 uncertainty가 감소한다는 점을 확인할 수 있다.

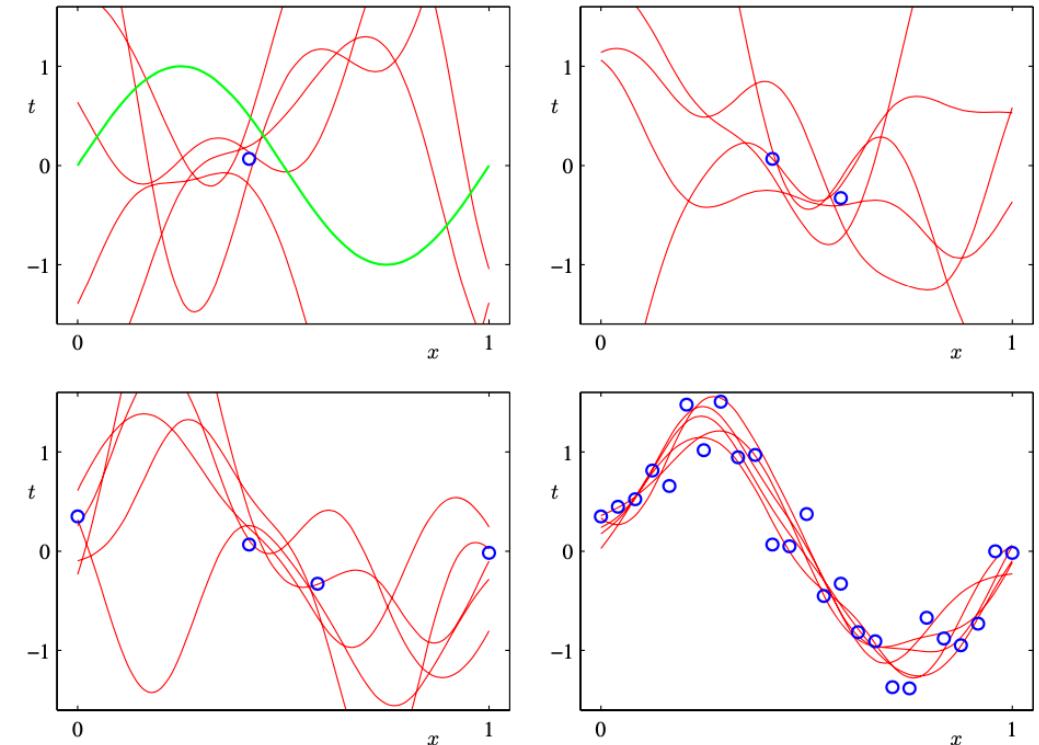
# Predictive Distribution

---

서로 다른  $x$ 의 예측값들에 대한 공분산을 살펴보기 위해  $w$ 에 대한 사후 분포로

부터 샘플들을 추출한 그 이에 대한 함수  $y(x, \mathbf{w})$ 를 그려볼 수 있다.

- $N$ 이 적은 경우 가능한 분산 범위가 크기 때문에 variance가 큰 결과를 얻게 되며,  $N$ 이 증가할수록 안정된 범위의 모델을 얻을 수 있다는 것을 확인할 수 있다.
- Gaussian과 같이 국소화된 기저 함수(localized basis function)을 사용하면 예측 분포의 분산의 두번째 항( $\phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x})$ )이 0으로 수렴하여  $\beta^{-1}$ 만 남게 된다. 이때, 해당 모델의 예측에 대한 신뢰는 높아지지만 이는 바람직한 것은 아니기에 Gaussian process를 적용함으로써 해결할 수 있다.
- 만약  $\mathbf{w}, \beta$ 를 모르는 상황이라면 prior는 Gaussian-gamma가 되고 posterior는 Student's t-distribution이 된다.



# Equivalent Kernel

---

Predictive distribution  $p(t | \mathbf{t}, \alpha, \beta) = \mathcal{N}(t | \mathbf{m}_N^T \phi(\mathbf{x}), \sigma_N^2(\mathbf{x}))$ 에서, predictive mean은 다음과 같다.

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

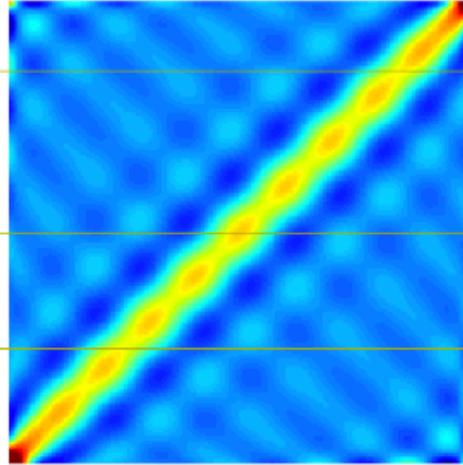
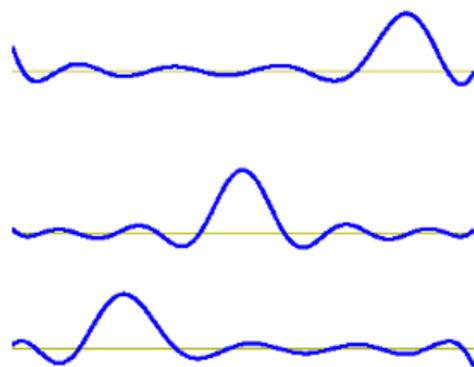
여기서  $k(\mathbf{x}, \mathbf{x}')$ 을 equivalent kernel 혹은 smoother matrix라 하고, 다음과 같이 표현할 수 있다.

$$k(\mathbf{x}, \mathbf{x}') = \beta \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}')$$

- training set의 target값들의 선형 결합을 입력 받아서 예측을 하는 regression function은 linear smoother라고 부른다.
- $\mathbf{S}_N$ 의 정의에  $\mathbf{x}_N$ 이 포함되어 있기 때문에 equivalent kernel은  $\mathbf{x}_N$ 에 종속적이다.

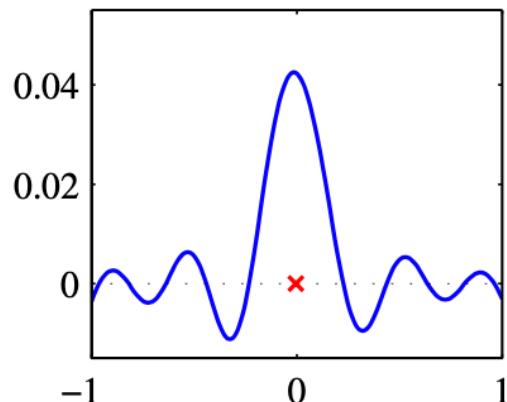
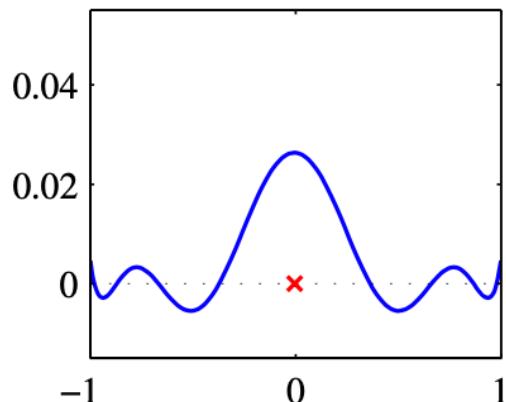
# Equivalent Kernel

---



## Gaussian 기저 함수에 대한 동등 커널 $k(x, x')$

- $x$ 와  $x'$ 이 가까이 있을 때 높은 값을, 멀리 있을 때 작은 값을 가진다는 것을 확인할 수 있다. (붉을수록 높은 값을 의미)
- 왼쪽 그래프를 통해  $x$ 와  $x'$ 이 가까울 때는 가중치를 높이고 멀리 있을 때는 가중치를 낮추는 것을 알 수 있다. 즉,  $x$ 와  $x'$ 가 가까울수록, local evidence를 높게 가중할 수 있다. 결국, kernel은 두 data points 사이 유사성의 정도를 측정하고, 참 값에 가까울수록 가중치를 높임으로써 estimating process가 관찰된 target values의 가중 평균임을 나타낸다.



Gaussian 기저함수일 때 뿐만 아니라, polynomial(왼쪽), sigmoidal(오른쪽) 와 같이 nonlocal한 기저함수에 대해서도 localized 함수로 그려진다.

# Equivalent Kernel

---

$y(\mathbf{x})$ 과  $y(\mathbf{x}')$ 의 공분산

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

- 서로 근처에 있는 포인트들의 예측 평균들은 상관성이 크며, 서로 멀리 떨어져 있는 포인트들의 예측 평균들은 상관성이 작다는 것을 볼 수 있다.
- Equivalent kernel은 비선형 함수의 벡터의 내적으로 나타낼 수 있다는 일반적인 커널 함수의 특징 또한 가진다.

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

- 이때,  $\psi(\mathbf{z})$ 는 다음과 같다.

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

- Equivalent kernel은 가중치들을 결정하고, 이 가중치들을 바탕으로 training set의 target 변수들이 합쳐져서 새로운  $\mathbf{x}$ 값에 대한 예측을 진행한다. 즉, equivalent kernel은 학습 데이터와 예측할 새로운 데이터  $\mathbf{x}$ 의 결합으로 이루어져 있다. 이 가중치들을 모든  $\mathbf{x}$  값들에 대해 합산하면 1이 된다.

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

- $t_n = 1$ 의 값을 가지는 모든 데이터  $n$ 을 이용하여 평균  $\hat{y}(\mathbf{x})$ 을 구하면 쉽게 확인할 수 있다.
- 기저 함수들의 선형 독립성이 보장될 때, 즉 기저 함수의 개수보다 샘플의 개수가 더 많고, 적어도 하나의 기저 함수는 상수값일 때, predictive mean  $\hat{y}(\mathbf{x}) = 1$ 이 된다.

3

# Bayesian Model Comparison

---

Bayesian Model Selection, Bayesian Model Comparison

# Bayesian Model Selection

---

베이지안 관점에서는 모델 선택의 문제에서 불확실성을 나타내기 위해 확률을 사용한다.

- 선택 가능한 모델의 수를  $L$ 이라 가정하자.  $\{M_i\}$  ( $i = 1, \dots, L$ )
- training set  $D$ 가 주어졌을 때, posterior distribution은 다음과 같이 정의할 수 있다:

$$p(M_i | D) \propto p(M_i)p(D | M_i)$$

- 여기서  $p(D | M_i)$  을 model evidence 혹은 marginal likelihood라고 한다.

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

- 모델에 대한 posterior distribution을 알게내면 predictive distribution을 다음과 같이 표현할 수 있다:

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

- 위와 같은 방식은 여러 모델을 이용하여 mixture 분포를 만들어내는 형태이다. 이때 예측 분포가 계산이 복잡해지고 다루기 어려워지기 때문에 가장 적합한 모델 하나를 선택하여 근사를 진행한다. 이 과정을 model selection이라고 한다.

$$M^* = \arg \max_{M_i} p(M_i | D) = \arg \max_{M_i} p(D | M_i)p(M_i)$$

- prior이 flat하다고 가정할 때, model evidence를 극대화 하는 모델을 가장 적합한 모델로 선택할 수 있다.

$$M^* = \arg \max_{M_i} p(D | M_i)$$

# Bayesian Model Selection

---

## Bayes Factor (두 모델의 비교)

- Bayes factor는 두 모델의 model evidence의 비로 정의된다.

$$\frac{p(D | M_i)}{p(D | M_j)}$$

- 이때 Bayes factor가 1보다 크다면  $M_i$  이  $M_j$  보다 데이터에 더 잘 들어 맞는다고 할 수 있다. 1보다 작을 경우에는 반대로  $M_j$ 를  $M_i$ 보다 더 선호하게 된다. 따라서 Bayes factor는 하나의 모델이 다른 하나의 모델보다 선호될 비율을 의미한다.

$$\frac{p(M_1 | D)}{p(M_2 | D)} = \frac{p(M_1)p(D | M_1)}{p(M_2)p(D | M_2)} = \frac{p(D | M_1)}{p(D | M_2)}$$

## Bayes Factor와 Kullback-Leibler Divergence (KL 발산)

- 베이지안 관점에서는 암묵적으로 고려되는 모델 중에 실제 모델 분포가 존재하고 여기로부터 데이터가 생성되었다고 가정한다. 이 가정이 맞다면 베이지안 관점에서의 모델 비교는 항상 올바른 모델을 선호하게 된다. 이를 알아보기 위해 두 모델  $M_1, M_2$ 를 설정하고, 이 중 실제 모델은  $M_1$ 일 경우를 가정해보자.
- 이때 Bayes facotr의 평균 값은 다음과 같은 식으로 표현할 수 있다.

$$\int p(D | M_1) \ln \frac{p(D | M_1)}{p(D | M_2)} dD$$

- 위 식은 KL divergence의 예시이고, 두 모델이 서로 다르다는 가정 하에 항상 0이 아닌 양수이다. 따라서 Bayes factor는 항상 올바른 모델을 선호한다는 것을 확인할 수 있다.

# Model with a Single Parameter

1차원의 파라미터  $w$ 만을 갖는 모델을 가정해보자.

- Model Evidence (Marginal Likelihood)는 다음 식과 같이 정의할 수 있다.

$$p(D) = \int p(D | w)p(w)dw$$

- 위 적분식을 근사하기 위해 두 가지 상황을 가정한다:

- 1) Posterior  $p(w | D)$ 가  $w_{MAP}$ 근처에서  $\Delta w_{posterior}$ 의 너비를 가지며 최고점을 형성한다(sharply peaked).

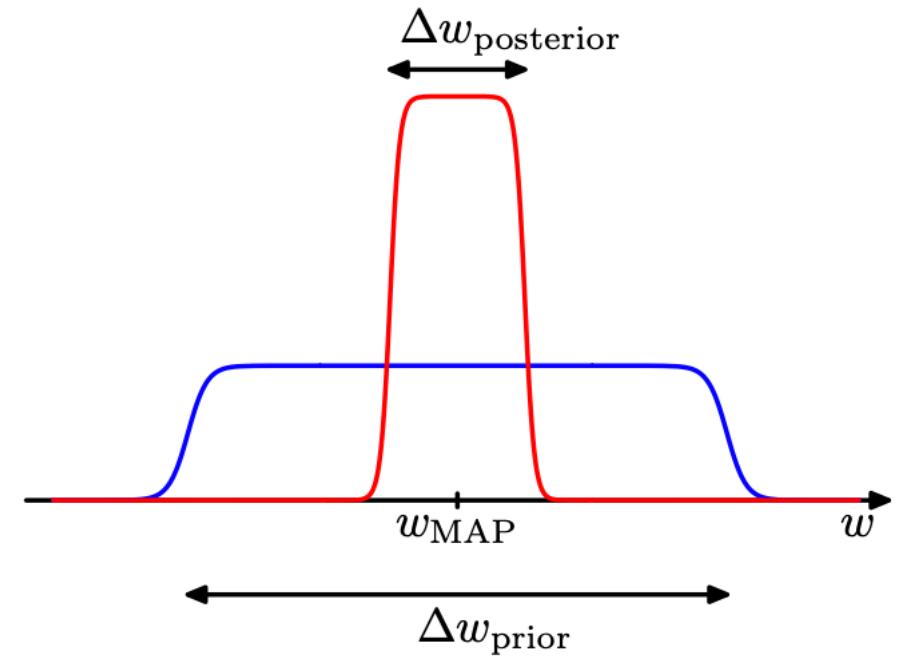
- 2)  $w$ 의 사전 분포는 모두 동일하며,  $p(w) = \frac{1}{\Delta w_{prior}}$

- 이 경우 사후 분포는 다음과 같이 근사된다:

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

- 이 식을 극대화하기 위해 로그를 취해 사용한다:

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



# Model with a Single Parameter

---

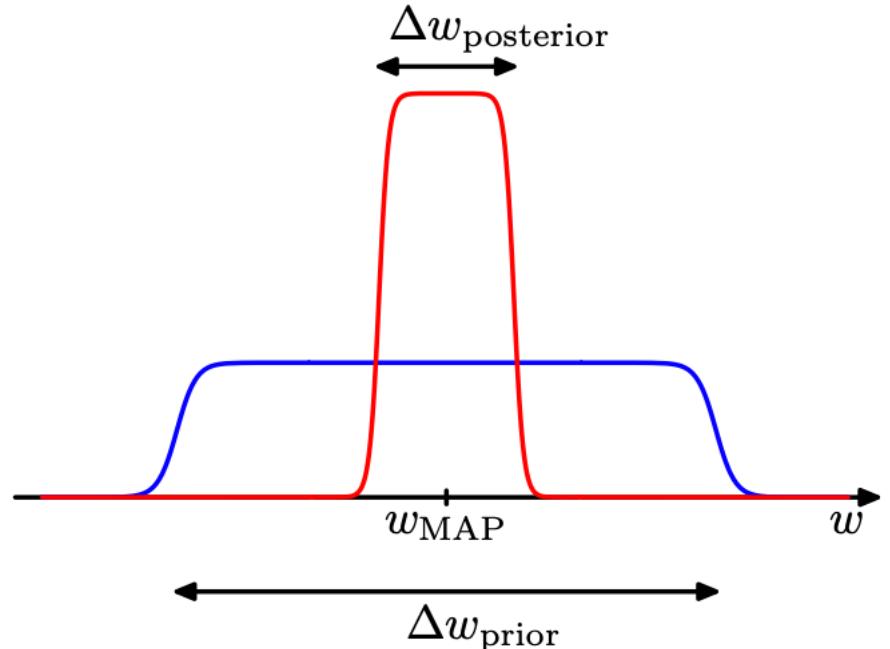
$$\ln p(D) \simeq \ln p(D | w_{MAP}) + \ln\left(\frac{\Delta w_{posterior}}{\Delta w_{prior}}\right)$$

**첫 번째 항:** 데이터를 가장 잘 표현하는 파라미터에 대한 값 (fit to the data)

- flat한 사전 분포의 경우, log likelihood에 해당한다.

**두번째 항:** 모델의 복잡도에서 기인하는 penalty term

- $\Delta w_{posterior} < \Delta w_{prior}$ 의 경우 두 번째 식은 음수가 된다.
- $\frac{\Delta w_{posterior}}{\Delta w_{prior}}$ 의 비율이 작아지면 두 번째 식은 그 크기(절댓값)가 증가한다.
- 매개변수가 사후 분포의 데이터에 잘 맞춰진다면 증가한다.



# Model with a Set of Parameters

---

모델이 M개의 매개변수 집합을 갖는 경우를 생각해보자.

- 모든 매개변수가 같은 비율의  $\frac{\Delta w_{posterior}}{\Delta w_{prior}}$  를 갖는다고 가정하면,

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

- M에 대한 모델의 복잡도가 선형적으로 증가함을 확인할 수 있다.

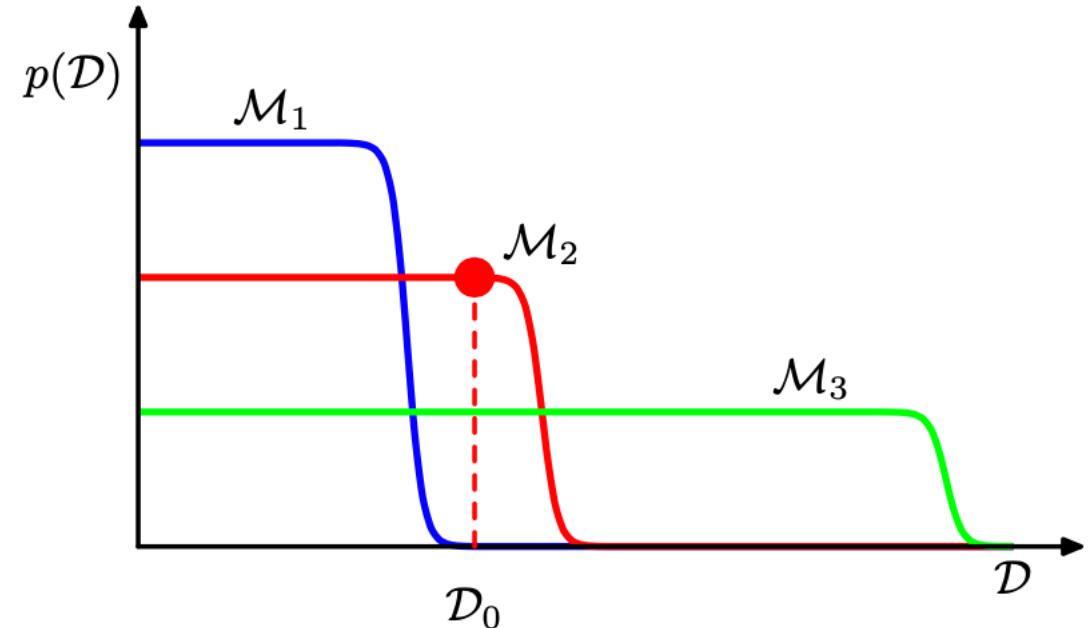
- 일반적으로 복잡한 모델을 사용하는 경우 파라미터의 수가 증가한다. 이 경우 보통 모델 fitting이 더 잘되기 때문에 첫 번째 식의 크기는 증가하게 되지만 M의 증가에 따라 penalty term의 크기도 증가하게 된다.
- 간단한 모델은 데이터에 제대로 fit할 수는 없지만 penalty term 값이 감소하는 것에 비해 복잡한 모델은 fit은 더 잘 되지만 penalty term의 크기가 증가한다.
- 결국, 그 사이의 적절한 복잡도를 가진 모델을 선호하게 된다.

# Example: Bayesian Model Comparison

$M_1, M_2, M_3$  3개의 모델이 주어졌다고 가정해보자.

- 복잡도:  $M_1 < M_2 < M_3$
- 특정 모델로부터 특정 데이터 집합 생성
- 1) 사전 분포  $p(w)$ 로부터 매개변수 값을 선택
- 2) 매개변수를 바탕으로 하는  $p(D | w)$ 로부터 추출
- 너무 단순한 모델은 데이터에 잘 근사하지 못하고, 너무 복잡한 모델은 예측 확률을 너무 넓은 데이터 집합들에 대해 퍼뜨려 각각의 데이터 집합들에 대해 작은 확률을 할당하게 된다.
  - $M_1$  (단순한 모델): 해당 데이터 집합을 근사하지 못함.
  - $M_3$  (복잡한 모델):  $M_2$ 에 비해 낮은 evidence 값을 가짐.

- $p(D | M_i)$ 는 정규화되어 있기 때문에 ( $\int p(D | M_i) dD = 1$ ) 특정 데이터 집합  $D_0$ 는 중간 정도의 복잡도( $M_2$ )를 가진 모델에 대해 가장 큰 evidence를 가지게 된다.



# 4

# Evidence Approximation

---

Evidence Function, Effective Number of Parameters

# The Evidence Approximation

---

- fully 베이지안 관점을 바탕으로 한 선형 기저 함수 모델에서는 사전 분포에 noise  $\beta$  와 parameter variance  $\alpha$  와 같은 초매개변수(hyper parameter)를 도입하여 추론을 진행한다.
- 초매개변수,  $\mathbf{w}$ 에 대한 marginalization은 수학적으로 구하거나 다루기 어렵기 때문에 먼저  $\mathbf{w}$ 에 대해 marginalization을 함으로써 얻어낸 marginal likelihood function을 극대화하는 초매개변수 값을 구하고, 이들을 해당 값으로 고정하는 방법을 선택한다.
- Predictive Distribution은 다음과 같다:  
$$p(t|\mathbf{t}) = \int \int \int p(t|\mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \alpha, \beta) p(\alpha, \beta|\mathbf{t}) d\mathbf{w} d\alpha d\beta$$
- 사후 분포  $p(\alpha, \beta|\mathbf{t})$ 가  $\hat{\alpha}, \hat{\beta}$  주위에 몰려있다면,  $\alpha = \hat{\alpha}, \beta = \hat{\beta}$ 으로 고정하여 다음과 같은 식으로 근사할 수 있다.  
$$p(t|\mathbf{t}) \simeq p(t|\mathbf{t}, \hat{\alpha}, \hat{\beta}) = \int p(t|\mathbf{w}, \hat{\beta}) p(\mathbf{w}|\mathbf{t}, \hat{\alpha}, \hat{\beta}) d\mathbf{w}$$
- 베이즈 정리로부터 얻은 비례식을 이용하여, 사전 분포가 상대적으로 flat하다고 가정하면  $\hat{\alpha}, \hat{\beta}$  은 marginal likelihood function 을 극대화함으로써 얻을 수 있다.
- log evidence 극대화의 두 가지 방법:
  - Evidence function을 미분하고 그 도함수=0 으로 설정하여  $\alpha, \beta$ 에 대한 재추정식 (re-estimation equation)을 구하는 방법
  - Expectation Maximization (EM) 알고리즘을 이용하는 방법

# Evaluation of the Evidence Function

---

- Marginal Likelihood Function은  $\mathbf{w}$ 에 대한 marginalization을 통해 얻을 수 있다.

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

- 정규화된 Gaussian 계수를 가지는 표준 형태로 만들어 식을 정리하면 다음과 같다:

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

- 우측에서 구한 식들을 사용하여 정리하고, 로그를 취하면 다음과 같은 log marginal likelihood function을 구할 수 있다:

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

\* 식 구하는 과정 (참고)

$\mathbf{M}$ 은  $\mathbf{w}$ 의 차원을 나타낸다고 할 때,

$$\begin{aligned} E(\mathbf{w}) &= \beta E_D(\mathbf{w}) + \alpha E_D(\mathbf{w}) \\ &= \frac{\beta}{2} \|\mathbf{t} - \Phi\mathbf{w}\|^2 + \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} \\ &= E_D(\mathbf{m}_N) + \frac{1}{2} (\mathbf{w} - \mathbf{m}_N)^T \mathbf{A} (\mathbf{w} - \mathbf{m}_N) \end{aligned}$$

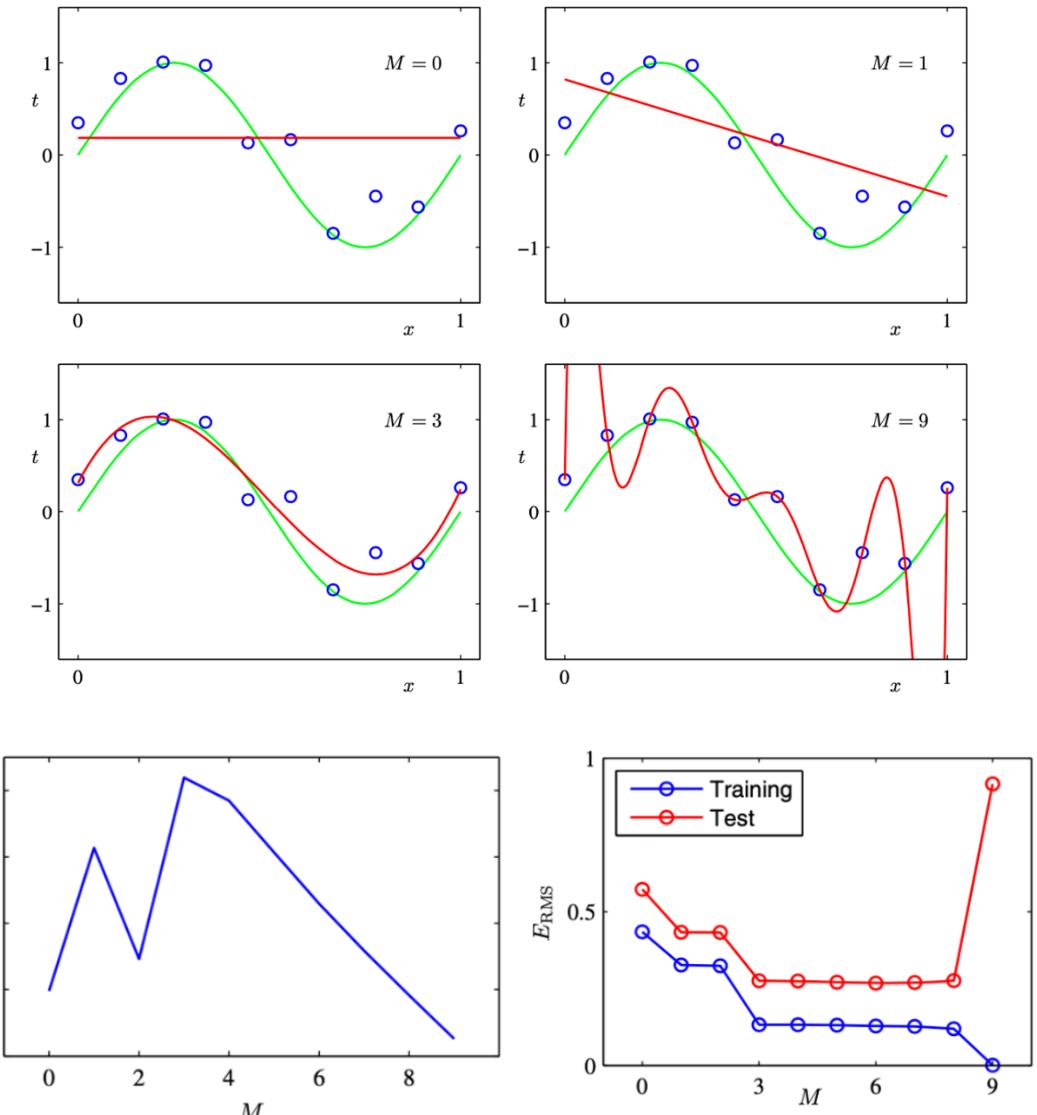
$\mathbf{A} = \alpha \mathbf{I} + \beta \Phi^T \Phi = \mathbf{S}_N^{-1}$  (Hessian matrix),  $\mathbf{m}_N = \beta \mathbf{A}^{-1} \Phi^T \mathbf{t}$  일 때,

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

# Example: Polynomial Regression Problem

앞서 Model Evidence는 likelihood term과 penalty term으로 구성되어 있음을 확인하였다.

- 가장 적절한  $M$ 을 선택하는 과정을 생각해보자.
  - $M=0, M=1$ : 예측이 제대로 되고 있지 않음을 확인할 수 있다.
  - $M=2$ : 기함수 형태의 실제값과 달리 우함수 형태의 예측값을 사용함으로써 model evidence가 감소한다.
  - $M=3$ 에서 그래프상 fit이 잘 되었고 model evidence가 가장 높은 것을 확인 가능하다.
  - $M=3$  이후로는 data fit에서의 적은 증가에 비해 complexity penalty 가 커지면서 결과적으로는 evidence values가 낮아지는 것을 확인할 수 있다.
- 따라서, model evidence가 가장 높은  $M=3$  을 선택한다.



# Maximizing the Evidence Function

---

먼저 다음과 같은 고유벡터식을 먼저 정의해보자.

$$(\beta \Phi^T \Phi) \mathbf{u}_i = \lambda_i \mathbf{u}_i$$

## 1) $\alpha$ 에 대하여 $p(\mathbf{t} | \alpha, \beta)$ 극대화

$\lambda_i + \alpha$ 를 고유값으로 갖는  $\mathbf{A}$ 에 대하여,

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

위 식을 이용하여 marginal likelihood function  $p(\mathbf{t} | \alpha, \beta)$ 를  $\alpha$ 에 대하여 미분한 결과는 다음과 같다.

$$\frac{M}{2\alpha} - \frac{1}{2} \mathbf{m}_N^T \mathbf{m}_N - \frac{1}{2} \sum_i \frac{1}{\lambda_i + \alpha} = 0$$

따라서,  $\gamma$ 와  $\alpha$ 는 다음과 같은 식으로 정리된다.

$$\gamma = \sum_i \frac{\lambda_i}{\lambda_i + \alpha}, \alpha = \frac{\gamma}{\mathbf{m}_N^T \mathbf{m}_N}$$

이 식은  $\alpha$ 에 대한 implicit solution이기 때문에 iterative procedure를 사용하여  $\alpha$ 값을 구할 수 있다:

초기  $\alpha$  값 설정  $\rightarrow \mathbf{m}_N$  구하기  $\rightarrow \gamma$  값 구하기  
 $\rightarrow$  구한  $\mathbf{m}_N, \gamma$ 로  $\alpha$  재추정  $\rightarrow \alpha$ 값이 수렴할 때까지 반복

# Maximizing the Evidence Function

---

## 2) $\beta$ 에 대하여 $p(\mathbf{t} | \alpha, \beta)$ 극대화

$d\lambda_i/d\beta = \lambda_i/\beta$  이기 때문에,

$$\frac{d}{d\beta} \ln |\mathbf{A}| = \frac{d}{d\beta} \sum_i \ln(\lambda_i + \alpha) = \frac{1}{\beta} \sum_i i \frac{\lambda_i}{(\lambda_i + \alpha)} = \frac{\gamma}{\beta}$$

위 식을 이용하여  $p(\mathbf{t} | \alpha, \beta)$ 를  $\beta$ 에 대하여 미분하면 다음과 같다.

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

따라서,  $\beta$ 는 다음과 같은 식으로 정리된다.

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

이 식은  $\beta$ 에 대한 implicit solution이기 때문에 iterative procedure를 사용하여  $\beta$ 값을 구할 수 있다:

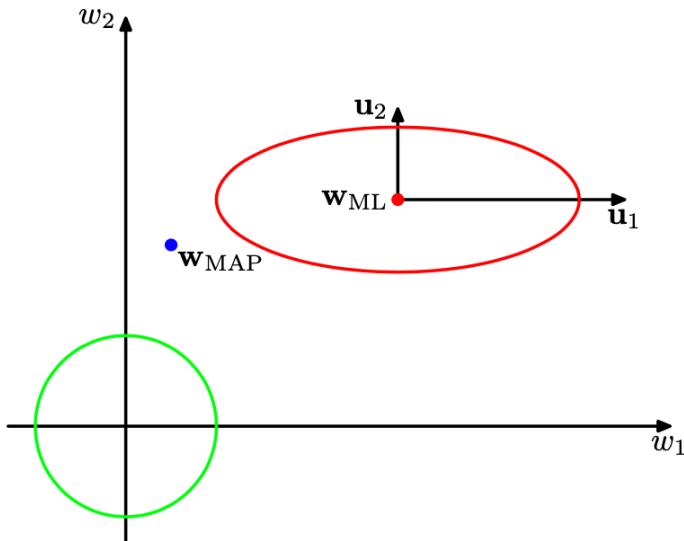
초기  $\beta$ 값 설정  $\rightarrow \mathbf{m}_N$  구하기  $\rightarrow \gamma$  값 구하기  
 $\rightarrow$  구한  $\mathbf{m}_N, \gamma$ 로  $\beta$  재추정  $\rightarrow \beta$ 값이 수렴할 때까지 반복

# Effective Number of Parameters

$\alpha = \frac{\gamma}{\mathbf{m}_N^T \mathbf{m}_N}$  식에서  $\alpha$ 가 주는 의미가 뭔지 해석해보자.

$\beta \Phi^T \Phi$ 는 정방향 행렬이기 때문에, 고유값도 양수이다.

- $\frac{\lambda_i}{\lambda_i + \alpha}$ 는 0과 1 사이,  $0 \leq \gamma \leq M$
- $\alpha = 0$ : 사후 분포의 mode값은  $\mathbf{w}_{ML}$ 로 사용
- $\alpha \neq 0$ : 사후 분포의 mode값은  $\mathbf{w}_{MAP} = \mathbf{m}_N$



1)  $\lambda_i \gg \alpha$ 의 경우 : **Well determined parameters**

- $w_i$ 는 maximum likelihood와 가까움.

- $\frac{\lambda_i}{\lambda_i + \alpha}$ 는 1과 가까워짐.

2)  $\lambda_i \ll \alpha$ 의 경우

- $w_i, \frac{\lambda_i}{\lambda_i + \alpha}$ 는 0과 가까워짐.

$\gamma$ 는 well determined parameters의 수를 측정하는 값에 해당한다.

# Effective Number of Parameters

---

Re-estimating  $\beta$ : maximum likelihood vs. marginal likelihood

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^N (t_n - \mathbf{w}_N^T \phi(\mathbf{x}n))^2 \quad \text{vs.} \quad \frac{1}{\beta} = \frac{1}{N - \gamma} \sum_{n=1}^N (t_n - \mathbf{m}_N^T \phi(\mathbf{x}n))^2$$

- 분모  $N, N - \gamma$  차이가 있음을 확인할 수 있다.
- 두 식 모두 inverse precision을 target과 모델 예측값의 차이의 제곱을 평균낸 것으로 표현한다.
- target distribution의 평균을  $M$ 개의 매개변수를 포함하고 있는  $\mathbf{w}^T \phi(\mathbf{x})$ 라고 표현할 때, 데이터에 의해 결정된 유효한 매개변수의 수는  $\gamma$ , 나머지는  $M - \gamma$ 개가 된다. 베이지안 결과의 분모가  $N - \gamma$ 가 됨으로써 maximum likelihood 사용하여 얻은 값의 bias를 보정해주고 있음을 알 수 있다.
- 단일 변수  $x$ 에 대한 Gaussian 분포 분산의 maximum likelihood estimate은 다음과 같고 (왼쪽 식), 편의 추정량임을 알고 있다. 따라서 분모에  $N$  대신  $N-1$ 을 사용함으로써 편향을 보정해주어 불편 추정량이 되도록 한다.

$$\sigma_{ML}^2 = \frac{1}{N} \sum_{n=1}^N (x_n - \mu_{ML})^2 \quad \text{vs.} \quad \sigma_{MAP}^2 = \frac{1}{N-1} \sum_{n=1}^N (x_n - \mu_{ML})^2$$

# Effective Number of Parameters

---

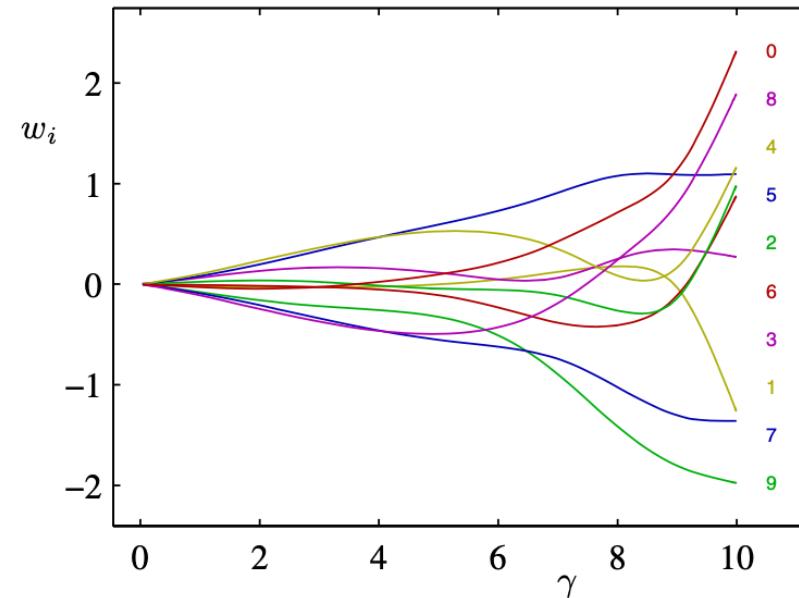
$N \gg M$ 의 경우 (데이터의 수 > 매개변수의 수)

- 데이터 집합의 크기가 증가함에 따라 고유값  $\lambda_i$ 가 증가하기 때문에, 모든 매개변수들을 데이터로부터 well determined될 수 있다.
- 이 경우  $\gamma = M$ 이고,  $\alpha, \beta$ 에 대한 재추정식은 다음과 같다.

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

$\alpha, \gamma$  와  $w_i$

$0 \ll \alpha \ll \infty$  일 때,  $0 \ll \gamma \ll M$ 임을 확인할 수 있다.



# 5

## Bayesian Framework

---

Limitations, Pros and Cons

# Limitations of Fixed Basis Functions

---

기저함수(basis functions)  $\phi_j(\mathbf{x})$ 가 고정 되었다는 가정

- 차원의 저주 (curse of dimensionality)
- 결과적으로 기저 함수의 수가 입력 공간의 차원  $D$ 와 함께 빠르게 증가해야한다.

문제를 완화할 수 있는 두 가지 성질:

1) 데이터 벡터  $\{\mathbf{x}_n\}$ 은 일반적으로 intrinsic dimensionality가 입력 공간의 차원보다 작은 비선형 매니폴드(manifold)에 근접하게 존재한다.

- 입력 변수 간의 강한 상관 관계 때문이다.
- radial basis function networks, support & relevance vector machines에 사용된다.
- 신경망 모델(Neural Network model)은 adaptive basis function을 사용하여 basis functions가 다른 input space의 영역이 data manifold에 해당하도록 매개변수를 조정한다.

2) 대상 변수(target variables)는 데이터 매니폴드내에 일부 방향성에 대해서만 영향을 받는다.

- 신경망에서의 활용: 신경망은 basis functions가 해당하는 input space를 선택하는데 이 속성을 사용한다 (입력 공간의 방향성 선택하는 방식).

# Bayesian Framework

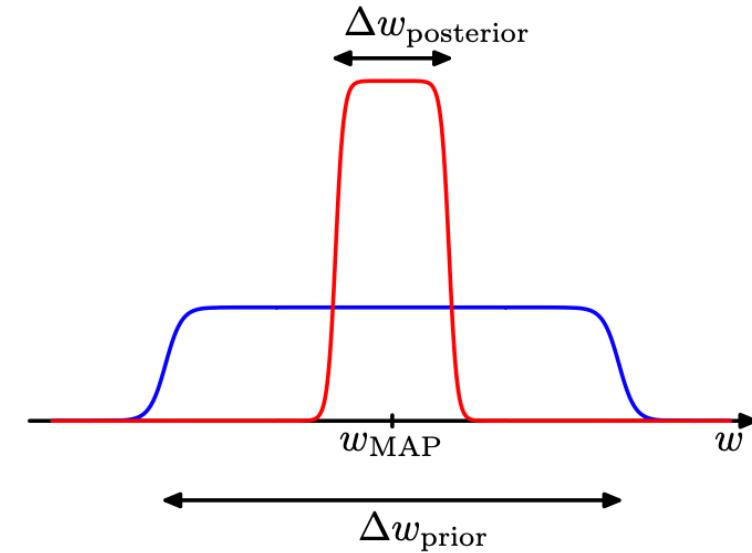
---

## 장점

- 훈련 데이터만을 사용하여 모델 간 비교가 가능하다.
- 과적합(over-fitting) 문제를 방지한다.

## 단점

- 모델의 형태를 가정해야 한다.
- 모델에 대한 잘못된 가정은 잘못된 결과값을 얻게 된다.
- 오른쪽 그림에서 알 수 있듯, Model evidence 영역은 사전 분포의 영향을 많이 받는다. 이 때문에 부적절한 사전 분포를 선택하는 경우, evidence가 정의되지 않을 수 있다. 즉, prior가 적절하지 않으면 잘못된 결과로 이어질 수 있다.
- 따라서 실제로 적용할 때는 학습 데이터와는 독립적인 테스트 데이터 집합을 이용하여 모델의 성능을 평가하게 된다.



---

감사합니다