ESC 2024 Winter Session 3rd Week

Linear Models for Regression

김상민, 김채영, 김효은, 조준태

1. Linear Basis Function Models

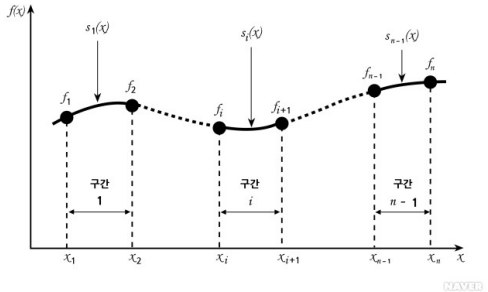
Regression에서 가장 간단한 모델은 input variables의 linear combination 형태, 즉 linear regression model이며 이는 $y(\mathbf{x}, \mathbf{w}) = w_o + w_1 x_1 + ... + w_D x_D$ (3.1), $where\ \mathbf{x} = (x_1, ..., x_D)^T$ 와 같은 형태를 보인다. 위 모델의 핵심은 매개 변수 $w_0, ..., w_D$의 선형 함수 형태라는 점이다. 또한, 입력 변수 x_i의 선형 함수라는 점도 확인할 수 있는데 이로 인해 한계점을 보이기도 한다. 예를 들어, 모델은 선형성을 가정하지만 실제 데이터에서 그러한 선형 관계가 항상 존재하는 것은 아니기 때문에 비선형관계가 나타나는 경우, 모델은 그 관계를 적절하게 파악하지 못한다. 또한, 데이터의 독립성을 가정하지만, 실제로는 데이터 포인트들 사이에 상관 관계가 존재하는 경우가 많아 오차를 과대 추정하거나 과소 추정하는 경우도 종종 발생한다. 따라서, 위와 같은 의 한계점을 해결하기 위해 입력 변수들의 비선형 함수들 즉, 기저함수들의 선형 결합을 활용하게 된다. 그 형태는 아래와 같다.

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

(3.2)식을 보다 예쁘게 정리하기 위해 $\phi_o(\mathbf{x} = \mathbf{1})$로 정의하면 아래 형태의 식으로도 표현이 가능하다.

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

식의 형태에서 파악할 수 있듯 원래의 변수가 x 벡터로 구성되어 있다면 basis function $\phi_j(\mathbf{x})$의 형태로 표현이 가능하다. 입력 변수들의 비선형 함수 형태인 basis function을 활용하므로 입력 벡터 x의 관점에서는 $y(\mathbf{x}, \mathbf{w})$를 비선형 함수 형태로 정의할 수 있고, $\mathbf{w}$의 관점에서는 선형 함수 형태로 정의할 수 있다는 장점이 존재한다. 하지만 Basis function이 입력 변수 x에 대해 global한 함수이므로, input space의 한 영역에서 발생하는 변화가 다른 영역에도 영향을 미친다는 단점이 있으며 이로 인해 함수를 근사할 때 문제가 발생한다. 이와 같은 단점은 input space를 여러 영역으로 쪼개고, 영역마다 다른 polynomial을 피팅하여 해결 가능하며 이러한 아이디어로 만들어진 것이 바로 spline fucntion이다.



Basis function으로 활용할 수 있는 다양한 함수들을 알아보자.
Gaussian basis function은 $\phi_j(x) = exp\{-\frac{(x-\mu_j)^2}{2s^2}\}$ (3.4)와 같은 형태를 띤다. μ_j는 입력 공간에서 기저 함수의 위치를 결정하고, s는 spatial scale을 결정한다. μ_j는 함수의 중심으로, μ_j가 변화하면

함수의 '위치'가 변화한다. 즉, μ_j는 X축 상에서 그래프가 어디에 위치할지 결정하며 μ_j값이 다른 여러개의 가우시안 기저 함수를 사용하면 입력 공간을 잘 나타낼 수 있는 기저를 형성할 수 있다. s는 함수의 표준편차로 s가 커지면 함수의 폭이 넓어지고, 작아지면 폭이 좁하진다. 이처럼 함수의 '스케일'을 조절하여 data point 주변의 어느정도 공간에 대해 함수가 영향력을 미치는지 결정한다. 그리고 향후 모델에서 기저 함수는 adaptive parameter w_j가 곱해질 것이므로, normalization coefficient(정규화 계수)가 중요하지 않다. Adaptive parameter는 data에 따라 그 값이 조절되는 변수이며 이러한 변수에 의해 기저 함수가 scaling되므로 정규화 계수는 중요하지 않다. Sigmoidal basis function은 $\phi_j(x) = \sigma(\frac{x-\mu_j}{s})$와 같은 형태를 보이고, $\sigma(a)$는 $\frac{1}{1+exp(-a)}$와 같이 정의되는 logistic sigmoid function 이다. 또한, $tanh(a) = 2\sigma(a) - 1$이므로 $tanh$함수도 사용할 수 있다. $tanh$함수와 logistic sigmoid function이 위와 같은 관계를 보이므로 $\sigma(a)$의 선형 결합 형태는 $tanh$ 함수의 선형 결합 형태로도 표현이 가능한 것이다. Fourier basis function은 각각의 주파수를 갖는 코사인과 사인 함수로 구성되어 주기성을 가진 함수를 근사하는 데 사용되는 기저 함수다. 주기 함수를 푸리에 기저 함수들의 합으로 분해할 수 있어 복잡한 신호나 데이터를 분석하고 이해하는 데 매우 유용한 도구라고 할 수 있다.
오차 제곱합(Sum of Squares Error/SSE) 함수를 최소화하며 polynomial을 fitting했고, 이러한 접근은 Maximum Likelihood 해를 구하는 것과 같음을 1주차 세션에서 확인했다. 앞으로 least squares approach(최소 제곱법)와 Maximum Likelihood(최대 가능도) 방법 간의 관계를 살펴보고자 한다.

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

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

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

$$maximizing\ lnp(\mathbf{t}|\mathbf{x},\mathbf{w},\beta) \Leftrightarrow\ minimizing\ \frac{1}{2}\Sigma_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2$$

$$\Leftrightarrow\ minimizing\ the\ sum\ of\ squares\ error\ function$$

Target variable t가 결정함수 $y(\mathbf{x},\mathbf{w})$와 gaussian noise ϵ의 합으로 주어진다고 가정해보자
$\mathbf{t} = \mathbf{y}(\mathbf{x},\mathbf{w}) + \epsilon$ $(\mathbf{3.7})$로 표현할 수 있고, $\epsilon \sim N(0,\beta^{-1})$이며 β는 precision(inverse variance)이므로 아래 형태로도 표현이 가능하다.

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

1주차 세션에서 배웠듯, squared loss function $(y(\mathbf{x},\mathbf{w}) - \mathbf{t})^{\mathbf{2}}$을 가정한다면, 새로운 변수 $\mathbf{x}$에 대한 최적의 예측값은 target variable의 조건부 평균이다.

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

$$\rightarrow \frac{\partial E[L]}{\partial y(\mathbf{x})} = 2\int \{y(\mathbf{x}) - t\}p(\mathbf{x},t)dt = 0\ \rightarrow\ y(\mathbf{x}) = \frac{\int tp(\mathbf{x},t)dt}{p(\mathbf{x})} = \int tp(t|\mathbf{x})dt = E[t|\mathbf{x}]$$

그리고 (3.8) 형태의 가우시안 조건부 분포의 조건부 평균은 다음과 같다.

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

이제 input $X = \mathbf{x_1}, ..., \mathbf{x_N}$와 그에 따른 target value $t_1, ..., t_N$으로 이루어진 데이터 조합을 고려해보자. 타겟 변수 t_N을 열벡터 $\mathbf{t}$로 묶을 수 있고 data points가 식 (3.8)에서 독립적으로 추출되었다는 가정 하에 likelihood function을 아래와 같이 얻을 수 있다.

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

$$\rightarrow ln\ p(\mathbf{t}|\mathbf{w}, \beta) = \Sigma_{n=1}^{N}\ lnN(t_n|\mathbf{w}^T\phi(\mathbf{x_n}), \beta^{-1}) = \frac{N}{2}ln\beta - \frac{N}{2}ln(2\pi) - \beta E_D(w) \quad (3.11)$$

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

이제 위 식의 $\mathbf{w}$, β에 Maximum Likelihood 방법을 적용해보자. 먼저 $\mathbf{w}$에 대해 최대화해보자

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

이 gradient를 0으로 놓으면 아래와 같은 결과를 얻게 된다.

$$0 = \Sigma_{n=1}^{N} t_n\phi(\mathbf{x_n})^T - \mathbf{w}^T(\Sigma_{n=1}^{N}\phi(\mathbf{x_n})\phi(\mathbf{x_n})^T) \quad (3.14)$$

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

(3.15)를 normal equations for least squares problem(최소 제곱 문제의 정규 방정식)이라고 하고, Φ는 $\Phi_{nj} = \phi_j(\mathbf{x_n})$을 원소로 갖는 $N \times M$행렬로 design matrix라고 부른다.

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

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

Moore-Penrose pseudo-inverse (3.17)는 역행렬 개념을 non square 행렬들에 대해 일반화한 것이다. 행렬 Φ가 가역인 정사각형 행렬이라면 $(\mathbf{AB})^{-1} = \mathbf{B^{-1}A^{-1}}$이라는 성질을 바탕으로 $\Phi^+ \equiv \Phi^{-1}$임을 확인할 수 있다. 식의 bias parameter w_0에 대해서도 알아보자.

$$E_D(w) = \frac{1}{2}\Sigma_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x_n})\}^2 \quad (3.12) = \frac{1}{2}\Sigma_{n=1}^{N}\{t_n - w_0 - \Sigma_{j=1}^{M-1}w_j\phi_j(\mathbf{x_n})\}^2 \quad (3.18)$$

w_0에 대한 미분값을 0으로 두고 w_0에 대해 식을 풀면 아래와 같은 결과를 얻을 수 있다.

$$w_0 = \bar{t} - \Sigma_{j=1}^{M-1}w_j\bar{\phi_j} \quad (3.19),\ where\ \bar{t} = \frac{1}{N}\Sigma_{n=1}^{N}t_n,\ \bar{\phi_j} = \frac{1}{N}\Sigma_{n=1}^{N}\phi_j(\mathbf{x_n}) \quad (3.20)$$

위처럼 w_0은 training set의 target values의 평균과 기저 함수 값의 평균들의 weighted sum, 총 2개의 term으로 이루어져있음을 확인할 수 있다. 즉, bias parameter w_0은 training set의 target values의 평균과 기저 함수 값의 평균들의 weighted sum 간의 차이를 보상하는 역할을 한다. 또한, β에 대해서 최대화를 진행하면, 아래와 같은 결과를 얻을 수 있다.

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

$\mathbf{w}_{ML} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{t}$ $(\mathbf{3.15})$와 같은 batch techniques는 한번에 전체 training set을 처리하여 data set이 클 경우에는 시간과 연산 비용 측면에서 문제가 발생한다. 이때 sequential 알고리즘이 유용하게

활용된다. Sequential 알고리즘은 전체를 한번에 처리하는 것아 아니라 한 번에 하나의 데이터 포인트를 고려하고, 그 때마다 모델의 parameter를 update한다. 만약 error function이 data points($E = \Sigma_D E_N$)의 오류 함수의 값과 같다면, SGD를 활용해서 sequential learning을 진행할 수 있다. SGD는 무작위로 하나의 샘플을 선택한 뒤 한 번의 update에 하나의 데이터 샘플만을 사용한다. 수식으로 살펴보자면, 패턴 n이 등장한 후, parameter vector $\mathbf{w}$를 아래와 같이 update한다.

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n \quad (3.22),\ where\ \tau : iteration\ number,\ \eta : learning\ rate\ parameter$$

Sum of Squares Error function의 경우에는 다음과 같다.

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)T}\phi_n)\phi_n \quad (3.23),\ where\ \phi_{\mathbf{n}} = \phi(\mathbf{x_n})$$

위 식은 Least Means Square (LMS) 알고리즘이며, η 값은 알고리즘이 수렴하도록 선택되어야 한다. Overfitting 문제를 해결하는 방법은 크게 2가지 : training set의 data를 늘리거나, regularization을 이용하는 것이 있다. 하지만, 전자는 현실적인 한계가 있기 때문에 우리에게 중요한 것은 후자이다. Error function에 가장 단순한 형태의 regularization term(정규화항)인 sum of squares of weight vector elements $E_W(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$ (3.25)를 더하면 오류 함수는 아래와 같은 형태를 보인다.

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

$$\frac{1}{2}\Sigma_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x_n})\}^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \quad (3.27)$$

위 식에서 λ는 데이터에 종속적인 에러 $E_D(\mathbf{w})$와 정규화항 $E_w(\mathbf{w})$간의 상대적인 중요도를 조절하는 regularization coefficient(정규화 계수)이다. 위에서 사용한 정규화항은 오류 함수가 $\mathbf{w}$에 대한 이차 함수 형태로 나타나 오류 함수를 최소화하는 값을 closed form으로 구할 수 있다는 장점이 있다. 식 (3.27)의 gradient를 0으로 두고, $\mathbf{w}$에 대해 풀어보면 아래의 결과를 얻을 수 있으며 이는 (3.15) least squares solution이 확장된 형태임을 확인할 수 있다.

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

보다 일반적인 형태의 regularizer(정규화항)을 사용하는 경우의 정규화 오류 함수(regularized error)는 어떨까? 이는 아래와 같은 모습을 띈다.

$$\frac{1}{2}\Sigma_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x_n})\}^2 + \frac{\lambda}{2}\Sigma_{j=1}^{M}|w_j|^q \quad (3.29)$$

q=2일 때의 경우가 위에서 언급된 quadratic regularizer (3.27)이고, q=1일 때의 경우는 lasso라고 한다. Lasso는$J(\theta) = MSE(\theta) + \alpha\Sigma_{i=1}^{n}|\theta_i|$ 형태로 MSE+가중치들의 절댓값의 합을 최소로 만들어야 한다는 제약으로 구성되어 있다고 할 수 있다. 이를 통해, λ가 커지면 몇몇 계수 $\mathbf{w_j}$가 0으로 가서 그에 상응하는 basis function이 아무 역할을 하지 못하는 sparse model을 만드는 성질이 있다.

2. Frequentist and Bayesian Perspective on Model Complexity

Bias-Var tradeoff
모델의 복잡도

우리는 Maximum Likelihood 방법에서 복잡한 모델을 사용할때 overfitting 문제가 있음을 확인 하였다. 이를 피하기 위하여 여러가지 방법(basis function수의 제한/regularization)들이 있었지만 어려움들이 있었다. 이러한 모델의 복잡도는 또한 Bias,Var와 관련이 있는데 살펴보자.

기대 오류의 분해

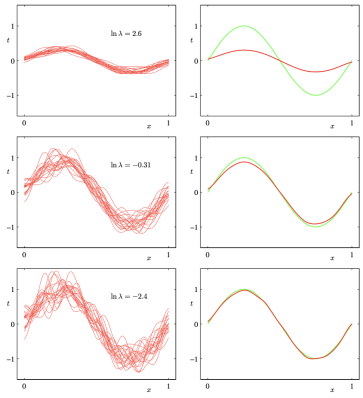
$$E[L] = \int [y(\mathbf{x}) - h(\mathbf{x})]^2 p(\mathbf{x}) d\mathbf{x} + \int \int [h(\mathbf{x}) - t]^2 p(\mathbf{x}, t) d\mathbf{x} dt$$

첫번째 항은 reducible error : $y(x)$ 를 $h(x)$에 가깝게 추정함으로써 줄일 수 있다.
두번째 항은 irreducible error : 내재된 노이즈이기에 줄일 수 없다.

reducible error의 분해

$$E[[y(\mathbf{x}) - h(\mathbf{x})]^2] = [E([y(x|D) - h(\mathbf{x})]]^2 + E[\{y(\mathbf{x}|D) - E[y(\mathbf{x}|D]\}^2]$$

첫번째 항 $(Bias)^2$: 모델의 복잡도가 증가할 수록 작아진다.
두번째 항 Var : 모델의 복잡도가 증가할수록 커진다.
이렇듯 편향과 분산 사이의 가장 좋은 밸런스를 가지는 모델이 좋은 모델이다. 아래 그림을 보면 λ가 작아질수록 모델의 복잡도가 커져 모델의 분산은 커지고 모델의 편향은 작아짐을 확인할 수 있다.



Frequentist 관점에서 모델의 복잡도

현실 속 사례에서는 한 개의 observation이 주어지기에 Freuquentist 관점에서의 모델의 복잡도의 해석이 실제적 가치가 제한적이다. 이에 모델의 복잡도에 관련된 실용적 teqnique를 제공하는 Bayesian 관점에 대해 살펴보자.

Bayesian Linear Regression
Likelihood function

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

Prior distribution

$$p(\mathbf{w}) = \mathbf{N}(\mathbf{w}|\mathbf{m_0}, \mathbf{S_0})$$

Posterior distribution

$$p(\mathbf{w}|\mathbf{t}) = \mathbf{N}(\mathbf{w}|\mathbf{m_N}, \mathbf{S_N})$$

$$\mathbf{m_N} = \mathbf{S_N}(\mathbf{S_0^{-1}}\mathbf{m_0} + \beta \mathbf{\Phi^T} \mathbf{t})$$

$$S_N^{-1} = S_0^{-1} + \beta\Phi^T\Phi$$

이렇듯 사전분포와 사후분포간에 congugacy를 이용하여 쉽게 사후분포를 도출할 수 있었다. 여기서 사후분포는 가우시안분포이기에 최빈값과 평균값이 일치한다. 또한 $S_0 = \alpha^{-1}\mathbf{I}$ $(\alpha \to \mathbf{0})$ 일때 MAP와 MLE가 같아지게 된다. 베이지안 선형회귀분석은 N=0일때는 사전분포를 사후분포로 사용하며 데이터가 추가될때마다 각 단계에서 순차적으로 사후분포를 업데이트 하는 방식으로 진행된다.

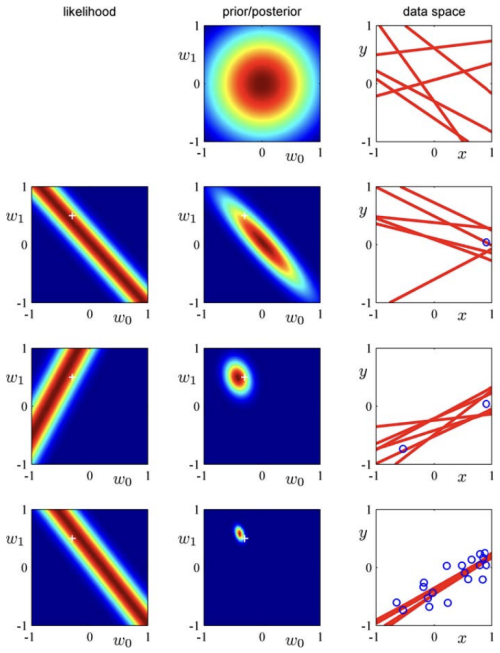
베이지안 선형회귀의 예시

- 단순화를 위한 가정

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

사후분포 : $p(\mathbf{w}|\mathbf{t}) = N(\mathbf{w}|\mathbf{m_N}, \mathbf{S_N})$ $where$ $\mathbf{m_N} = \beta\mathbf{S_N}\phi^{\mathbf{T}}\mathbf{t}$, $\mathbf{S_N^{-1}} = \alpha\mathbf{I} + \beta\phi^{\mathbf{T}}\phi$

- target variable 생성 $t = -0.3 + 0.5x + \epsilon$ 에서 랜덤하게 t를 생성한다. (이때, ϵ is iid $N(0, 0.04)$)



위 그림에서 1행은 N=0, 2행은 N=1, 3행은 N=3, 4행은 N=20이다. 1열은 Likelihood function, 2열은 posterior distribution, 3열은 data space이고 1열과 2열에서 + 점은 앞서 가정한 실제 모수이다. 또한 그래프는 등고선의 형태로 주어졌다. 3열의 파란색 원은 data point, 빨간색은 추정된 회귀직선들이다. 이 그래프를 해석해보자. N의 크기에 따라 어떻게 학습하는지와 순차적인 학습의 속성에 주목하자. 1행을 보자. 데이터가 없기에 가능도함수는 그려지지 않았고 사후분포는 앞서 가정한 사전분포를 사용했다. 여기서 빨간색 선은 사후분포로부터 추정된 6개의 추정선들이다. 2행을 보자. 추가된 데이터로 그려진 Likelihood함수가 생겼고 1행의 사후분포와 이 Likelihood함수를 곱하고 정규화하여 2행의 사후분포를 만들어냈다. 이렇게 만든 사후분포로 6개의 추정선 빨간색선을 만들어냈다. 이때 정밀도 β에 따라 이 포인트와 벗어난 정도가 조절이 된다. 3행을 보자. 추가된 데이터로 그려진 Likelihood함수가 생겼고 2행의 사후분포와 이 Likelihood함수를 곱하고 정규화하여 3행의 사후분포를 만들어냈다. 이렇게 만든 사후분포로 6개의 추정선 빨간색선을 만들어냈다. 여기서 Likehood는 새롭게 추가된 데이터1개에 대한 것이다. 이때 방금 만든 사후분포는 데이터 2개에 대한 가능도함수와 처음 사전분포를 합쳐서 만든 사후분포와 동일하다. 4행 또한 같은 방식으로 20번째 데이터를 관찰한 후 생성되었다.이렇게 데이터가 늘어남에 따라 사후분포가 실제모수 + 점에 가까워지고 분산 또한 줄어드는 것을 볼 수 있다.

Predictive distribution
우리는 실제로는 $\mathbf{w}$ 자체에 관심이 있는 것이 아니라 새로운 $\mathbf{x}$의 값에 대한 $\mathbf{t}$의 예측에 관심이 있으며 이를 위해서는 아래와 같은 형태를 띠는 Predictive distribution을 고려할 필요가 있다.

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

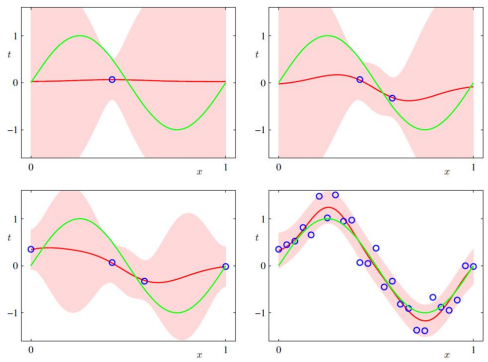
$\mathbf{t}$: vector of target values from the training set
$p(t|\mathbf{w},\beta)$: the conditional distribution of the target variable, Gaussian distribution
$p(\mathbf{w}|\mathbf{t},\alpha,\beta)$: the posterior weight distribution, Gaussian distribution

$$p(t|\mathbf{x},\mathbf{t},\alpha,\beta) = \mathcal{N}(t|\mathbf{m}_N^T\phi(\mathbf{x}),\sigma_N^2(\mathbf{x}))$$

Variance of the predictive distribution 은 아래와 같은 형태를 띤다.

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

여기에서 $\frac{1}{\beta}$ 는 데이터의 noise를 나타내며, $\phi(\mathbf{x})^T\mathbf{S}_N\phi(\mathbf{x})$ 는 파라미터 $\mathbf{w}$의 불확실성을 반영한다.
$\sigma_{N+1}^2(\mathbf{x}) \leq \sigma_N^2(\mathbf{x}) \because$ $as\ N \to \infty,\ \phi(\mathbf{x})^T\mathbf{S}_N\phi(\mathbf{x}) \to 0$



Data points의 주변에서, 그리고 많은 data points가 관측될수록, 불확실성은 줄어든다.

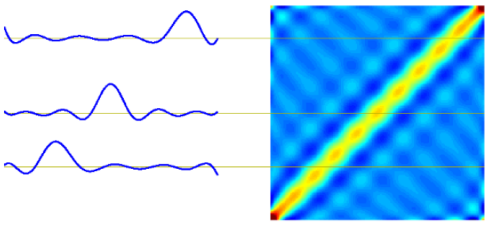
Equivalent kernel
Predictive mean

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

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

Smoother matrix or Equivalent kernel

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



위 그림은 3개의 다른 x에 대해 x'의 kernel function을 나타낸 것이다. 붉을수록 $k(\mathbf{x}, \mathbf{x}')$이 높은 값을 갖게 된다. 즉, x와 x'가 가까울수록, local evidence를 높게 가중할 수 있다.

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

Equivalent kernel은 $y(\mathbf{x})$와 $y(\mathbf{x}')$의 covariance로 볼 수 있다. 이를 통해서 가까운 점에서 predictive mean은 높은 상관 관계가 있으며, 멀리 떨어진 점들은 낮은 상관 관계가 있음을 알 수 있다.
Equivalent kernel은 학습 데이터와 예측할 새로운 데이터 x의 결합으로 이루어지는데, 이 가중치의 합이 1이 된다.

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

3. Bayesian Model Comparison

베이지안 관점에서의 모델 비교는 모델의 선택에 있어서 불확실성을 나타내기 위해 확률을 사용한다.

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

$p(\mathcal{M}_i)\ where\ i = 1, ..., L$: prior probability distribution expressing uncertainty
$\mathcal{D}$: training set
$p(\mathcal{D}|\mathcal{M}_\rangle)$: model evidence는 marginal likelihood

$$\frac{p(\mathcal{M}_i|D)}{p(\mathcal{M}_j|D)} \propto \frac{p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_|)}{p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_|)} \propto \frac{p(\mathcal{D}|\mathcal{M}_\rangle)}{p(\mathcal{D}|\mathcal{M}_|)}$$

Prior는 알고 있는 값이 되어, 우리가 주목할 것은 Bayes Factor $p(\mathcal{D}|\mathcal{M}_\rangle)/p(\mathcal{D}|\mathcal{M}_|)$이다.

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

위의 mixture distribution 예에서 전체적인 predictive distribution은 predictive distribution인 $p(t|\mathbf{x}, \mathcal{M}_i, \mathcal{D})$에 posterior probabilities인 $p(\mathcal{M}_i|\mathcal{D})$로 가중하여 얻을 수 있다. Model averaging의 가장 간단한 근사는 하나의 가장 가능성이 높은 모델을 예측에 사용하는 것이며 이를 model selection이라고 한다.
Model이 하나의 parameter w만 가진다고 가정해보자. 사후 분포는 $p(\mathcal{D}|w)p(w)$에 비례할 것이다. 만약 사후 분포가 w_{MAP} 주위에 $\Delta w_{posterior}$의 폭으로 sharply peaked 됐다고 가정해보자. prior가 Δw_{prior}의 폭으로 평평하여 $p(w) = 1/\Delta w_{prior}$ 이라고 한다면 아래와 같이 근사할 수 있다.

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

위 근사 식에 로그를 취하면, $lnp(\mathcal{D}) \simeq lnp(\mathcal{D}|w_{MAP}) + ln(\frac{\Delta w_{posterior}}{\Delta w_{prior}})$가 된다.
첫 번째 term은 가장 가능성이 높은 파라미터 값으로 주어진 데이터의 적합을 나타내고, 두 번째 term은 복잡성에 따른 모델을 penalize한다. $\Delta w_{posterior} < \Delta w_{prior}$이면 두 번째 term은 음수이기 때문에 $\Delta w_{posterior}/\Delta w_{prior}$가 작아지면 절댓값이 커진다. 따라서, 파라미터가 posterior distribution의 data에 맞춰진다면 penalty term은 증가한다.
모델이 M개의 파라미터 집합을 갖는다고 한다면 각각의 파라미터에 대해 유사한 근사치를 적용할 수 있다. 모든 파라미터가 같은 $\Delta w_{posterior}/\Delta w_{prior}$ 비율을 갖는다고 가정하면, 아래와 같이 구할 수 있다.

$$lnp(\mathcal{D}) \simeq lnp(\mathcal{D}|w_{MAP}) + Mln(\frac{\Delta w_{posterior}}{\Delta w_{prior}})$$

복잡한 모델 : 모델 피팅이 잘 이루어지므로 첫 번째 term 증가, M에 의해 penalty term 증가
간단한 모델 : 첫 번째 term 감소, penalty term 감소
따라서, 두 모델 사이에서 적절한 복잡도를 갖는 모델을 선호하게 된다.
베이지안 모델 비교 방법은 데이터가 생성된 실제 분포가 고려 중인 모델에 속해있다고 가정한다. 두 개의 모델 $\mathcal{M}_1$, $\mathcal{M}_2$ 중 $\mathcal{M}_1$이 실제 모델인 상황을 생각해보자. Data sets의 분포에 대한 베이즈 요인의 평균을 구하면 아래와 같은 형태가 나온다.

$$\int p(\mathcal{D}|\mathcal{M}_1) ln \frac{p(\mathcal{D}|\mathcal{M}_1)}{p(\mathcal{D}|\mathcal{M}_2)} d\mathcal{D}$$

위는 Kullback Leibler 발산의 예이며, 두 분포가 일치하지 않는다면 항상 양수가 된다는 특징이 있다. 따라서 평균적으로 베이즈 요인은 항상 옳은 모델을 선호한다.

Bayesian Framework
장점 : over-fitting 문제를 방지하고, 훈련된 데이터를 기반으로 model을 비교한다.
단점 : 가정한 모델이 유효하지 않으면 결과가 잘못될 수 있다.
따라서, 실제 적용에서는 최종 시스템을 평가하기 위해 독립적인 test set of data를 두어야 한다

4. The Evidence Approximation

Fully Bayesian 관점을 바탕으로 한 선형 기저 함수 모델에서는 사전 분포에 α(precision of the prior), β(precision of the noise)와 같은 초매개변수(hyperparameters)를 도입하여 추론을 진행한다. 보통 초매개변수, $\mathbf{w}$에 대한 marginalization은 계산이 복잡하고 수학적으로 다루기 어렵기 때문에 marginal likelihood(혹은 model evidence)를 극대화하는 초매개변수 값을 구하고, 이들을 해당 값으로 고정하는 방법을 선택한다. $\alpha, \beta, \mathbf{w}$에 대하여 marginalize해서 얻은 predictive distribution은 다음과 같다.

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

이때 사후 분포 $p(\alpha,\beta|\mathbf{t})$가 $\hat{\alpha},\hat{\beta}$ 주위에 몰려있다면(sharply peaked), 초매개변수 값을 $\hat{\alpha},\hat{\beta}$으로 고정하여 다음과 같은 식으로 근사할 수 있다.

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

앞서 베이즈 정리를 통해 사후 분포는 $p(\alpha,\beta|\mathbf{t}) \propto p(\mathbf{t}|\alpha,\beta)p(\alpha,\beta)$와 같은 비례관계에 있음을 확인할 수 있다. 사전분포가 상대적으로 flat하다가고 가정하면, $\hat{\alpha},\hat{\beta}$은 marginal likelihood function을 극대화함으로써 구할 수 있다. 여기서 극대화에는 두 가지 방법이 있다. 첫 번째는 로그를 취한 marginal likelihood function을 미분한 후 해당 도함수=0 으로 설정하여 α,β에 대한 재추정식(re-estimation equation)을 구하는 방법이고, 두 번째는 Expectation Maximization 알고리즘(EM 알고리즘)을 이용하는 방법이다. 이 중, 3장에서는 전자를 이용하여 초매개변수 값을 구하는 과정을 보여준다.
먼저, marginal likelihood function (혹은 evidence function)에 대해 알아보자. $\mathbf{w}$에 대한 marginalization을 통해 얻은 marginal likelihood function은 다음과 같다.

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

정규화된 Gaussian 계수를 가지는 표준 형태를 이용하면, 위 식은 다음과 같이 표현된다.

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

위 식을 정리하기 위해 $E(\mathbf{w})$ 식을 구해보자. M이 $\mathbf{w}$의 차원이라고 할 때, $E(\mathbf{w})$는 다음과 같다.

$$E(\mathbf{w}) = \beta E_D(\mathbf{w}) + \alpha E_D(\mathbf{w}))$$

이 식은 regularized sum-of-squares error function과 상수만 다를 뿐 동일한 식임을 알 수 있고, 이를 참고하여 $E(\mathbf{w})$를 다음과 같은 식으로 나타낼 수 있다.

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

앞서 우리는 $\mathbf{A}, \mathbf{m}_i$를 다음과 같이 정의한 바 있다.

$$\mathbf{A} = \alpha\mathbf{I} + \beta\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{S}_N^{-1}$$

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

이때 $\mathbf{A}$는 $\nabla\nabla E(\mathbf{w})$인 Hessian matrix이다. 위에서 구한 식을 이용하여 $\mathbf{w}$에 대한 적분을 진행하면 다음과 같다.

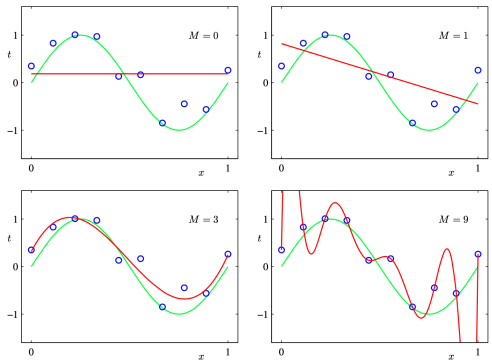
$$\int exp\left\{-E(\mathbf{w})\right\} d\mathbf{w} = exp\left\{-E(\mathbf{m}_N)\right\} \int exp\left\{-\frac{1}{2}(\mathbf{w} - \mathbf{m}_N)^T \mathbf{A}(\mathbf{w} - \mathbf{m}_N)\right\} d\mathbf{w}$$

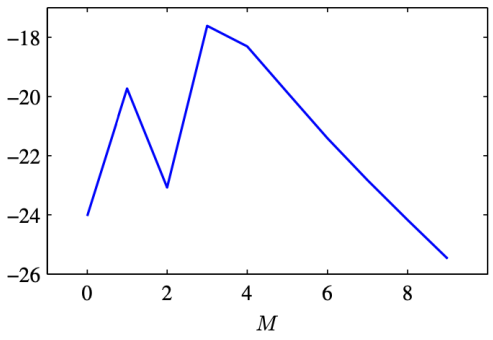
$$= \left\{-E(\mathbf{m}_N)\right\} (2\pi)^{M/2} |\mathbf{A}|^{-1/2}$$

따라서, log marginal likelihood는 다음과 같다.

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

그래프를 통해서도 model evidence를 최대로 하는 모델을 찾아 선택할 수 있다. 1장의 polynomial regression problem 예시를 통해 알아보자.





해당 예시에서 $\alpha = 5 * 10^{-3}$으로 고정되어 있다. 1장의 curve fitting 과정에서 M이 커질 때 예측이 어떻게 되는지 살펴보았다. 이와 관련하여 model evidence 값의 변화를 M 값의 변화와 curve fitting을 연관지어서 알아보자. M이 0, 1일 때는 실제 값에서 많이 벗어나게 예측이 되고 model evidence 값도 작다는 것을 확인할 수 있다. M이 2일 때, 기함수 형태의 실제 값과 달리 우함수 형태의 예측 값을 사용함으로써 model complexity의 크기는 증가했지만 오히려 예측이 더 벗어나면서 model evidence는 M이 1일 때보다 더 감소한다. M=3 일 때, 그래프 상 fit이 가장 잘 되었으며 아래 그래프를 통해 model evidence도 최댓값을 가지는 것을 확인할 수 있다. 3 이후로는 data fit에서의 적은 증가에 비해 model complexity에 대한 penalty가 커지면서 결과적으로는 evidence 값이 감소한다. 따라서, 그래프를 통해 model evidence가 가장 높은 M=3이 가장 적절한 모델이며, 이를 선택할 것임을 알 수 있다.
다음으로 evidence function을 극대화하는 α, β를 구하는 과정을 살펴보자. 먼저, 다음과 같은 고유벡터 식을 정의한다. $(\beta\mathbf{\Phi}^T\mathbf{\Phi})\mathbf{u}_i = \lambda i\mathbf{u}_i$ 이에 따라, $\mathbf{A}$는 $\alpha + \lambda_i$의 고유값을 가진다.
α에 대한 $p(\mathbf{t}|\alpha, \beta)$의 극대화 과정을 알아보자. $\alpha + \lambda_i$를 고유값으로 갖는 $\mathbf{A}$를 α에 대해 미분한 식은 다음과 같다.

$$\frac{d}{d\alpha} ln\,|\mathbf{A}| = \frac{d}{d\alpha} ln \prod(\lambda i + \alpha) = \frac{d}{d\alpha} \sum_i ln(\lambda i + \alpha) = \sum_i \frac{1}{\lambda_i + \alpha}$$

따라서 stationary point는 다음 식을 만족한다.

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

이 식을 정리하면 γ는 다음과 같은 식을 통해 구할 수 있다.

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

$$\gamma = \sum_i \frac{\lambda_i}{\lambda_i + \alpha}$$

따라서, evidence function을 극대화하는 α값은 다음과 같다.

$$\alpha = \frac{\gamma}{\mathbf{m}_N^T\mathbf{m}_N}$$

이는 α에 대한 음함수해(implicit solution)이기 때문에, iterative procedure를 통해 α값을 구할 수 있다. iterative procedure는 다음과 같다 :
초기 α 값 설정 $\rightarrow \mathbf{m}_N$ 구하기 $\rightarrow \gamma$ 값 구하기 $\rightarrow$ 구한 $\mathbf{m}_N, \gamma$ 값으로 α 재추정 $\rightarrow \alpha$가 수렴할 때까지 반복

다음으로 β에 대한 $p(\mathbf{t}|\alpha, \beta)$의 극대화 과정을 살펴보자. 이번에는 β에 대하여 미분을 진행한다.

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

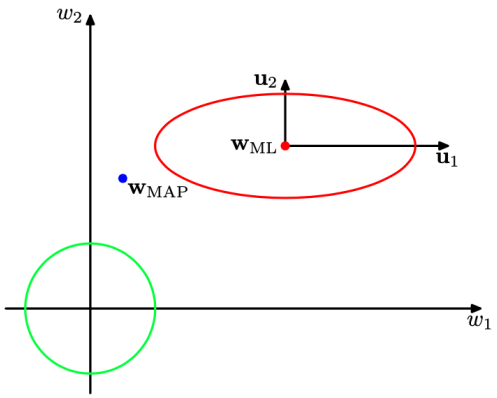
stationary point는 다음 식을 만족한다.

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

위 식을 β에 대해 정리하면 다음과 같다.

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

α 값을 구할 때와 같이, iterative procedure을 이용하여 β 값을 구할 수 있다 :
초기 β 값 설정 $\rightarrow \mathbf{m}_N$ 구하기 $\rightarrow \gamma$ 값 구하기 $\rightarrow$ 구한 $\mathbf{m}_N, \gamma$ 값으로 β 재추정 $\rightarrow \beta$가 수렴할 때까지 반복
앞서 구한 $\alpha = \frac{\gamma}{\mathbf{m}_N^T \mathbf{m}_N}$ 식에서 α가 주는 의미가 뭔지 해석해보자. 여기서 고유값 λ_i는 likelihood의 곡률을 측정하며, $\beta \mathbf{\Phi}^T \mathbf{\Phi}$는 정방향 행렬(positive definite)이기 때문에 고유값도 양수이다. 이 때문에 $\frac{\lambda_i}{(\lambda_i + \alpha)}$는 0과 1 사이, $0 \leqslant \gamma \leqslant M$이다. 또한, $\alpha = 0$일 때 사후 분포의 mode값은 $\mathbf{w}_{ML}$, 0이 아닐 때는 $\mathbf{w}_{MAP} = \mathbf{m}_N$으로 사용한다.



$\lambda_i \gg \alpha$ 경우, w_i는 maximum likelihood 값과 가까워지며 $\frac{\lambda_i}{\lambda_i + \alpha}$는 1과 가까워진다. 반대로, $\lambda_i \ll \alpha$ 경우에는 w_i과 $\frac{\lambda_i}{\lambda_i + \alpha}$ 모두 0과 가까워진다. 따라서 γ는 Well determined parameters의 수를 측정하는 값에 해당한다.
이와 관련하여 maximum likelihood와 marginal likelihood를 통해 얻은 β 값을 비교해보자. 두 식은 각각 다음과 같다.

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

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

두 식 모두 inverse precision을 target과 모델 예측 값의 차이의 제곱을 평균낸 것으로 표현한다. 하지만 분모가 $N, N - \gamma$라는 점에서 차이가 있다. 두 식의 차이에 대해 알아보기 위해 단일 변수 x에 대한 Gaussian 분포 분산의 maximum likelihood estimate를 다시 떠올려보자. ML 추정량은 다음과 같으며, 이는 편의 추정량이다.

$$\sigma_{ML}^2 = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu_{ML})^2$$

분모에 N 대신 $N - 1$을 사용함으로써 편향을 보정해주어 불편 추정량이 되도록 할 수 있다. 즉, 분모 인자 $N - 1$이 하나의 자유도가 평균값을 근사하는데 사용되었음을 고려한다.

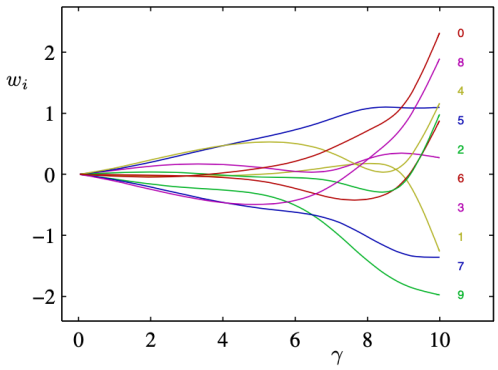
$$\sigma_{MAP}^2 = \frac{1}{N - 1} \sum_{n=1}^{N} (x_n - \mu_{ML})^2$$

이와 유사하게 β에 대한 두 식의 차이를 해석할 수 있다. Target distribution의 평균을 M개의 매개변수를 포함하고 있는 $\mathbf{w}^T \phi(\mathbf{x})$라고 할 때, 데이터에 의해 결정된 유효한 매개변수(effective parameters)의 수는 γ, 나머지는 $M - \gamma$가 된다. 베이지안 결과의 분모가 $N - \gamma$가 됨으로써 maximum likelihood

를 사용해서 얻은 값의 편향을 보정해주고 있음을 알 수 있다.
다음으로 데이터의 수가 매개변수의 수보다 클 때 ($N \gg M$) α, β에 대한 재추정식에 대해 알아보자. 데이터의 집합의 크기가 증가함에 따라 고유값 λ_i가 증가하기 때문에, 모든 매개변수들은 데이터로부터 well determined될 수 있다. 이 경우에 α, β에 대한 재추정식은 다음과 같다.

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

α, γ, w_i가 어떤 식으로 연관되어 있는지는 다음 그래프를 통해 알 수 있다. Gaussian 기저 함수 모델에서 10개의 매개변수 w_i와 γ 사이 그래프이며 $0 \ll \alpha \ll \infty$일 때 $0 \ll \gamma \ll M$임을 확인 가능하다.



5. Bayesian Framework

우리는 이번 장을 통해 고정된 비선형 기저 함수를 가지는 선형 결합 모델들을 알아보았다. 하지만 기저 함수가 고정되었다는 가정이 문제를 파생하기도 한다. 기저 함수 $\phi_j(\mathbf{x}$가 고정되어 있다는 가정으로 인해 관찰된 학습 데이터의 차원이 증가하는 경우 1장에서 언급한 차원의 저주(curse of dimensionality))가 나타나게 된다. 결국 차원이 증가할수록 기저 함수의 수가 입력 공간의 차원 D와 함께 빠르게 증가하게 된다. 하지만, 이러한 문제를 완화할 수 있는 두 가지 성질이 존재한다. 첫 번째로, 데이터 벡터 x_n은 일반적으로 intrinsic dimensionality가 입력 공간의 차원보다 작은 비선형 매니폴드(manifold)에 근접하게 존재한다는 성질이다. 이는 입력 변수 간의 강한 상관 관계 때문이다. 이러한 성질은 radial basis function networks, support와 relevance vector machines에 사용된다. 신경망 모델(Neural Network Model)은 adaptive basis function을 사용하여 기저 함수가 다른 입력 공간의 영역이 데이터 매니폴드에 해당하도록 매개변수를 조정한다. 두 번째 성질은 대상 변수가 데이터 매니폴드내 일부 방향성에 대해서만 영향을 받는다는 점이다. 신경망에서는 기저 함수가 해당하는 입력공간을 선택하는데 이를 활용한다. 즉, 입력 공간의 방향성을 선택하는 방식에 이 성질을 사용하는 것이다.
베이지안 관점은 장점과 단점 모두 존재한다. 베이지안 관점은 훈련 데이터만을 사용하여 모델 간 비교가 가능하다는 점과 과적합(over-fitting) 문제를 방지한다는 점에서 장점을 가진다. 하지만 베이지안 관점은 모델의 형태를 미리 가정해야 하기 때문에 모델에 대한 잘못된 가정이 잘못된 결과값을 불러온다는 점에서 한계가 있다. 또한, model evidence의 경우 사전 분포의 영향을 많이 받기 때문에 부적절한 사전 분포를 선택하는 경우 evidence가 정의되지 않을 수 있다는 점도 단점이다. 따라서 실제로 적용할 때는 학습 데이터와는 독립적인 테스트 데이터 집합을 이용하여 모델의 성능을 평가하게 된다.