Mathematical Bases of Machine Learning Technics

Hyunwoo Kim of Inha University

Contents

- Bayes' Theorem
- Maximum Likelihood Estimation as Base of Machine Learning
- Information Theory View
- Concept of Convolution
- Kernel Trick

Frequentism vs Bayesian

빈도론(Frequentism) : 모델의 모수(parameter)는 고정된 특정한 값

- 확률 : 사건이 일어난 빈도의 비율
- 신뢰구간 n% : 무한대의 표본을 추출할 때 n%가 모집단 값을 가짐

베이지안(Bayesian) : 모수는 불확실성을 갖는 확률 분포

- 확률 : 어떤 사건이 일어날 것이라고 알고 있는 정도
- 사전 확률을 생각 → 데이터를 통해 가능도 계산 → 사전 확률 보정

Bayes' Theorem

우도(likelihood)

사전 확률(prior)

$$p(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{x})}$$

사후 확률(posterior)

사전 확률 : 일종의 선입견, 예측

우도 : 모델의 모수가 데이터를 얼마나 잘 설명하는가

Maximum Likelihood Estimation(MLE)

"The happening done is the event that occurs most frequently, i.e., the event that you see takes place with maximum probability"

$$\mathcal{L}_{\mathbf{x}}(\boldsymbol{\theta}) \equiv -\log p(\mathbf{x} \,|\, \boldsymbol{\theta})$$

$$p(y_n \,|\, \mathbf{x}_n, \boldsymbol{\theta}) \equiv \mathcal{N}\left(y_n \,|\, \mathbf{x}_n^T \boldsymbol{\theta},\, \sigma^2\right)$$

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{i=0}^{N} \log p(y_n \,|\, \mathbf{x}_n, \boldsymbol{\theta}) \xrightarrow{\text{MLE}} \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$$

Information Theory View

Pick a color.

Is it in the lower half? (Y/N)
Is it on the right side of the upper half? (Y/N)
Is it on the left side of the second quadrant? (Y/N)
Is it in the upper half of the right side of the second quadrant? (Y/N)

무슨 색? : 정보가 없다면, 하나를 찍었을 때 맞힐 확률 $= \frac{1}{16}$ $\longrightarrow$ $2^4 = 16$ $\longrightarrow$ $W = 2^n$, $n = \log_2 W$

답 : 예 / 아니오

정확히 맞히기 위한 질문 수는? : 4개

Information Theory View

정보의 양

$$n = \log_2 W$$

$$\frac{1}{W} = P(x)$$

$$I(x) = -\log P(x)$$: Shannon entropy

동전 던질 때 앞면이 나오는 정보

$$-\log_2 0.5 = 1$$

주사위 던질 때 1이 나오는 정보

$$-\log_2 \frac{1}{6} \approx 2.5849$$

Information Theory View

교차 엔트로피(cross entropy)란?

$$H(P, Q) = E_{X \sim P}[-\log Q(x)] = -\sum_{x} P(x) \log Q(x)$$

P를 데이터의 분포라고 하고, Q를 모델에 의해 예측된 결과 분포라고 하자

→ 교차 엔트로피는 negative log-likelihood의 기대값이 된다!

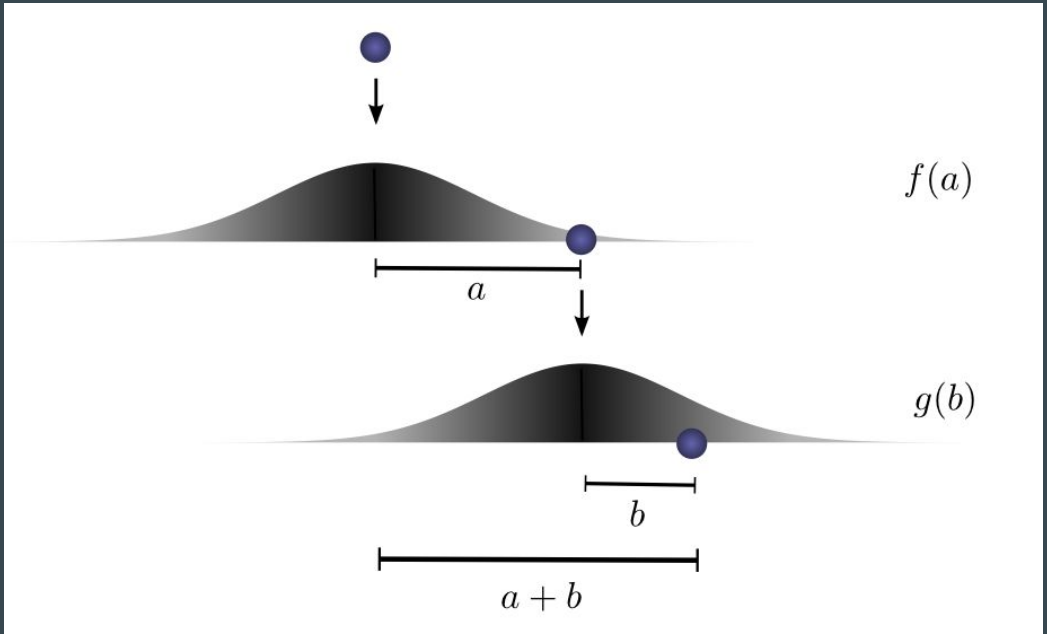
$$-\sum_{x} P(y \mid x) \log P(y \mid x, \theta)$$

간단한 0, 1 이진 분류 문제에서

$$-P(x) \log Q(x) = -\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \log 0 \\ \log 1 \end{pmatrix} = -(-\infty + 0) = \infty$$

$$-P(x) \log Q(x) = -\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \log 1 \\ \log 0 \end{pmatrix} = -(0 + 0) = 0$$

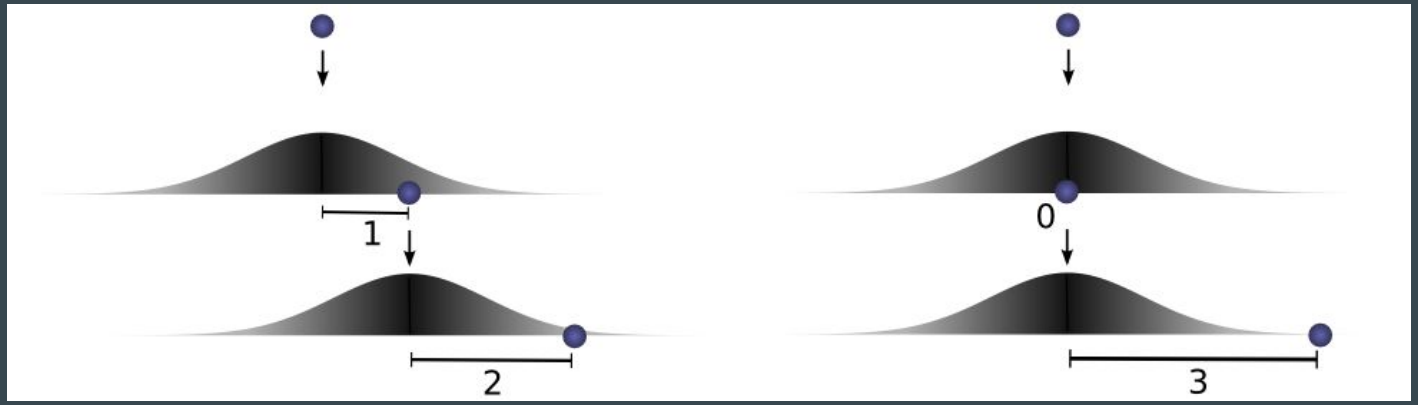
Concept of Convolution



공을 떨어뜨리는 상황을 가정

- 최종 도달 거리 c를 a+b로 두자
- 그러면 공이 c에 도착할 확률은 $f(a) \cdot g(b)$
- 만약 c가 고정된 값이라면?

Concept of Convolution



c를 3이라고 두면, c에 도착할 수 있는 경우의 수는 다양

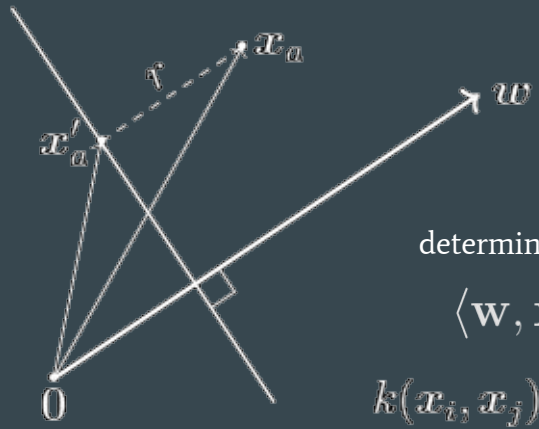
$$f(0)\cdot g(3)+f(1)\cdot g(2)+f(2)\cdot g(1)\cdots=\sum_{a+b=c} f(a)\cdot g(b)$$

b = c - a 로 두면 $\sum_{a} f(a)\cdot g(c-a)=(f\otimes g)(c)$

further reading : https://colah.github.io/posts/2014-07-Understanding-Convolutions/

Kernel Trick of Support Vector Machine(SVM)

Normal SVM review



determine hyperplane

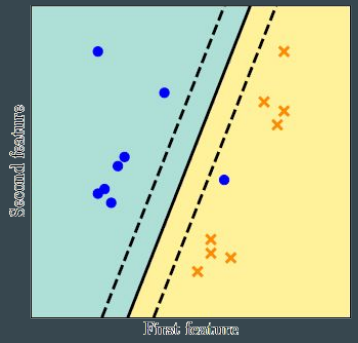
$$\langle \mathbf{w}, \mathbf{x} \rangle + b$$

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

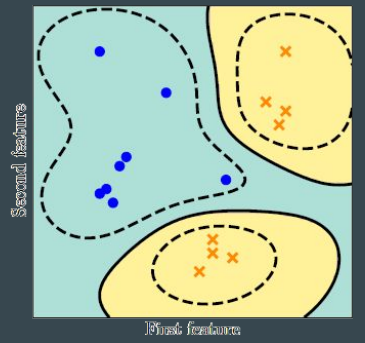
Nonlinear Kernel Map on Hilbert Space

모든 데이터에 함수를 적용하여 Support Vector와 내적까지 해 줘야 함

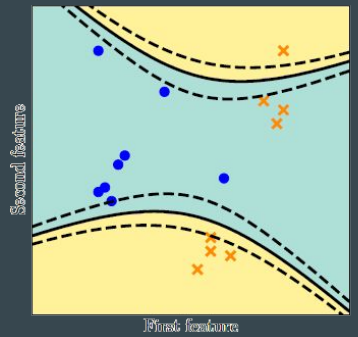
: 계산량이 기하급수적으로 증가 → Trick이 필요



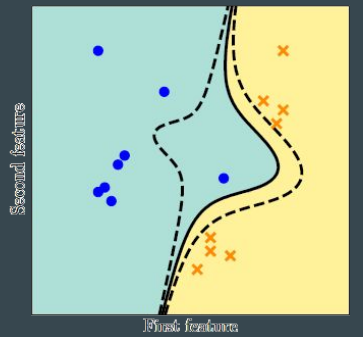
linear kernel



radial basis function kernel



degree 2 polynomial kernel



degree 3 polynomial kernel

Kernel Trick of Support Vector Machine(SVM)

SVM은 최대-최소 문제

convex이기 때문에 dual 문제를 만들 수 있음

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle - \sum_{i=1}^{N} \alpha_i$$

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

kernel $\quad \widetilde{\mathbf{x}}_i := \Phi(\mathbf{x}_i) \longrightarrow \sum_{i,j} \widetilde{\alpha}_i \widetilde{\alpha}_j y_i y_j \left\langle \widetilde{\mathbf{x}}_i, \widetilde{\mathbf{x}}_j \right\rangle$

$$\widetilde{\mathbf{w}} = \sum_{i=1}^{N} \widetilde{\alpha}_i y_i \widetilde{\mathbf{x}}_i \qquad \widetilde{f}(\widetilde{\mathbf{x}}) = \langle \widetilde{\mathbf{w}}, \widetilde{\mathbf{x}} \rangle + \widetilde{b} = \left\langle \sum_{i=1}^{N} \widetilde{\alpha}_i y_i \widetilde{\mathbf{x}}_i, \widetilde{\mathbf{x}} \right\rangle + \widetilde{b}$$

Kernel Trick of Support Vector Machine(SVM)

kernel $\quad \widetilde{\mathbf{x}}_i := \Phi(\mathbf{x}_i) \longrightarrow \sum_{i,j} \widetilde{\alpha}_i \widetilde{\alpha}_j y_i y_j \langle \widetilde{\mathbf{x}}_i, \widetilde{\mathbf{x}}_j \rangle$

$$\widetilde{\mathbf{w}} = \sum_{i=1}^{N} \widetilde{\alpha}_i y_i \widetilde{\mathbf{x}}_i \qquad \widetilde{f}(\widetilde{\mathbf{x}}) = \langle \widetilde{\mathbf{w}}, \widetilde{\mathbf{x}} \rangle + \widetilde{b} = \left\langle \sum_{i=1}^{N} \widetilde{\alpha}_i y_i \widetilde{\mathbf{x}}_i, \widetilde{\mathbf{x}} \right\rangle + \widetilde{b}$$

linear mapping은 텐서 연산으로 표현 가능

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$$

$$\Phi(\mathbf{x}) = A\mathbf{x} \longrightarrow K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T A^T A \mathbf{x}_j$$

어떤 함수를 적용할 지만 결정하면 inner product 계산만 하면 됨

: kernel trick