PRML과 부수적인 자료들을 공부하여 간단히 정리하였습니다.

1.1 Example : polynomial curve fitting

- 예를 들어, 회귀에서 error function이 quadratic function of w이면 w에 대한 미분은 w에 linear하고 unique한 closed form의 해를 구할 수 있다.
- 모델의 overfitting을 항상 조심하고 데이터의 수가 늘어날수록 그 정도는 약해진다. MLE 방법은 overfitting에 취약하며 Bayesian 모델링으로 보완할 수 있다.
- ridge와 같이 error function에 패널티항을 추가하여 overfitting을 막는 방법도 있다. 이를 shrinkage 방법이라 부른다. (딥러닝에서는 weight decay)
- 이런 모델의 복잡한 정도를 정하는 데에 validation data set을 만들기도 하는데 이는 다소 낭비이므로 다른 방법을 공부할 것이다. (아마 Bayesian approach일듯)

1.2 Probability Theory

패턴인식에서 가장 중요한 컨셉은 uncertainty이다. Probability Theory는 이런 uncertainty을 quantification하고 manipulation하는 방법을 제시한다. (확률을 이용하여)

- The rules of Probability
  - sum rule : $p(X) = \sum_Y p(X, Y)$
  - product rule : $p(X, Y) = p(Y/X)p(X)$
- Bayes' Throrem (rule)
  - $p(Y/X) = \frac{p(X/Y)p(Y)}{p(X)}$
  - $p(Y)$: prior probability
  - $p(Y/X)$: posterior probability

1.2.1 Probability densities

- probability density : if the probability of a real-valued variable x falling in the interval $(x, x + \delta x)$ is given by $p(x)\delta x$ for $\delta x \to 0$, then $p(x)$ is called the *probability density*
  - 값은 항상 0 이상, 합하면 1을 가진다.

1.2.2 Expectations and covariances

- expection of $f(x)$: $E[f(x)] = \int p(x)f(x)dx$
  - it can be approximated as

$$E[f] \approx \frac{1}{N}\sum_{n=1}^{N} f(x_n)$$

- $E_x[f(x, y)]$는 y에 대한 함수이다. x에 대해 averaged over 된 것이다.
- conditional expection : $E[f(x)/y] = \int p(x/y)f(x)dx$
- variance of $f(x)$: $var[f] = E[(f(x) - E[f(x)])^2]$
  - $f(x)$가 mean 주위에서 얼마나 variability가 있는지 보여준다.

1.2.3 Bayesian probabilities

우리가 일반적으로 알고 있는 확률(probability)은 frequentist의 견해이다. bayesian은 frequentist와는 아예 다른 접근법을 갖는다.

- Frequentist
  - 분모가 되는 전체 사건이 무한대로 일어나고 우리가 궁금한 사건이 그 중 몇번 일어나는지를 확률로 생각한다.
  - parameter 추정이 목표이며 parameter는 fixed 되어 있다고 생각한다.

    - 주로 estimator로서 likelihood function을 최대화하는 MLE로 사용한다.
- Bayesian
    - 확률 : uncertainty를 quantification한 것으로 생각한다.
    - parameter는 fixed 된 것이 아니라 (probability) distribution을 갖는 것이라고 생각한다.
    - posterior distribution을 찾는 것이 목표이다.
- Bayes' theorem

$$p(\mathbf{w}/D) = \frac{p(D/\mathbf{w})p(\mathbf{w})}{p(D)}$$

    - parameter에 대해 원래 갖고 있던 믿음을 data D에 대한 정보를 얻은 뒤에 posterior probability로 업데이트 한다. (분모는 posterior가 합이 1이 되기 위한 normalization constant)
    - prior probability : $p(w)$
    - likelihood function : $p(D/w)$
    - posterior probability : $p(w/D)$
    - posterior $\propto$ likelihood * prior

1.2.4 The Gaussian distribution

$$N(x/\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$$

- mean : μ
- variance : σ^2
- standard deviation : σ
- precision : $\beta = 1/\sigma^2$
- normal (gaussian) 분포는 mode와 mean이 같다.

i.i.d (independent and identically distributed : data point가 독립적이고 같은 분포에서 나왔다) 인 경우, likelihood function은

$$p(\mathbf{x}/\mu, \sigma^2) = \prod_{n=1}^{N} N(x_n/\mu, \sigma^2)$$

이고 이를 최대화하는 mean과 variance의 MLE는 sample mean, sample variance이다. MLE를 구하는 방법은 likelihood function에 log를 취한 후 미분하여 0을 만족하는 parameter를 찾으면 된다. 이때 단점은 maximum likelihood 접근법이 분포의 variance를 underestimate한다(bias 발생)는 점이다. N이 커지면 문제가 없지만 복잡한 모델에서는 이런 bias때문에 문제가 발생할 수 있다. (나중에 공부한다)

1.2.5 Curve fitting re-visted

data를 통해 polynomial curve를 fitting해보자. target t에 대한 uncertainty를 probability를 통해 표현하면 (under gaussian noise distribution)

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

위의 식을 이용하여 우리는 parameter $\mathbf{w}$ 추정한다. likelihood를 최대로 하는 MLE를 찾으면 되는 것이다. log likelihood function은 아래와 같은 모양을 갖는다.

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

위 값을 최대화 하는 $\mathbf{w}_{ML}$을 찾으면 된다. 이는 결국 least square 방법과 동일한 의미를 갖는다. (추정선과 target의 차이를 최소화해야되므로) parameter를 추정한 뒤에 이제 prediction을 해야한다. 우리는 확률모델을 갖고 있기에 t에 대한 point estimate 만이 아니라 predictive distribution을 만들수 있다.

$$p(t/x, \mathbf{w}_{ML}, \beta_{ML}) = N(t/y(x, \mathbf{w}_{ML}), \beta_{ML}^{-1})$$

지금까지는 frequentist의 영역이였다면 Bayesian들은 어떤 접근을 하는지 살펴보자. 일단 우리가 추정해야하는 parameter에 대한 prior를 갖고 있다. prior distribution를 gaussian 분포로 가정하면 아래와 같이 나타낼 수 있다. (Mth order의 polynomial)

$$p(\mathbf{w}/\alpha) = N(\mathbf{w}/\mathbf{0}, \alpha^{-1}\mathbf{I}) = (\frac{\alpha}{2\pi})^{(M+1)/2} \exp\{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\}$$

이를 통해 우리는 posterior distribution를 구할 수 있다. posterior는 likelihood와 prior의 곱에 비례하므로

$$p(\mathbf{w}/\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}/\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}/\alpha)$$

위의 posterior distribution을 최대화로 만드는 parameter를 *MAP* (MLE에 대응되는 point estimate)라고 부른다. posterior distribution에 negative log를 취하면 posterior를 최대로 만드는 것은 아래를 최소화 하는 것과 같다.

$$\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}$$

위 결과를 통해 posterior distribution를 maximizing하는 것은 regularized sum-of-squares error function을 minimizing하는 것과 동일하다는 것을 알 수 있다.. (L2 regularization, Ridge regression)

1.2.6 Bayesian curve fitting

위의 Bayesian 접근법은 point estimate를 구했기 때문에 살짝 아쉽다. 좀 더 Bayesian적인 방법은 $\mathbf{w}$의 모든 값에 대해 integral over하는 것이다. $\mathbf{w}$에 대해 marginalize하면 되는데 이는 뒤에 자주 나오는 방법이므로 잘 기억하자. 이제 predictive dist를 구해보자.

- training data : $\mathbf{x}, \mathbf{t}$
- new data : x
- hyperparameter (assume we know) : α, β (아래식에서는 생략)
- $\mu(x), s^2(x)$는 뒤에 나온다.

$$p(t/x, \mathbf{x}, \mathbf{t}) = \int p(t/x, \mathbf{w})p(\mathbf{w}/\mathbf{x}, \mathbf{t})d\mathbf{w} = N(t/\mu(x), s^2(x))$$

$$\mu(x) = \beta\phi(x)^T\mathbf{S}\sum_{n=1}^{N}\phi(x_n)t_n$$

$$s^2(x) = \beta^{-1} + \phi(x)^T\mathbf{S}\phi(x)$$

$$\mathbf{S}^{-1} = \alpha\mathbf{I} + \beta\sum_{n=1}^{N}\phi(x_n)\phi(x)^T$$

- vector $\phi(x)$: element $\phi_i(x) = x^i$ for $i = 0, ...M$

1.3 Model Selection

여러 가지 모델을 선택할 때, train score로 model을 선택하는 것은 적절하지 않다. 그래서 validation set을 이용한다. 하지만 validation set에 overfitting하는 경우도 있기에 test set으로 최종 점검까지 하는 것이다. data가 제한적인 경우 cross validation의 방법을 사용한다. 하지만 이는 상당히 computationally expensive하다.

1.4 The Curse of Dimensionalty

차원의 저주를 보여주는 몇가지 예시들

- nearest neighborhood 알고리즘에 해당하는 부분 : sample space를 cubic형태로 나눈다고 생각했을 때, 차원이 커짐에 따라 지수적으로 cubic의 갯수가 많아진다. 따라서 cubic에 data가 텅 비지 않으려면 많은 양의 데이터가 필요하다.
- polynomial 의 경우 : Mth order의 polynomial 모델을 사용한다고 하면 D^M 으로 parameter의 수가 증가한다.

- data를 sphere하게 생각해보자. 차원이 높아질수록 sphere의 표면쪽에 data가 몰려있다. 즉, 중심쪽이 sparse해지는 것이다.

1.5 Decision Theory

Decision Theory는 크게 두 가지의 과정으로 이루어져 있다.

- Determining $p(x,t)$ from a training data set : **inference**
- 이를 통하여 새로운 데이터에 대해 결정(분류,회귀) : **decision**

Decision Theory의 목표는 적절한 Probability들을 이용하여 optimal한 decision을 내리는 것이다. 2-class classification의 상황을 예시로 뒤의 내용을 진행한다. 우리는 input data를 통해 해당 data의 class를 구분하고 싶기에 $p(C_k/\mathbf{x})$를 구해야 한다. Bayes' theorem을 생각해보면 posterior를 구해야 하는 것이다.

$$p(C_k/\mathbf{x}) = \frac{p(\mathbf{x}/C_k)p(C_k)}{p(\mathbf{x})}$$

우리는 misclassfication을 최소화하기 위해서 둘 중 더 큰 posterior probability갖는 class에 input data를 분류한다.

1.5.1 Minimizing the misclassfication rate

우리의 목표가 misclassfication을 최소화하는게 목표라고 하자. 각 $\mathbf{x}$를 class에 분류해야하고 이를 위해 rule이 필요하다. 그 rule에 따라 input space를 region R_k로 나눠야 한다. 이 region을 *decision regions* 라고 한다. (R_k의 data는 class k라고 분류) decision region간의 경계선을 *decision boundary*, *decision surface* 라고 부른다. misclassfication의 확률은

$$p(mistake) = P(\mathbf{x} \in R_1, C_2) + P(\mathbf{x} \in R_2, C_1) = \int_{R_1} p(\mathbf{x}, C_2)d\mathbf{x} + \int_{R_2} p(\mathbf{x}, C_1)d\mathbf{x}$$

mistake의 확률을 최소화하기 위해서는 각 integral의 값을 최소화해야 한다. 따라서 만약 $p(\mathbf{x}, C_1) > p(\mathbf{x}, C_2)$의 경우, data를 class1으로 분류해야한다. $p(\mathbf{x}, C_k) = p(\mathbf{x})p(C_k/\mathbf{x})$ 이고 우변의 $p(\mathbf{x})$는 공통된 부분이므로 우리는 $p(C_k/\mathbf{x})$만 고려하면 된다.

1.5.2 Minimizing the expected loss

위에서 misclassfication rate를 줄이는 부분에 대해서 살펴봤다. 하지만 실제로 분류를 할 때는 이 접근으로는 부족하다. 예를 들어, 암환자를 분류하는 문제라고 생각해보자. 암이 걸리지 않은 환자를 걸렸다고 잘못 판단하는 것과 암이 걸렸는데 걸리지 않았다고 판단하는 것. 둘 중 후자가 훨씬 심각한 문제이다. 이런 경우 후자에 대해 더 가중치가 있어야 하지 않을까?

- *loss function (cost function)* : overall measure of loss incurred in taking any of the available decisions or actions
- L_{kj} : (k인데 j로 분류한 경우) loss matrix의 element를 의미한다. misclassfication에 대한 loss라고 이해하면 된다. 예를 들면 암환자의 loss matrix는 아래와 같은 모양이다.

$$\begin{bmatrix} 0 & 100 \\ 1 & 0 \end{bmatrix}$$

(inference가 끝난 뒤에 decision하는 과정에 해당) optimal solution은 loss function을 최소화하는 것이다. 하지만 loss function은 true class을 알아야 계산할 수 있다. 우리는 true class를 모른. (예를 들어, 환자의 신상데이터가 있고 이를 통해 암환자인지 아닌지 찾아야하는 상황) 따라서 우리는 average loss를 최소화하는 방법을 선택한다.

$$E[L] = \sum_k \sum_j \int_{R_j} L_{kj} p(\mathbf{x}, C_k)d\mathbf{x}$$

우리의 목표는 expected loss를 최소로 만드는 적절한 R_j를 찾는 것이고 이는 각 데이터 $\mathbf{x}$가 $\sum_k L_{kj}p(\mathbf{x}, C_k)$를 최소화 한다는 것을 의미한다.

- 최종적으로 expected loss를 최소화 하기 위해서는 $\mathbf{x}$를 값 $\sum_k L_{kj}p(C_k/\mathbf{x})(= E[L(C_k, \hat{C}_k)/X = \mathbf{x}])$ 이 최소가 되는 class j로 분류하는 것이다.

1.5.3 The reject option

class에 따라 posterior의 비교를 통해 결정하기 애매한 상황이 생긴다. 이런 경우에는 probability에 따라 결정하기 보다는 다른 방법을 사용하는 것이 적절할 수도 있다. (예를 들면, 해당 데이터를 model이 아니라 전문가가 판단하는 방법) 이런 경우 *regect option* 이 있다고 할 수 있는 것이다.

1.5.4 Inference and decision

decision 문제를 해결하는 방법을 3가지로 분류할 수 있다. 앞쪽의 방법일수록 복잡한 방법이다.

- **generative model**
  - 아래 식에서 posterior를 구하기 위해서는 분자, 분모를 다 구해야 한다. 한마디로 *approachs that explicitly or implicitly model the distribution of inputs as well as outputs.*
  - 다른 표현으로는, joint distribution $p(\mathbf{x}, C_k)$을 구해서 marginalize하여 분모도 구하여 posterior를 구한다.

$$p(C_k/x) = \frac{p(x/C_k)p(C_k)}{p(x)}$$

- **discriminative model**
  - *approachs that model the posterior probabilities directly*
  - 예를 들면 SVM, Tree models, KNN 등등
- **discriminative function**
  - *maps each input x directly onto a class label*
  - 따라서 확률을 고려하지 않는다.
  - inference와 decision stage를 하나로 묶은 것이다.

각각 장단점이 존재한다. 예를 들면, 번에서 prior $p(\mathbf{x})$를 구했으므로 해당 값이 너무 작은 새로운 data는 무시하는 판단을 할 수 있다. (outlier detection하는 것처럼) 하지만 주로 최소한 posterior dist는 구한다. (1,2번 선호) posterior를 구하면 어떤 장점이 있는지 알아보자.

- Minimizing risk
  - 이전에 봤던 loss matrix를 수정하여 decision criterion을 수정하기 쉽다.
- Reject option
  - expected loss뿐만 아니라 misclassfication rate를 최소화 하는 rejection criterion을 정할 수 있게 해준다.
- Compensating for class priors
  - posterior는 prior에 비례하므로 prior를 적절하게 바꿔줌으로서 posterior를 보완할 수 있다.
- Combing models
  - 특정 문제를 subproblem으로 나누어서 생각할 수 있다. 예를 들면, naive bayes model과 같이 independent를 이용하여 posterior를 나누어서 생각할 수 있는 장점이 생긴다.

1.5.5 Loss functions for regression

이전까지 classification에 대해 살펴봤으므로 이번에는 regression에 대해 살펴보자. expected or average loss는

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

이다. regression에서 주로 사용하는 loss function은 squared loss이고 이를 통해 다시 쓰면

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

이다. 우리는 이를 최소화하는 $y(\mathbf{x})$를 찾는 것이 목표이므로 미분하여 구할 수 있다.

$$\frac{dE[L]}{dy(\mathbf{x})} = 2\int \{y(\mathbf{x}) - t\}p(\mathbf{x}, t)dt = 0$$

$$y(\mathbf{x}) = \frac{\int tp(\mathbf{x}, t)dt}{p(\mathbf{x})} = \int tp(t/\mathbf{x})dt = E_t[t/\mathbf{x}]$$

이는 우리가 알고 있는 regression function의 모양이다. (conditional average of t conditioned on x) 이를 이용하여 추가적인 접근을 해보자면

$$\{y(\mathbf{x}) - t\}^2 = \{y(\mathbf{x}) - E[t/\mathbf{x}] + E[t/\mathbf{x}] - t\}^2$$

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

두번째 항은 variance of the distribution of t, averaged over x 이다. 따라서 이는 irreducible minumum value of the loss function 을 의미한다.

+ Decision Theory 추가

monk의 강의에서 decision theory를 다루는데 해당 내용을 추가하고자 한다.

- 일단, loss function은 "0-1 loss" 으로 생각하자.
  - true = prediction : 0
  - true != prediction : 1

두 가지 상황으로 나누어서 살펴보자. 하지만 두 경우 모두의 공통적인 결론은

- $p(y/x)$ 가 핵심이라는 것이다.

1. Minimizing conditional expected loss

: Given x, minimize $L(y, \hat{y})$ ... but don't know true class y

- $(X, Y) \sim P$: discrete

$$E[L(Y, \hat{y})/X = x] = \sum_y L(y, \hat{y})P(y/x) = \sum_{y \neq \hat{y}} 1 * P(y/x) = 1 - P(\hat{y}/x)$$

$$\therefore \hat{y} = argmin_y E[L(Y, \hat{y})/x] = argmax_y P(y/x)$$

2. Choosing f to minimize expected loss

: Choose $f(f(x) = y)$ to minimize $L(y, f(x))$ but don't know x or y

$$E[L(Y, \hat{Y})] = E[L(Y, f(X))] = \sum_{x,y} L(y, f(x))P(x, y)$$

$$= \sum_x \{\sum_y L(y, f(x))P(y/x)\}P(x) = \sum_x g(x, f(x))p(x) = E_x[g(x, f(x))]$$

- suppose for some x', t
  - $g(x', f(x')) \geq g(x', t)$
- $f_0(x) =$
  - if $x \neq x', f(x)$
  - if $x = x', t$
- 모든 $x,\ g(x, f(x)) \geq g(x, f_0(x))$

$$\therefore E_x[g(x, f(x))] \geq E_x[g(x, f_0(x))]$$

Choose f to min $g(x, f(x))$

$$f(x) = argmin_t g(x,t)$$

Big picture

- ## Generative model
  - estimate $p(x,y)$ using data
  - and then $p(y/x) = \frac{p(x,y)}{p(x)}$
- parameter / latent : θ 라고 하자
  - θ는 distribution에 관한 parameter / latent
  - D는 random (new data)

$$p(y/x, D) = \int p(y/x, D, \theta) p(\theta/x, D) d\theta$$

- $p(y/x, D)$: predictive distribution
- $p(\theta/x, D)$: posterior distribution

$p(y/x, D, \theta)$ 이 부분은 주로 closed form(eg. regression y=wx)으로 구해지며 어렵지 않다. 하지만 posterior 부분은 closed form으로 못 구하는 경우가 많다. 또한 integral 부분도 계산이 어려운 경우가 많다. 그렇다면 이를 어떻게 해결할까? 크게 4가지의 방법을 살펴보자.

- exact inference
  - Multivariate Gaussian, Conjugate prior, Graphical model
- point estimate
  - MLE, MAP (1.2.5를 보면 integral 없이 계산)
  - optimization, EM
- deteministic approximation
  - Laplace approximation, Variational method, Expectation propagation
- stochastic approximation
  - Sampling 기법들 (eg. MCMC)