ML IN PROBABILISTIC PERSPECTIVE

Week 3. Linear Models for Classification

오태환

ESC, YONSEI UNIVERSITY

April 15, 2020

Table of Contents

Overview

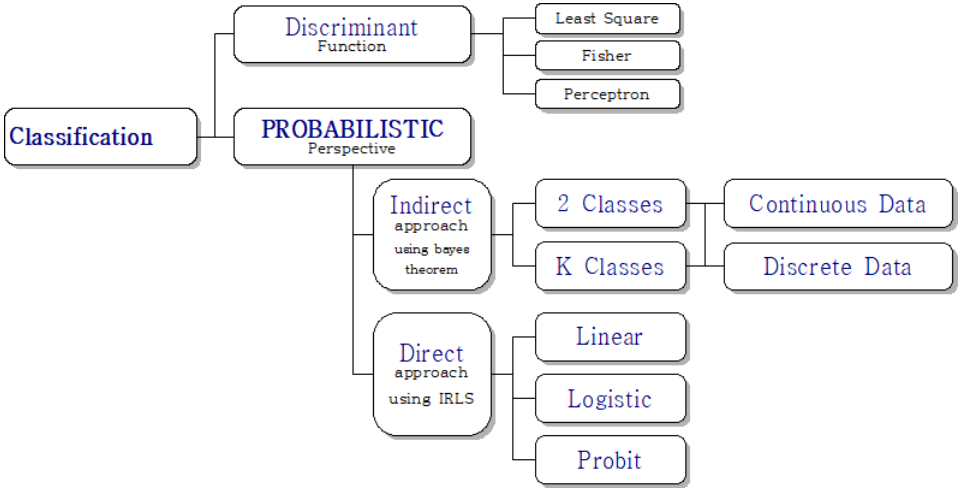

Classification
Discriminant Function
Least Square
Fisher
Perceptron
PROBABILISTIC Perspective
Indirect approach using bayes theorem
2 Classes
K Classes
Continuous Data
Discrete Data
Direct approach using IRLS
Linear
Logistic
Probit

Table of Contents

Discriminant Function

- Least Squares

$E_D(\tilde{W}) = \frac{1}{2}\left\{(X\tilde{W} - T)^T(X\tilde{W} - T)\right\}$ 를 최소화하는 $\tilde{W}$를 구하는 것

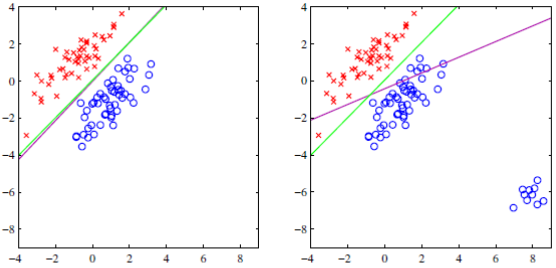


Figure 4.4 The left plot shows data from two classes, denoted by red crosses and blue circles, together with the decision boundary found by least squares (magenta curve) and also by the logistic regression model (green curve), which is discussed later in Section 4.3.2. The right-hand plot shows the corresponding results obtained when extra data points are added at the bottom left of the diagram, showing that least squares is highly sensitive to outliers, unlike logistic regression.

Outlier에 민감하다

Discriminant Function

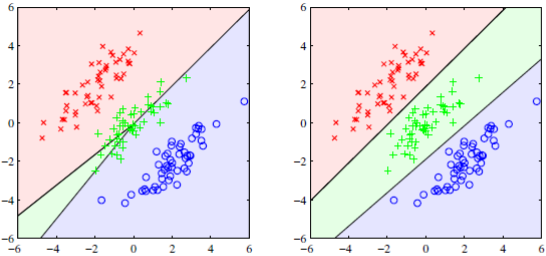


Figure 4.5 Example of a synthetic data set comprising three classes, with training data points denoted in red (×), green (+), and blue (○). Lines denote the decision boundaries, and the background colours denote the respective classes of the decision regions. On the left is the result of using a least-squares discriminant. We see that the region of input space assigned to the green class is too small and so most of the points from this class are misclassified. On the right is the result of using logistic regressions as described in Section 4.3.2 showing correct classification of the training data.

3개 이상의 클래스에선 잘 맞지 않는다

Discriminant Function

- Fisher's

S_B(*between-class covariance*)와 S_W(*within-class covariance*)를 모두 고려(class overlap을 막기 위함)

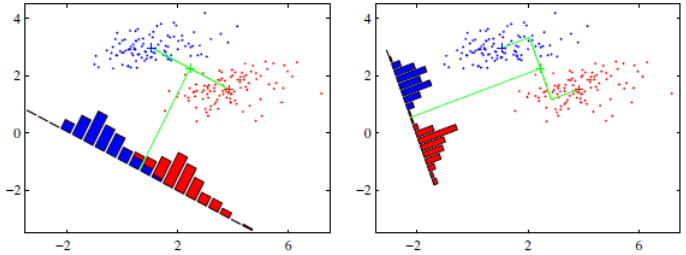


Figure 4.6 The left plot shows samples from two classes (depicted in red and blue) along with the histograms resulting from projection onto the line joining the class means. Note that there is considerable class overlap in the projected space. The right plot shows the corresponding projection based on the Fisher linear discriminant, showing the greatly improved class separation.

왼쪽은 S_B만 고려, 오른쪽은 S_B와 S_W 둘 다 고려. 즉 오른쪽이 FISHER

Discriminant Function

- Perceptron

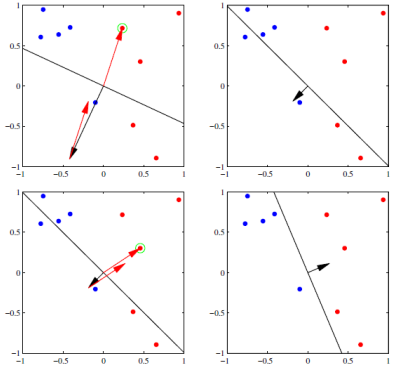


Figure 4.7 Illustration of the convergence of the perceptron learning algorithm, showing data points from two classes (red and blue) in a two-dimensional feature space (ϕ_1, ϕ_2). The top left plot shows the initial parameter vector $\mathbf{w}$ shown as a black arrow together with the corresponding decision boundary (black line), in which the arrow points towards the decision region which classified as belonging to the red class. The data point circled in green is misclassified and so its feature vector is added to the current weight vector, giving the new decision boundary shown in the top right plot. The bottom left plot shows the next misclassified point to be considered, indicated by the green circle, and its feature vector is again added to the weight vector giving the decision boundary shown in the bottom right plot for which all data points are correctly classified.

Table of Contents

Goal of Probabilistic Perspective Classification

★우리의 목표★

$P(C_k|x)$를 구하는 것!

- Indirect Approach

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}.$$

*Bayes theorem*을 이용해 간접적으로 $P(C_k|x)$를 구해보자

- Direct Approach

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \mathbf{H}^{-1}\nabla E(\mathbf{w}).$$

IRLS를 이용해 직접 $P(C_k|x)$를 구해보자

Indirect Approach

★우리의 목표★

$P(C_k|x)$를 구하는 것!

- 2 Classes

두 클래스 중 첫 번째 클래스를 생각해보자.

$p(C_1|\mathbf{x}) =$

Indirect Approach

- K Classes

좀 더 일반적인 상황을 알아보자. k클래스를 생각하면

$p(C_k|\mathbf{x})=$

Indirect Approach

- Continuous Data
- 1st Step : Get W & W_0 by Class-Conditional Densities $p(\mathbf{x}|C_k)$

2Classes

$p(\mathbf{x}|C_k)$가 Gaussian분포이고, 모든 class의 $p(\mathbf{x}|C_k)$ 같은 covariance matrix를 공유한다고 가정하자. 그러면 k class의 pdf를 다음과 같이 만들 수 있다.

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}.$$

위의 logistic sigmoid를 생각해보면, $p(C_1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + w_0)$이다. 따라서 a에 위의 pdf를 대입하여 다음과 같은 $\mathbf{w}$와 w_0를 구할 수 있다.

$$\begin{aligned} \mathrm{w} &= \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \\ w_0 &= -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}. \end{aligned}$$

여기에 적절한 $p(C_1)$, $p(C_2)$, μ_1, μ_2, Σ를 구해 대입하면 w, w_0를 구할 수 있다!

Indirect Approach

- 1st Step : Get w & w_0 by Class-Conditional Densities $p(x|C_k)$

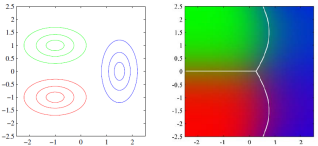
K Classes

$$a_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

where we have defined

$$\begin{aligned} \mathbf{w}_k &= \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k \\ w_{k0} &= -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(\mathcal{C}_k). \end{aligned}$$

여기서 우린 Σ가 모든 클래스에서 같다고 가정했다. 그렇기 때문에 Linear Decision Boundary를 가진다. 하지만 Σ가 클래스마다 다르면 Quadratic Decision Boundary를 가진다.



Indirect Approach

- 2nd Step : Get $p(C_1)$, $p(C_2)$, μ_1, μ_2, Σ by Maximum Likelihood

위의 Gaussian, shared covariance 가정에 더해, $\{\mathbf{x}_n, t_n\}$의 dataset을 가지고 있다고 하자.($n = 1, ..., N$, if x in C_1 ,$t_n = 1$, if in C_2 ,$t_n = 0$). 또한 $p(C_1) = \pi$, $p(C_2) = 1 - \pi$라고 하자.
그렇다면 C_1에서 온 $\mathbf{x}_n$에선 t_n=1이고 다음과 같은 분포를 가진다.

$$p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\mathbf{x}_n|\mathcal{C}_1) = \pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}).$$

또한 C_2에서 온 $\mathbf{x}_n$에선 $t_n = 0$이고 다음과 같다.

$$p(\mathbf{x}_n, \mathcal{C}_2) = p(\mathcal{C}_2)p(\mathbf{x}_n|\mathcal{C}_2) = (1 - \pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}).$$

따라서 다음과 같은 likelihood function을 얻을 수 있다.

$$p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \left[\pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})\right]^{t_n} \left[(1 - \pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})\right]^{1-t_n}$$

where $t = (t_1, t_2, ..., t_N)^T$. 여기서 $p(C_1)$, $p(C_2)$, μ_1, μ_2, Σ들을 각각 편미분해 각각의 MLE를 얻는다.

Indirect Approach

- 2nd Step : Get $p(C_1)$, $p(C_2)$, μ_1, μ_2, Σ by Maximum Likelihood (2 Classes) 자 이제 각각의 MLE를 구해보자.
- $p(C_k)$ 구하기

위에서 구한 likelihood에 log를 씌우고 π에 대해 편미분하면

$$\sum_{n=1}^{N} \{t_n \ln \pi + (1 - t_n) \ln(1 - \pi)\} . \tag{4.72}$$

Setting the derivative with respect to π equal to zero and rearranging, we obtain

$$\pi = \frac{1}{N} \sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2} \tag{4.73}$$

상당히 직관적이게 $p(C_k)$는 전체 N중 C_1의 비율로 나타난다.

Indirect Approach

- 2nd Step : Get $p(C_1)$, $p(C_2)$, μ_1, μ_2, Σ by Maximum Likelihood (2 Classes)
- μ_k 구하기

μ_k에 대해 편미분 한 후, 0이라고 두면

Setting the derivative with respect to $\boldsymbol{\mu}_1$ to zero and rearranging, we obtain

$$\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n \mathbf{x}_n \tag{4.75}$$

which is simply the mean of all the input vectors $\mathbf{x}_n$ assigned to class $\mathcal{C}_1$. By a similar argument, the corresponding result for $\boldsymbol{\mu}_2$ is given by

$$\boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1 - t_n) \mathbf{x}_n \tag{4.76}$$

which again is the mean of all the input vectors $\mathbf{x}_n$ assigned to class $\mathcal{C}_2$.

역시 상당히 직관적이게 μ_k는 C_k클래스에 속한 **x**의 평균으로 나타난다.

Indirect Approach

- 2nd Step : Get $p(C_1)$, $p(C_2)$, μ_1, μ_2, Σ by Maximum Likelihood (2 Classes)
- Σ 구하기

일단 위의 loglikelihood에서 Σ에 대한 terms만 뽑아내고 정리해보자

$$
\begin{aligned}
&-\frac{1}{2}\sum_{n=1}^{N} t_n \ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N} t_n (\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_1) \\
&-\frac{1}{2}\sum_{n=1}^{N} (1-t_n) \ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N} (1-t_n) (\mathbf{x}_n - \boldsymbol{\mu}_2)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_2) \\
&= -\frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{N}{2}\mathrm{Tr}\left\{\boldsymbol{\Sigma}^{-1}\mathbf{S}\right\} \qquad (4.77)
\end{aligned}
$$

where we have defined

$$
\begin{aligned}
\mathbf{S} &= \frac{N_1}{N}\mathbf{S}_1 + \frac{N_2}{N}\mathbf{S}_2 && (4.78)\\
\mathbf{S}_1 &= \frac{1}{N_1}\sum_{n\in\mathcal{C}_1} (\mathbf{x}_n - \boldsymbol{\mu}_1)(\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}} && (4.79)\\
\mathbf{S}_2 &= \frac{1}{N_2}\sum_{n\in\mathcal{C}_2} (\mathbf{x}_n - \boldsymbol{\mu}_2)(\mathbf{x}_n - \boldsymbol{\mu}_2)^{\mathrm{T}}. && (4.80)
\end{aligned}
$$

그리고 전과 같이 Σ에 대해 편미분 후 0으로 두면, $\Sigma = S$가 나온다.
이것을 직관적으로 보면 각 클래스 내 **x**의 분산을 각자의 비율로 가중평균한 값이다.
이제 $p(C_1)$, $p(C_2)$, μ_1, μ_2, Σ를 알았으니, 위의 w, w_0식에 대입해 w, w_0를 구하자!

Indirect Approach

- Discrete Data

교재에서도 간단하게 소개만 하므로 간단하게 넘어가겠다.

Let us now consider the case of discrete feature values x_i. For simplicity, we begin by looking at binary feature values $x_i \in \{0, 1\}$ and discuss the extension to more general discrete features shortly. If there are D inputs, then a general distribution would correspond to a table of 2^D numbers for each class, containing $2^D - 1$ independent variables (due to the summation constraint). Because this grows exponentially with the number of features, we might seek a more restricted representation. Here we will make the *naive Bayes* assumption in which the feature values are treated as independent, conditioned on the class $\mathcal{C}_k$. Thus we have class-conditional distributions of the form

$$p(\mathbf{x}|\mathcal{C}_k) = \prod_{i=1}^{D} \mu_{ki}^{x_i}(1 - \mu_{ki})^{1-x_i} \tag{4.81}$$

which contain D independent parameters for each class. Substituting into (4.63) then gives

$$a_k(\mathbf{x}) = \sum_{i=1}^{D} \{x_i \ln \mu_{ki} + (1 - x_i) \ln(1 - \mu_{ki})\} + \ln p(\mathcal{C}_k) \tag{4.82}$$

which again are linear functions of the input values x_i. For the case of $K = 2$ classes, we can alternatively consider the logistic sigmoid formulation given by (4.57). Analogous results are obtained for discrete variables each of which can take $M > 2$ states.

모든 features가 independent하다고 가정하는 것을 *Naive Bayes* assumption이라 한다는 것만 알고 넘어가자.

Direct Approach

★우리의 목표★

$P(C_k|x)$를 구하는 것!

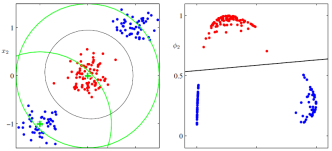
지금까지 본 Bayes Theorem을 통한 Indirect Approach는 다음과 같은 단점이 있다.

- 구해야 하는 parameter의 갯수가 많다
- Class-Conditional density의 분포를 가정하는데, 이 가정이 실제와 다르면 예측 또한 오차가 발생한다.

그렇기 때문에 $P(C_k|x)$의 parameter의 MLE를 직접 구해보도록 하자! 이 때 iterative reweighted least square, 즉 IRLS 알고리즘을 사용하게 된다.

- Fixed basis function

Direct Approach를 진행하기 앞서, 먼저 해야할 것이 있다. 지금까지 우리는 $\mathbf{x}$를 그대로 써왔다. 하지만 다음과 같은 Data에는 변환을 해주는게 좋아보인다. 이 때 우리는 Fixed basis function $\phi(x)$로 변환을 하게 된다.



Direct Approach

- iterative reweighted least square (IRLS)

지난주에, 우리는 linear regression에서 $\mathbf{w}$의 MLE가 하나의 값으로 딱 떨어진다는 것을 배웠다(이는 Gaussian noise에서 w에 대한 log likelihood가 이차식의 형태로 나타나기 때문이다). 하지만 위에서 배운 logistic sigmoid나 softmax함수에선 하나의 값으로 딱 떨어지게 하는 것이 불가능하다(W에 대한 이차식 형태가 아니기 때문). 그렇지만 방법이 있다. 바로 error function을 최소화하는 $\mathbf{w}$의 값을 찾는 것이다. error fuction은 concave이기 때문에 unique minimum이 나올 수 있다. 그리고 이러한 $\mathbf{w}$를 찾는데 효과적인 것이 바로 이전에 배운 gradient decent 방식과 비슷한 *Newton-Raphson* iterative optimization scheme이다.

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \mathbf{H}^{-1}\nabla E(\mathbf{w}).$$

여기서 $E(w)$가 error function이고, H는 Hessian 행렬이라고 하며 $E(w)$의 $\mathbf{w}$에 대한 이계도함수를 원소로 가지고 있는 행렬이다.

Direct Approach

- Linear Regression

3장에서 했던 Linear Regression 에서 $\mathbf{w}$의 MLE를 IRLS방식으로 구해보자.

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (\mathbf{w}^{\mathrm{T}}\phi_n - t_n)\phi_n = \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\mathbf{w} - \mathbf{\Phi}^{\mathrm{T}}\mathbf{t} \tag{4.93}$$

$$\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} \phi_n \phi_n^{\mathrm{T}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} \tag{4.94}$$

여기서 Φ는 whose nth row가 Φ_n^T인 N × M design matrix이다.
이걸 위의 *Newton-Raphson* update에 넣어보자.

$$\begin{aligned} \mathbf{w}^{(\mathrm{new})} &= \mathbf{w}^{(\mathrm{old})} - (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\left\{\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\mathbf{w}^{(\mathrm{old})} - \mathbf{\Phi}^{\mathrm{T}}\mathbf{t}\right\} \\ &= (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t} \end{aligned} \tag{4.95}$$

우리가 알던 Least square방식의 값과 같다. 이는 error function이 quadratic꼴이고, 이 때 *Newton-Raphson* update 방식에선 one step만에 $\mathbf{w}$의 MLE를 찾을 수 있기 때문이다.

Direct Approach

- Logistic Regression

이제 Logistic Regression에서 $\mathbf{w}$의 MLE를 IRLS방식으로 구해보자.
여기서 다음의 계산을 위해 하고 넘어가야할 것들이 있다. 바로 Notation을 정리하는 것이다. $p(C_k|\phi)$가 위에서 배운 Logistic Sigmoid함수를 따른다고 생각하자. 그러면

- data가 $\phi_n, t_n(0or1)$으로 구성될 때
- $y_n = y(\phi_n) = p(C_1|\phi_n) = \sigma(\mathbf{w}^T\phi)$, $p(C_2|\phi_n) = 1 - p(C_1|\phi_n)$이고
- log likelihood는 $p(t|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}(1-y_n)^{1-t_n}$ 이라고 정의할 수 있다.

또한 logistic sigmoid함수 $\sigma(a)$의 미분값은 다음과 같다.

$$\frac{d\sigma}{da} = \sigma(1-\sigma).$$

Direct Approach

- Logistic Regression

logistic sigmoid의 error function $E(\mathbf{w})$는 다음과 같이 negative log likelihood로 나타내며 이를 cross entropy fucntion이라고 부른다.

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

이것의 gradient와 Hessian을 구하면

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\phi_n = \mathbf{\Phi}^{\mathrm{T}}(\mathbf{y} - \mathbf{t}) \tag{4.96}$$

$$\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1 - y_n)\phi_n\phi_n^{\mathrm{T}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi} \tag{4.97}$$

여기서 $\mathcal{R}$은 $\mathcal{R}_{nn} = y_n(1 - y_n)$인 $\mathcal{N}$ X $\mathcal{N}$ matrix이다.

Direct Approach

- Logistic Regression

위에서 구한 gradient와 Hessian을 *Newton-Raphson* update에 넣어보자.

$$\begin{aligned}\mathbf{w}^{(\text{new})} &= \mathbf{w}^{(\text{old})} - (\mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}(\mathbf{y}-\mathbf{t}) \\ &= (\mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi})^{-1}\left\{\mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{\Phi}^{\mathrm{T}}(\mathbf{y}-\mathbf{t})\right\} \\ &= (\mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{z} \qquad (4.99)\end{aligned}$$

where $\mathbf{z}$ is an N-dimensional vector with elements

$$\mathbf{z} = \mathbf{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{R}^{-1}(\mathbf{y}-\mathbf{t}). \qquad (4.100)$$

이는 위에서 구한 linear regression에서의 값과 비슷해보인다. 하지만 이 식은 R의 원소인 $y_n(1-y_n)$ 이 $\mathbf{w}$를 포함하고 있기 때문에($y_n = \sigma(\mathbf{w}^T\phi)$) one step으로 구하는 것은 불가능하고, 여러 번 step을 밟는 computational한 방법을 써야한다. 이 때, $\mathbf{R}$은 revised weighing matrix라고 불리며 한 스텝마다 곱해진다. 이것 때문에 우리는 이러한 알고리즘을 Iterative Reweighted Least Square(IRLS)로 부르는 것이다.

Direct Approach

- Logistic Regression

다른 방식으로 생각해보자. t_n은 0 또는 1의 값을 가지고, 1의 값을 가질 확률을 y_n이라고 할 수 있다. 따라서 binary distribution이라고 생각하면

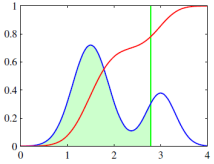
$$\mathbb{E}[t] = \sigma(\mathbf{x}) = y \tag{4.101}$$
$$\mathrm{var}[t] = \mathbb{E}[t^2] - \mathbb{E}[t]^2 = \sigma(\mathbf{x}) - \sigma(\mathbf{x})^2 = y(1-y) \tag{4.102}$$

이다. 따라서 $\mathbf{R}$은 t의 variance matrix라고 볼 수 있다.
$cf)$ 선형 근사를 통해 a_n 구해보기(물론 컴퓨터가 발달한 지금은 쓸 필요가 없다)

Direct Approach

- Probit Regression

지금까지 우리는 logistic sigmoid라는 아름다운 함수로 표현되는 $p(C_k|\phi)$를 사용해왔다. 하지만 세상은 항상 아름답지만은 않다. 다른 모양의 $p(C_k|\phi)$를 써야할 수도 있지 않을까? 이에 대한 해답 중 하나가 바로 CDF를 사용하는 Probit Regression이다.



여기서 빨간 선은 Mixed Gaussian $p()$의 cdf인 $f(a = \mathbf{w}^T\phi)$ 이다. 이것은 최대가 1이고 최소가 0인 sigmoid와 비슷한 모양이다. 따라서 이걸 logistic sigmoid 대신 쓰는 것이 바로 Probit regression이다.
하지만 이것은 단점이 있다. logistic은 x값이 $exp(-x)$에 들어가는데, 여기선 gaussian이라 $exp(-x^2)$에 들어가게 된다. 따라서 outlier에 더 민감하게 반응한다