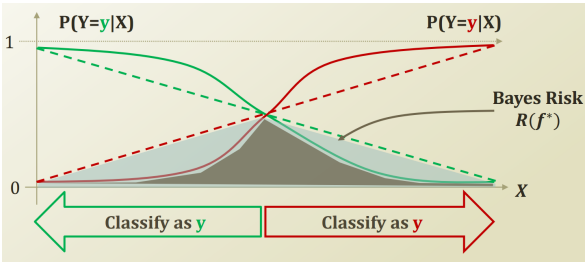
💜 CH4: Logistic Regression

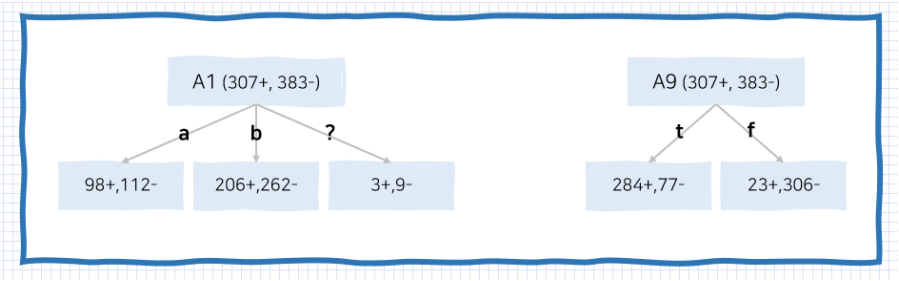
4.1 Logistic Regression

나이브 베이즈 분류기에서는 최적의 분류기란 어떤 것인지에 대해 아래의 그림을 통해 알아보았다. 아래는 두 분류기의 확률 밀도 함수를 점선과 실선으로 각각 나타낸 것이다.

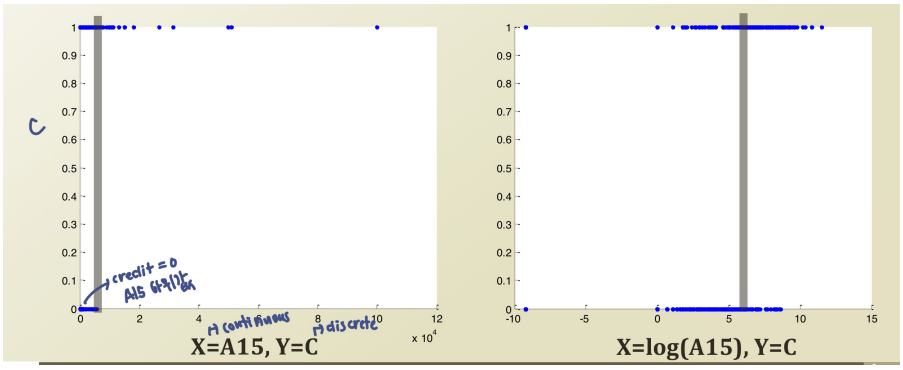


위 그림에서 베이즈 위험도를 통해 점선보다 실선 분류기가 더 좋은 성능을 보인다고 판단했다. 실선 분류기가 점선보다 베이즈 위험도가 낮은 이유는 결정 경계(Decision Boundary)에서 확률이 더 급격하게 변했기 때문이다. 즉, 결정 경계에서 급하게 변할수록 더 좋은 분류기가 된다고 할 수 있다.

로지스틱 회귀를 설명하기 위해 사용할 데이터셋은 CH2에 Decision Tree를 설명할 때 사용한 데이터셋으로 어떤 변수로 분류를 하느냐에 따라 성능이 달라졌다.



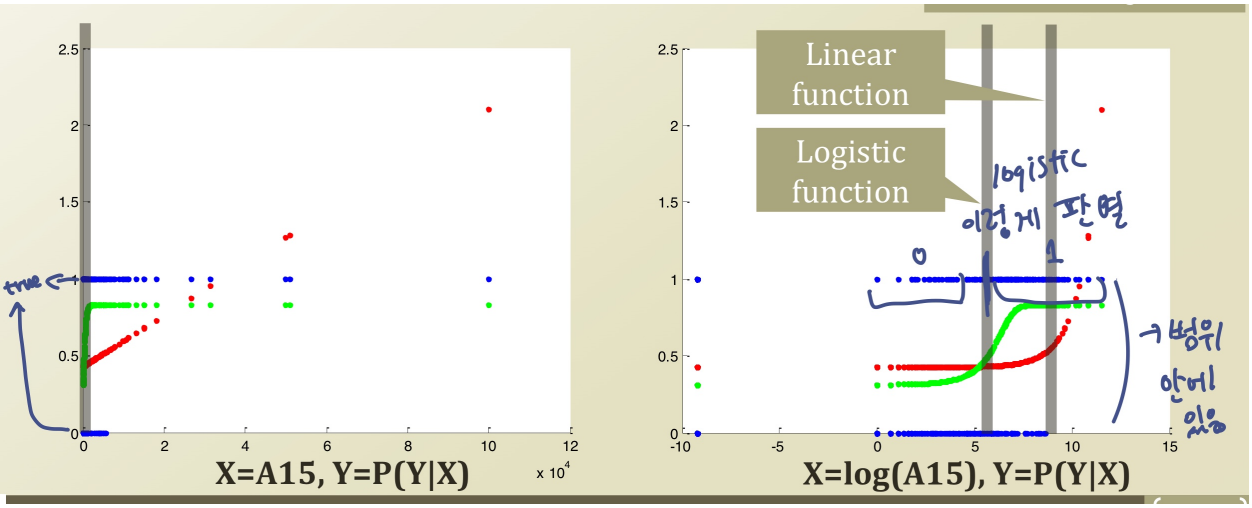
A15변수는 연속형 변수인데, 이 변수에 속한 데이터가 C 클래스로 분류될 확률은 $P(Y = C|X = A15)$이며, 이를 그래프로 그리면 다음과 같다.



4.1.1 Linear Function vs Non-linear Function

위의 그림에서 데이터가 fitting된 것을 보면, 데이터가 1 또는 0 주위에 몰려있는 것을 확인할 수 있다. 이를 만약 선형 회귀 직선으로 fitting하게 된다면, 이는 회귀 가정을 만족하지 않으며, 데이터의 범위를 넘어서기 때문에 예측력이 떨어진다.

이 부분에서 다루게 될 로지스틱 회귀는 결정경계에서 확률이 급격하게 변하는 분류기다. 로지스틱 회귀에서 결정 경계에서 확률이 급격하게 변하는 시그모이드 함수를 사용한다. 로지스틱 회귀를 사용하여 fitting하면 데이터셋 범위 사이에 fitting한것을 확인할 수 있다. 이를 그림으로 보면 다음과 같다.



4.1.2 Sigmoid Function

시그모이드 함수(Sigmoid function)란, 실수 정의역 범위에서 최댓값이 1 이하이고, 최솟값이 0 이상인 미분가능한 단조증가함수이다. 시그모이드 함수는 특정한 하나의 함수가 아니라 위의 조건을 만족하는 함수의 집합으로, $\tanh x, \arctan x, 1/(1+e^{-x})$등 다양한 형태의 종류가 있다. 그 중, 3번째에 있는 함수를 로지스틱 함수(logistic function)이라고 한다. 로지스틱 함수를 사용한 회귀이기 때문에 로지스틱 회귀라는 이름이 붙은 것이다.

시그모이드 함수를 정리하자면

- Bounded: 범위가 -1~1로 제한
- Differentiable: 미분 가능
- Real Function
- Defined for all real inputs
- With positive derivate

로지스틱 함수를 사용하는 이유는 미분이 쉽기 때문이다. 일반적으로 분류기를 최적화하기 위해서는 특정 확률이 최대 또는 최소가 되는 점은 찾는데, 이러한 점을 찾는데 있어, 미분은 필수다. 로지스틱 함수는 이 미분 계산을 쉽게 만들어주기 때문에, 데이터셋이 커지더라도 수행 시간을 적절한 선에서 유지할 수 있다는 장점이 있다.

4.1.2 Logistic Function

로지스틱 함수의 수식은 다음과 같다.

$$y = \frac{1}{1+e^{-x}}$$

오즈(Odds)는 확률의 또 다른 표현법 중 하나로, 어떤 사건의 일어날 확률을 x라고 했을 때 오즈는 다음과 같이 나타낼 수 있다.

$$Odds = \frac{x}{1-x}$$

오즈의 단점은 x가 커지는 경우에 대해서는 값이 양의 무한대로 증가하지만, x가 작아지는 경우에 대해서는 0이하로 작아지지 않는다. 즉, 오즈의 범위는 0~∞이다 예시를 들어보면, $x = 0.5$일 때를 기준으로 $x = 0.99$로 매우 클 때와 $x = 0.01$로 매우 작을 때 오즈가 어떻게 변하는지 확인해보자.

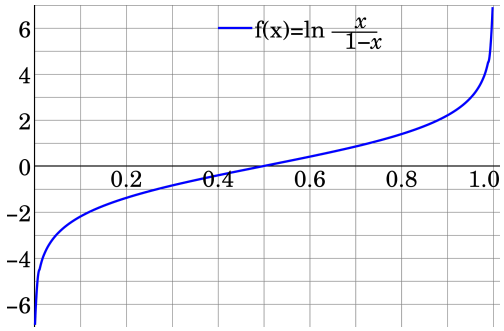
$$Odds(x = 0.5) = \frac{0.5}{1-0.5} = 1$$

$$Odds(x = 0.99) = \frac{0.99}{1-0.99} = 99$$

$$Odds(x = 0.01) = \frac{0.01}{1-0.01} = \frac{1}{99}$$

$x = 0.5$일 때, 즉 어떤 사건이 일어날 때와 일어나지 않을 때의 확률이 동일할 때의 확률은 1이다. 이를 기준으로 $x > 0.5$인 방향으로 변하면 오즈가 급격하게 커지지만, $x < 0.5$인 방향으로 변할 때는 오즈가 완만하게 줄어드는 것을 볼 수 있다. 둘의 증가폭을 맞춰 주기 위해 고안된 방법이 로그 오즈, 즉, 로짓(Logit)이다. 로짓함수의 수식과 그래프는 다음과 같다.

$$logit = log(\frac{x}{1-x})$$



이제 위에서 사용했던 예시를 다시 사용해서 로짓을 나타내보자.

$$logit(x=0.5)=\log(\frac{0.5}{1-0.5})=\log(1)=0$$

$$logit(x=0.99)=\log(\frac{0.99}{1-0.99})=\log(99)=4.595$$

$$logit(x=0.01)=\log(\frac{0.01}{1-0.01})=\log(\frac{1}{99})=-4.595$$

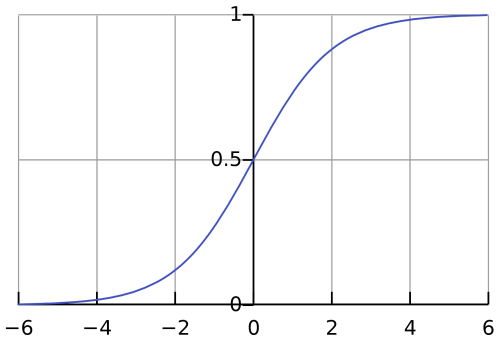
오즈와 달리, 이제 $x=0.5$인 경우, 값이 0이 된다. 그리고 x가 0.5보다 크던 작던 로짓 값이 동일한 폭으로 증갑하는 것을 볼 수 있다. 즉, 이제 범위는 $-\infty \sim 0, 0 \sim \infty$이다. 그렇기 때문에 실제로는 오즈보다는 로짓을 더욱 많이 사용하며, 로지스틱 함수를 유도하는데도 로짓을 사용한다. 로짓을 x로 놓고 역함수를 취하면 다음과 같다진다.

$$f(x)=\log(\frac{x}{1-x}) \longrightarrow^{inverse} \quad \log(\frac{e^x}{1+e^x})$$

도출된 역함수의 식에서 분모를 분자로 나눠주면 처음 보앗던 로지스틱 함수와 유사한 형태나 나타난다.

$$\text{Logsitic Function}: \log(\frac{1}{1+e^{-x}})$$

이렇게 도출된 로지스틱 함수는 특성값이 x인 인스턴스가 특정 클래스에 속할 확률을 구하는 데 사용되며, 로지스틱 함수는 로짓 함수의 역함수이므로 다음과 같은 그래프를 가진다.



4.1.3 Logistic Regression

로지스틱 함수가 유도되는 과정을 우도(likelihood)를 사용하여 다음과 같이 쓸 수 있다.

선형회귀에서 $\hat{f}=X\theta$로 나타냈던 것을 이용하여 위의 식을 다음과 같이 변형할 수 있다.

$$X\theta=log(\frac{p}{1-p}) \quad where\ \hat{f}=X\theta=ax+b$$

$$\therefore p=\log(\frac{1}{1+e^{-X\theta}})$$

다음으로 베르누이 시행의 식을 살펴보다. 베르누이 시행에서 $y=1$일 때의 확률은 $\mu(x)$라고 하면, $P(y|x)$의 식은 다음과 같다.

$$P(y|x)=\mu(x)^y(1-\mu(X))^{1-y}$$

이를 로지스틱 함수를 사용하여 $\mu(x)$에 대한 식을 바꾸면, 다음과 같다. 이때, 하나의 인스턴스에 대해서만 이야기를 하고 있음으로 벡터의 집합 $X\theta$대신, θ^Tx를 사용한 것이다.

$$\mu(x)=\frac{1}{1+e^{-\theta^Tx}}=P(y=1|x)$$

이를 전체 인스턴스에 관한 식으로 바꾸면 다음과 같이 쓸 수 있다.

$$X\theta = log(\frac{P(Y|X)}{1-P(Y|X)}) \longrightarrow P(Y|X) = \frac{e^X\theta}{1+e^X\theta} = \frac{1}{1+e^{-X\theta}}$$

위의 식을 부면, θ를 제외한 파라미터들은 데이터를 통해 알 수 있으니 지금부터 θ를 optimize하는 방법을 알아보자.

4.1.4 Parameter Estimation

로지스틱 회귀가 파라미터를 추정하는 방식은 $P(Y|X)$를 추정하는 최대 조건부 우도 추정(Maximum Conditional Likelihood Estimation, MCLE)에 기반하고 있다. 먼저 MLE부터 다시 살펴보자.

$$\hat{\theta} = \arg\max_{\theta} P(D|\theta)$$

최대 조건부 우도 추정식(MCLE)은 최대 우도 추정식을 변형하여 다음과 같이 쓸 수 있다.

$$\hat{\theta} = \arg\max_{\theta} P(D|\theta) = \arg\max_{\theta} \prod_{1\leq i\leq N} P(Y_i|X_i;\theta) = \arg\max_{\theta} \log(\prod_{1\leq i\leq N} P(Y_i|X_i;\theta)) = \arg\max_{\theta} \sum_{1\leq i\leq N} \log(P(Y_i|X_i;\theta))$$

위의 식에서 로그를 사용할 수 있는 이유는 로그 함수는 단조 증가 함수이기 때문에, 식을 계산하는 것에 대하여 큰 영향을 주지 않기 때문이다. 이제 베르누이 시행,$P(Y_i|X_i;\theta) = \mu(X_i)^{Y_i}(1-\mu(X_i))^{1-Y_i}$,에 있는 θ를 optimize 해보자.

$$\log(P(Y_i|X_i;\theta)) = Y_i\log(\mu(X_i)) + (1-Y_i)log(1-\mu(X_i)) = Y_i\{\log(\mu(X_i)) - log(1-\mu(X_i))\} + log(1-\mu(X_i)) = Y_i\log(\frac{\mu(X_i)}{1-\mu(X_i)}) + log(1-\mu(X_i)) = Y_iX_i\theta + log$$

이제 이 식을 다음식에 대입한다.

$$\hat{\theta} = \arg\max_{\theta} \sum_{1\leq i\leq N} \log P(Y_i|X_i;\theta) = \arg\max_{\theta} \sum_{1\leq i\leq N} \{Y_iX_i\theta - \log(1+e^{X_i\theta})\}$$

최대 우도 추정에서 했던 것과 같이 $\arg\max$ 이하의 식을 θ로 미분하여 0이 되는 θ를 찾아보자.

$$\frac{\partial}{\partial\theta_j}\{\sum_{1\leq i\leq N} Y_iX_i\theta - \log(1+e^{X_i\theta})\} = \{\sum_{1\leq i\leq N} Y_iX_{i,j}\} + \{\sum_{1\leq i\leq N} -\frac{1}{1+e^{X_i\theta}} \times e^{X_i\theta} \times X_{i,j}\} = \sum_{1\leq i\leq N} X_{i,j}(Y_i - \frac{e^{X_i\theta}}{1+e^{X_i\theta}}) = \sum_{1\leq i\leq N} X_{i,j}(Y_i - P(Y_i=1|X_i;\theta)) = 0$$

이 식은 보다싶이, 선형회귀의 최소제곱법처럼 단 하나의 닫힌 해(closed form)이 나오지 않는다. 그러므로 우리는 다른 방법을 사용하여 해를 추론해야 한다. 그 방법 중 하나는 경사하강법이다. 여기서는 최대 우도를 구하는 즉, $\arg\max$인 지점을 구하는 것이므로 경사 상승법을 사용해야 한다.

4.2 Gradient Method

4.2.1 Taylor Expansion

테일러 급수 또는 테일러 전개는 어떤 미의 함수$f(x)$를 아래 식과 같이 근사 다항함수로 표현하는 것을 말한다.

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + ... = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

테일러 급수가 필요한 이유는 쉽게 말하면 우리가 잘 모르거나 복잡한 함수를 다루기 쉽고 이해하기 쉬운 다항함수로 대체시키기 위함입니다. 또한 어떤 함수를 테일러 급수로 표현하면 그 함수의 특성을 분석하기가 좀더 용이해지기 때문이다.

4.22 Gradient Descent/ Ascent

Gradient Descent란 그라디언트를 이요해 최소값을 찾을 때의 방법으로 딥러닝에서도 사용된다. 경사 하강법의 알고리즘을 다음과 같다. 그라디언트가 가리키는 방향은 등고선에서의 수직 방향을 가리키는데, 이는 등고선에서 가장 빠르게 증가 감소하는 방향을 가리킨다. 즉, 등고선에서 값이 변할 때마다 그 값에서 그라디언트가 가르키는 방향으로 움직이면 최대값이 되고, 이 과정을 Gradient Acsent라고 한다. 반대의 경우인 Gradient Descent는 loss 값을 최소화할 때 사용하며, 각 등고선 값에서 그라디언트에 -1을 곱하여 정확히 반대 방향으로 바꾸는 것이다.

정리하자면, 경사 하강법 /상승법은 미분 가능한 함수와 초기값이 주어졌을 때, 함수에 대해 더 높거나 낮은 값이 반복적으로 이동시키는 방법이다. 이동하기 위해서는 방향, 속력 2개를 알아야 한다. 속력이 느리더라고 방향이 맞으면 올바른 값으로 수렴할 수 있지만, 방향이 틀리면 올바르게 수렴하기 어렵기 때문에 방향이 중요하다. ~~(방향과 속력은 어떻게 구하나..? 바로 1차 미분과 2차 미분이다...!)~~

속력 h는 방향을 가지는 단위 벡터로 현재 위치에서 속력 h로 방향 u로 이동한다. 즉, x_1이라는 위치에서 속력 h로 u방향으로 가겠다는 것이다. Taylor Expansion을 적용할 때, 방향 u를 잘 정해야 하며, h를 최적화한 후 다음 차례의 x값을 구할 수 있다. 앞에서 말했듯이 값을 줄여나가면, Gradient Descent, 값을 늘리면 Gradient Ascent가 된다. 이를 식으로 표현하면 다음과 같다.

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + O(||x-a||^2) \quad \text{Assume } a = x_1 \text{ and } x = x_1 + hu,\ u \text{ is the unit direction vector for the partial derivative}$$

$$f(x_1+hu) = f(x_1) + hf'(x_1)u + h^2O(1)$$

$$f(x_1+hu) - f(x_1) \approx hf'(x_1)u$$

이때, 사용되는 h는 작은 값이어야 한다. 예를 들어 h가 0.1인 경우, 미분 사추가 많아질수록 $h^{(n)}O(1)$값이 0으로 수렴하기 때문에 뒤에 에러 부분이 무시 가능하다. 우리의 목적은 negative한 방향으로 갱신하는 것이기 때문에, 최소화하는 문제이기 때문에 다음과 같이 쓸 수 있다.

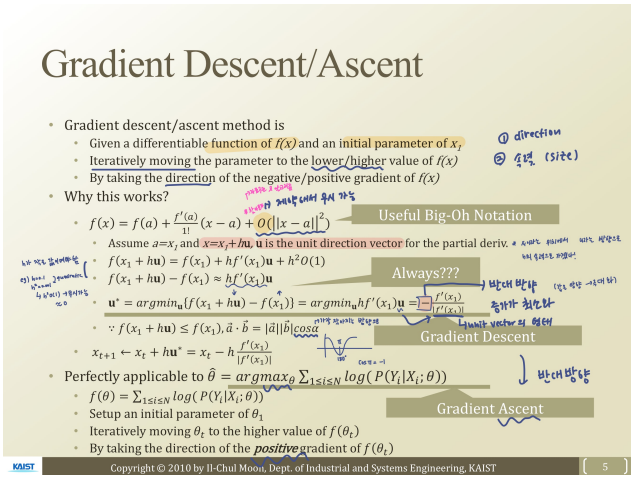
$$u^* = \arg\min_u \{f(x_1+hu) - f(x_1)\} = \arg\min_u hf'(x_1)u = -\frac{f'(x_1)}{|f'(x_1)|} \qquad where\ f(x_1+hu) \leq f(x_1), \vec{a}\cdot\vec{b} = |\vec{a}||\vec{b}|\cos\alpha$$

이제 준비가 끝났으니 경사하강법의 식을 쓸 수 있다. 경사 하강법에서는 x_t가 x_{t+1}으로 negative한 방향으로 갱신 되기 때문에 식은 다음과 같다.

$$x_{t+1} \leftarrow x_t + hu^* = x_t - h\frac{f'(x_1)}{|f'(x_1)|}$$

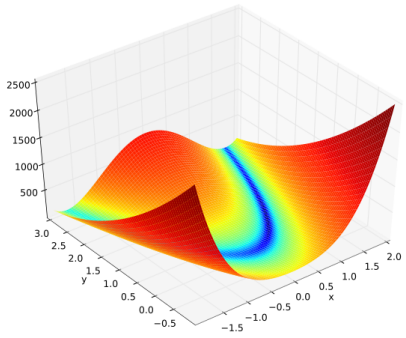
경사 상승법은 반대로 기울기가 상승하는 방향으로 갱신하기 때문에 부호만 반대로 바꿔주면 된다.

$$x_{t+1} \leftarrow x_t + hu^* = x_t + h\frac{f'(x_1)}{|f'(x_1)|}$$



갑분 예시 (넘어가도 됨)

θ에 대해서 어떻게 갱신되는지 알아보기 전에, 먼저 예시를 한 번 살펴보자. 아래의 예시는 로젠브록 함수(Rosenbrock function)로 수학적 최적화에서 최적화 알고리즘을 시험해볼 용도로 사용하는 비블록 함수이다.

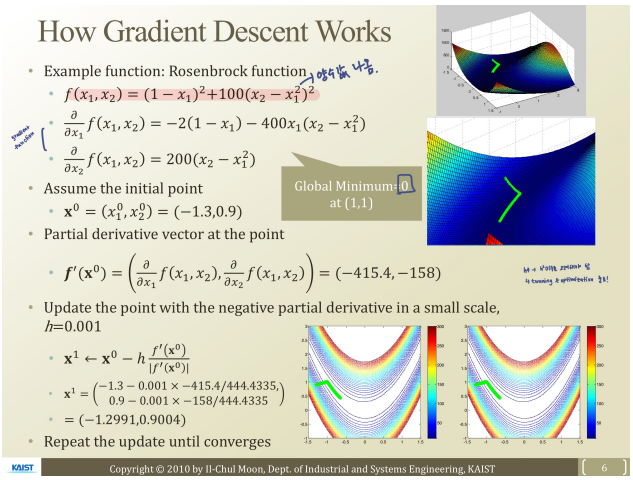


$$f(x,y) = (a-x)^2 + b(y-x^2)^2$$

그래프를 보면, 길고 좁은 포물선 모양의 골짜기를 들러내며, 보이다시피 global minima로 수렴하는 것이 힘들어보인다. Global minima는 $f(x=a, y-a^2)=0$이며, 일반적으로$a=1, b=100$을 대입해 사용한다.

$$f(x_1,x_2) = (1-x_1)^2 + 100(x_2-x_1^2)^2$$

구하는 방법은 다음과 같다.



이제 다시 본론으로 돌아와 θ에 대해서 어떻게 갱신되는지 알아보자. 위에서 미분산 식을 가져와서 다음과 같이 쓸 수 있다. 이때, C는 유닛 벡터에 대한 normalizing constant이다.

$$\theta_j^{t+1} \leftarrow \theta_j^t + h\frac{\partial f(\theta^t)}{\partial \theta_j^t} = \theta_j^t + h\{\sum_{1\le i\le N} X_{i,j}(Y_i - P(Y=1|X_i;\theta^t))\} = \theta_j^t + \frac{h}{C}\{\sum_{1\le i\le N} X_{i,j}(Y_i - \frac{e^{X_i\theta^t}}{1+e^{X_i\theta^t}})\}$$

앞서 선형 회귀에서 θ를 최소화 하는 과정은 다음과 같았다.

$$\hat{\theta} = \arg\min_\theta (f-\hat{f})^2 = \arg\min_\theta (Y-X\theta)^2 = \arg\min_\theta (Y-X\theta)^T(Y-X\theta) = \arg\min_\theta (\theta^T X^T X\theta - 2\theta^T X^T Y + Y^T Y) = \arg\min_\theta (\theta^T X^T X\theta - 2\theta^T X^T Y)$$

$$\nabla_\theta(\theta^T X^T X\theta - 2\theta^T X^T Y) = 0$$
$$2X^T X\theta - 2X^T Y = 0$$
$$\theta = (X^T X)^{-1} X^T Y$$

이때, θ는 closed form으로 결과를 도출해낼 수 있었다. 하지만, 만약 데이터/feature의 크기가 커질수록 matrix를 inverse하기가 쉽지 않기 때문에, 여기서도 경사 하강법/상승법을 사용하여 θ값을 추론해 낼 수 있다.

$$\hat{\theta} = \arg\min_\theta (f-\hat{f})^2 = \arg\min_\theta (Y-X\theta)^2 = \arg\min \sum_{1\le i\le N} (Y^i - \sum_{1\le j\le d} X_j^i\theta_j)^2$$

$$\frac{\partial}{\partial\theta_k}\sum_{1\le i\le N}(Y^i - \sum_{1\le j\le d} X_j^i\theta_j)^2 = -\sum_{1\le i\le N} 2(Y^i - \sum_{1\le j\le d} X_j^i\theta_j)X_k^i$$
$$\theta_k^{t+1} \leftarrow \theta_k^t - h\frac{\partial f(\theta^t)}{\partial\theta_k^t} = \theta_k^t + h\sum_{1\le i\le N} 2(Y^i - \sum_{1\le j\le d} X_j^i\theta_j)X_k^i$$

4.3 Naive Bayes vs Logistic Regression

4.3.1 Gaussian Naive Bayes

앞서 배운 나이브 베이즈 분류기와 로지스틱 회귀의 성능을 비교하고자 한다. 우리가 3장에서 정의한 나이브 베이즈 분류기는 discrete한 데이터에 대한 분류기로, 식은 다음과 같았다.

$$f_{NB}(x) = \arg\max_{Y=y} \prod_{1\le i\le d} P(X_i = x_i|Y=y)$$

만약, 데이터가 continuous한 경우, 위의 식에 가우시안 분포를 적용하여, 가우시안 나이브 베이즈 분류기를 만들 수 있다. 가우시안 분포와 가우시안 나이브 베이즈 분류기의 식은 다음과 같다.

$$P(X_i|Y,\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(X_i-\mu)^2}{2\sigma^2}}$$

$$P(Y=y) = \pi_1$$

$$f_{GNB}(x) = P(Y)\prod_{1\le i\le d} P(X_i|Y) = \pi_k \prod_{1\le i\le d} \frac{1}{\sigma_k^i C} e^{-\frac{1}{2}(\frac{X_i-\mu_k^i}{\sigma_k^i})^2}$$

이제 나이브 베이즈 식을 통해, 로지스틱 회귀 식을 도출해보자. 이를 도출하기 위해, 나이브 베이즈 가정이 필요하다. 이 과정을 식을 통해 알아보자.

$$P(Y=y|X) = \frac{P(X|Y=y)P(Y=y)}{P(X)} = \frac{P(X|Y=y)P(Y=y)}{P(X|Y=y)P(Y=y)+P(X|Y=n)P(Y=n)} = \frac{P(Y=y)\prod_{1\le i\le d}P(X_i|Y=y)}{P(Y=y)\prod_{1\le i\le d}P(X_i|Y=y)+P(Y=n)\prod_{1\le i\le d}P(X_i|Y=n)}$$

마지막 식의 분자 부분의 $P(X_i|Y=y)$는 위에 정의 했듯이 가우시안 분포를 뜻하며, 이를 다시 쓰면, 다음과 같이 식을 쓸 수 있다.

$$P(Y=y|X)=\frac{\pi_1\prod_{1\le i\le d}\frac{1}{\sigma_1^i C}\exp(-\frac{1}{2}(\frac{X_i-\mu_1^i}{\sigma_1^i})^2)}{\pi_1\prod_{1\le i\le d}\frac{1}{\sigma_1^i C}\exp(-\frac{1}{2}(\frac{X_i-\mu_1^i}{\sigma_1^i})^2)+\pi_2\prod_{1\le i\le d}\frac{1}{\sigma_2^i C}\exp(-\frac{1}{2}(\frac{X_i-\mu_2^i}{\sigma_2^i})^2)}=\frac{1}{1+\frac{\pi_2\prod_{1\le i\le d}\frac{1}{\sigma_2^i C}\exp(-\frac{1}{2}(\frac{X_i-\mu_2^i}{\sigma_2^i})^2)}{\pi_1\prod_{1\le i\le d}\frac{1}{\sigma_1^i C}\exp(-\frac{1}{2}(\frac{X_i-\mu_1^i}{\sigma_1^i})^2)}}$$

식을 보면, 미지수가 너무 많아 더 이상 나아갈 수가 없다. 따라서, 두 변수가 같은 분산을 가지고 있다고 가정을 해보자, $\sigma_2^i=\sigma_1^i$.

$$P(Y=y|X)=\frac{1}{1+\frac{\pi_2\prod_{1\le i\le d}\frac{1}{\sigma_2^i C}\exp(-\frac{1}{2}(\frac{X_i-\mu_2^i}{\sigma_2^i})^2)}{\pi_1\prod_{1\le i\le d}\frac{1}{\sigma_1^i C}\exp(-\frac{1}{2}(\frac{X_i-\mu_1^i}{\sigma_1^i})^2)}}=\frac{1}{1+\frac{\pi_2\prod_{1\le i\le d}\exp(-\frac{1}{2}(\frac{X_i-\mu_2^i}{\sigma_2^i})^2)}{\pi_1\prod_{1\le i\le d}\exp(-\frac{1}{2}(\frac{X_i-\mu_1^i}{\sigma_1^i})^2)}}=\frac{1}{1+\frac{\pi_2\exp(-\sum_{1\le i\le d}\{-\frac{1}{2}(\frac{X_i-\mu_2^i}{\sigma_2^i})^2\})}{\pi_1\exp(-\sum_{1\le i\le d}\{-\frac{1}{2}(\frac{X_i-\mu_1^i}{\sigma_1^i})^2\})}}=\frac{1}{1+\frac{\exp(-\sum_{1\le i\le d}\{-\frac{1}{2}(\frac{X_i-\mu_2^i}{\sigma_2^i})^2\})+\log\pi_2)}{\exp(-\sum_{1\le i\le d}\{-\frac{1}{2}(\frac{X_i-\mu_1^i}{\sigma_1^i})^2\}+\log\pi_1)}}$$

등분산 가정을 유지한체 계속 식을 정리해보면 다음과 같이 로지스틱 회귀의 모양과 동일한 식을 도출할 수 있다.

$$P(Y=y|X)=\frac{1}{1+\exp(-\sum_{1\le i\le d}\{\frac{1}{2}(\frac{X_i-\mu_2^i}{\sigma_2^i})^2\}+\log\pi_2-\sum_{1\le i\le d}\{\frac{1}{2}(\frac{X_i-\mu_1^i}{\sigma_1^i})^2\})-\log\pi_1)}=\frac{1}{1+\exp(-\frac{1}{2(\sigma_1^i)^2}\sum_{1\le i\le d}\{(X_i-\mu_1^i)^2-(X_i-\mu_2^i)^2\}+\log\pi_2-\log\pi_1)}=$$

Naive Bayes classifier 와 Logistic Regression의 차이는 무엇이까? 각각의 공식을 얻기위한 Assumption 이 다르고, Estimation해야 할 Parameter의 수가 다르다.

나이브 베이즈 분류기 같은 경우 다음과 같은 가정을 해야하며, 추정해야하는 파라미터는 총 $2\times 2\times d+1=4d+1$이다.

- Naive Bayes Assumption
- Same variance assumption between classes
- Gaussian distribution for $P(X|Y)$
- Bernoulli distribution for $P(Y)$

로지스틱 회귀 같은 경우는 다음과 같고, 추정해야하는 파라미터는 총 $d+1$개이다.

- fitting to the logistic function

그러면 어느쪽이 더 나은 모델이라고 할수 있을까? 가정이나, 추정해야하는 파라미터의 개수를 봐서는 로지스틱 회귀가 간단해서 좋아보인다. 정확도 수준에서는 보통 Logistic Regression이 좀더 좋은 성능을 보이지만, 나이브 베이즈 분류기는 prior 정보를 추가할 수 있다는 장점이 있다. 즉, 주어진 정보와 상황에 따라 모델을 적절히 사용해야 한다.

4.3.2 Generative-Discriminative Pair

Generative Model: $P(Y|X)=P(X,Y)/P(X)=P(X|Y)P(Y)/P(X)$

생성모델은 주어진 학습 데이터를 학습하여 학습 데이터의 분포를 따르는 유사한 데이터를 생성하는 모델이다. 이와 같이 학습 데이터와 유사한 샘플을 뽑아야 하기 때문에 생성 모델에는 락습 데이터의 분포를 어느 정도 안 상태에서 생성하거나 잘 모르지만 그럼에도 생성하는 다양한 모델들이 존재한다. 결국 생성모델에서 가장 중요한 것은 학습 데이터의 분포를 학습하는 것이 제일 중요하다.

Discriminative Model:

판별모델은 생성모델과 반대로 입력 받은 데이터를 어떤 기준으로 판별하는 것이 목표인 모델링으로 예시로는 로지스틱 회귀가 있다.