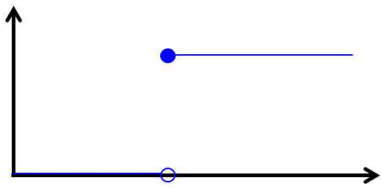
Machine Learning Lecture Note

This lecture note is based on the CS229 lecture materials from Standford University.

Lecture 4. Perceptron, Generative learning algorithms (GDA, Naive Bayes)

Perceptron learning 알고리즘

로지스틱 회귀 알고리즘을 변형하여 다음과 같은 함수를 고려하면,



$$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

$h_\theta(x) = g(\theta^T x)$ 라고 하면, θ 업데이트는 $\theta_j := \theta_j + \alpha\left(y^i - h_\theta(x^i)\right)x_j^i$ 이다.

퍼셉트론 알고리즘은 이미 소개된 다른 알고리즘과 외형적으로 유사하지만, 로지스틱 회귀 및 선형 회귀 알고리즘과는 다르다. 퍼셉트론 알고리즘의 예측에 의미 있는 확률적 해석을 부여하거나 최대 우도 추정 알고리즘으로 도출하는 것은 어렵다.

Generative discriminant analysis

Discriminates (분류) : Learn $p(y|x)$ or learn $h_\theta(x) = \begin{cases} 0 \\ 1 \end{cases}$

Generative learning algorithm 프레임워크:

Learn $p(x|y)$ (x: feature, y: class), $p(y)$ (class prior)

Bayes rule: $p(y=1|x) = \dfrac{p(x|y=1)p(y=1)}{p(x)}$, $p(x) = p(x|y=1)p(y=1) + p(x|y=0)p(y=0)$

Gaussian Discriminant Analysis (GDA)

입력 데이터 x가 연속적인 랜덤변수로 표현된다면 GDA를 상용할 수 있다. GDA에서는 $p(x|y)$은 Gaussian 분포로 가정한다.

$z \sim N(\mu, \Sigma)$, $z \in R^n, \mu \in R^n, \Sigma \in R^{n \times n}$

$E[x] = \mu$

$Cov(z) = E[(z-\mu)(z-\mu)^T] = E[zz^T] - E[z]E[z]^T$

$$p(z) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\right)$$

$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\right)$$

$p(y) = \phi^y(1-\phi)^{1-y}$, $(p(y=1) = \phi)$

훈련 데이터가 다음과 같이 주어지면, $\{x^i, y^i\}_{i=1}^m$

파라미터들$(\phi, \mu_0, \mu_1, \Sigma)$에 대하여, Likelihood 함수로 표현하면 다음과 같다:

$$\begin{aligned} L(\phi, \mu_0, \mu_1, \Sigma) &= \prod_{i=1}^m p(x^i, y^i; \phi, \mu_0, \mu_1, \Sigma) \\ &= \prod_{i=1}^m p(x^i|y^i; \phi, \mu_0, \mu_1, \Sigma) p(y^i; \phi) \end{aligned}$$

Discriminant 학습 알고리즘:

Conditional likelihood $L(\theta) = \prod_{i=1}^{m} p(y^i|x^i;\theta)$

Maximum likelihood estimation:

$$\max_{\phi,\mu_0,\mu_1,\Sigma} \log L(\phi,\mu_0,\mu_1,\Sigma) = l(\phi,\mu_0,\mu_1,\Sigma)$$

$$\phi = \frac{1}{m}\sum_{i=1}^{m} y^i = \frac{1}{m}\sum_{i=1}^{m} 1\{y^i = 1\}$$

$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^i = 0\}x^i}{\sum_{i=1}^{m} 1\{y^i = 0\}}$$

$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^i = 1\}x^i}{\sum_{i=1}^{m} 1\{y^i = 1\}}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m} (x^i - \mu_{y^i})(x^i - \mu_{y^i})^T$$

Prediction:

$$\operatorname{argmax}_y p(y|x) = \operatorname{argmax}_y \frac{p(x|y)p(y)}{p(x)} = \operatorname{argmax}_y p(x|y)p(y)$$

로지스틱 회귀와 비교

GDA는 모델링 가정이 더 강력하며, 모델링의 가정이 정확할수록 데이터 효율성이 더 높다 (즉, 학습을 위해 더 적은 훈련데이터가 요구된다). 로지스틱 회귀의 가정이 더 약하며, 모델링에 대한 가정의 편차에 대해 훨씬 더 강건하다. 따라서 로지스틱 회귀는 거의 항상 GDA보다 더 나은 성능을 보인다. 이러한 이유로 실제로는 로지스틱 회귀가 GDA보다 더 자주 사용된다.

Naive Bayes

Feature vector (특징 벡터) x : 데이터 포인트를 숫자 값의 목록으로 표현하는 방법으로, 각 값은 해당 객체의 구체적인 특성이나 특징을 나타낸다.

예) 사전 데이터

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \text{a} \\ \text{aardvark} \\ \text{aardwolf} \\ \vdots \\ \text{buy} \\ \vdots \\ \text{zygmurgy} \end{matrix}$$

$x \in \{0,1\}^n$, $n = 10{,}000$

$x_i = 1$ {단어가 사전에 존재}

$x_i = 0$ {단어가 사전에 존재하지 않음}

이메일 스팸 필터링 문제

특징 벡터를 사용하여 생성 모델 (generative model) 구축: $p(x|y), p(y)$

$p(x|y)$를 모델링 하기 위해서 다음의 강한 가정을 이용한다: x_i는 주어진 y에 대해서 조건부 독립적 (conditionally independent) 이다. (Naive Bayes 가정)

$$
\begin{aligned}
p(x_1,\ldots,x_{10000}|y) &= p(x_1|y)p(x_2|y,x_1)p(x_3|y,x_1,x_2)\cdots p(x_{10000}|y,x_1,\ldots,x_{9999}) \\
&= p(x_1|y)p(x_2|y)p(x_3|y)\cdots p(x_{10000}|y) \\
&= \prod_{j=1}^{n} p(x_j|y)
\end{aligned}
$$

Naive Bayes 가정을 기반으로 한 데이터 분류 알고리즘을 **Navie Bayes classifier**라고 한다.
Parameters:

$\phi_{j|y=1} = p(x_j = 1|y=1)$ $\quad y=1$ 스팸 이메일
$\phi_{j|y=0} = p(x_j = 1|y=0)$ $\quad y=0$ 스팸이 아닌 이메일
$\phi_y = p(y=1)$

Joint likelihood:

$$L(\phi_y, \phi_{j|y}) = \prod_{i=1}^{m} p\left(x^i, y^i; \phi_y, \phi_{j|y}\right)$$

Maximum likelihood estimator:

$p(y=1) = \phi_y$
$p(x_j = 1|y=0) = \phi_{j|y=0}$
$p(x_j = 1|y=1) = \phi_{j|y=1}$

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_j^i = 1 \wedge y^i = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^i = 1\right\}},\ \phi_{j|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_j^i = 1 \wedge y^i = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^i = 0\right\}},\ \phi_y = \frac{\sum_{i=1}^{m} 1\left\{y^i = 1\right\}}{m}$$

Make prediction:

$$p(y=1|x) = \frac{p(x|y=1)p(y=1)}{p(x)} = \frac{p(x|y=1)p(y=1)}{p(x|y=1)p(y=1) + p(x|y=0)p(y=0)}$$

Laplace smoothing

스팸 필터링을 위한 특징 벡터 (즉, 훈련 데이터 셋)에 나타나지 않은 새로운 단어가 출현한 경우,

$$p(y=1|x) = \frac{p(x|y=1)p(y=1)}{p(x|y=1)p(y=1) + p(x|y=0)p(y=0)} = \frac{0}{0+0}$$

이러한 문제를 해결하기 위해, $\{1,\ldots,k\}$의 값을 갖는 확률변수 y의 평균을 추정하도록 한다.

$$\phi_j = \frac{\sum_{i=1}^{m} 1\left\{y^i = j\right\} + 1}{m + k}$$

따라서, Naive Bayes classifier with Laplace smoothing은 다음과 같다:

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_j^i = 1 \wedge y^i = 1\right\} + 1}{\sum_{i=1}^{m} 1\left\{y^i = 1\right\} + 2},\ \phi_{j|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_j^i = 1 \wedge y^i = 0\right\} + 1}{\sum_{i=1}^{m} 1\left\{y^i = 0\right\} + 2}$$