

CS229 Lecture notes

2021-08-15 | Jihyun Lee

Content

Supervised learning

- Part I . Linear Regression
 1. LMS algorithm
 2. The normal equations
 3. Probabilistic interpretation
 4. Locally weighted linear regression
- Part III . Generalized Linear Models
 1. The exponential family
 2. Constructing GLMs
 - a. Ordinary Least Squares
 - b. Logistic Regression
 - c. Softmax Regression

Supervised learning

Machine Learning

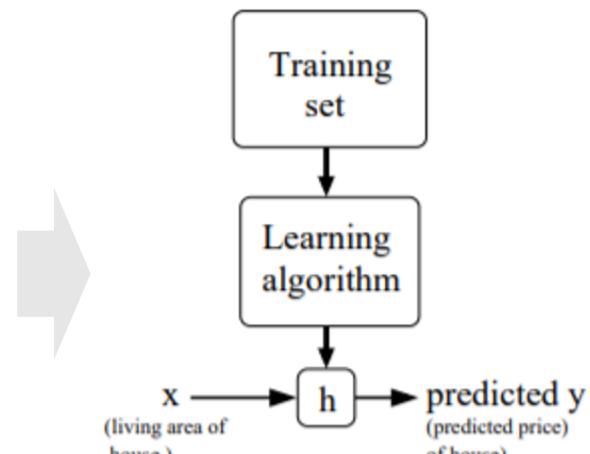
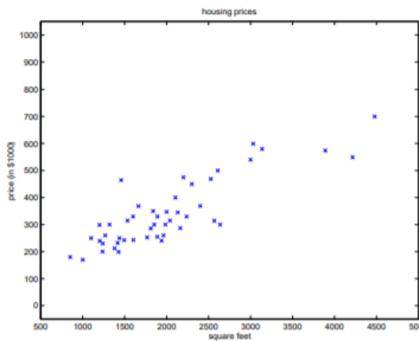
학습	설명	종류	알고리즘
Supervised Learning	<ul style="list-style-type: none">정답 (label) 이 무엇인지 알려주며 모델 학습하는 방법정확한 input과 output 존재	<ul style="list-style-type: none">ClassificationRegression	<ul style="list-style-type: none">Classification : kNN, Support Vector (Part II)Regression : Linear Regression, Locally Weighted Linear (Part I)
Unsupervised Learning	<ul style="list-style-type: none">정답 (label) 없이, 비슷한 데이터를 군집화 하여 예측하는 방법label 이 되어있지 않은 데이터로부터 패턴이나 형태를 찾아야 함		<ul style="list-style-type: none">ClusteringK MeansDensity Estimation
Reinforcement Learning	<ul style="list-style-type: none">분류할 수 있는 데이터가 존재하는 것도 아니고, 데이터가 있다해도 label 이 따로 정해져 있지 않으며, 모델이 한 행동에 대해 보상 (reward) 을 받으며 학습agent, environment, state, action, reward 개념		<ul style="list-style-type: none">DQNA3C

Part I. Linear Regression

Supervised Learning

- notation

Living area (feet ²)	Price (1000\$)
2104	400
1600	330
2400	369
1416	232
3000	540
:	:



- 1) hypothesis $h(x)$: function h 는 새로운 값을 잘 예측 하도록 함.
- 1) $(x^{(i)}, y^{(i)})$: training example / set
(input variables & features, target)

우리가 예측해야 하는 target variable

- continuous : regression problem
- small number of discrete values : classification problem

다음 형태의 data가 주어졌을 때,

우리는 어떻게 다른 집 가격을 '예측 (predict)' 할 수 있을까?

Supervised Learning 을 수행하기 위해서 우리는

- 모델 가설 (represent functions / hypotheses h) 를 세워야 함.
- 우리가 모델을 세운다는 것 => 모델의 파라미터 (θ)를 결정하는 것.

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \quad | \quad h(x) = \sum_{i=0}^n \theta_i x_i = \theta^T x,$$

θ : 파라미터, weights

n : input variables (input features), 몇 개의 feature 를 추출할 것인지. => 이후 단원에서 feature selection 다 름.

우리는 θ 를 어떻게 결정해야 하는가?

- $h(x)$ (예측한 결과) 를 정답 label (y) 과 가깝게 θ 를 조정해야 함!

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2.$$

Objective function (목적 함수) = Cost function (비용 함수)

- 작은 값을 가질 수록 좋음. (비용은 작을수록 좋으니까)
- ordinary least squares (최소 자승법) => LMS algorithm

1. LMS algorithm

- 우리는 $J(\theta)$ 를 최소화하는 θ 를 선택하고 싶음.

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2.$$

- 이를 위해서 'initial guess' θ 를 선택한 후, 반복적으로 $J(\theta)$ 를 작게 업데이트!

$\theta_j := \theta_j - \alpha \cdot \frac{\partial}{\partial \theta_j} J(\theta)$ <p>\rightarrow 각각의 $\theta_0, \theta_1, \theta_2, \dots, \theta_n$ 업데이트.</p> $\begin{cases} \theta_0 := \theta_0 - \alpha \cdot \frac{\partial}{\partial \theta_0} J(\theta) \\ \theta_1 := \theta_1 - \alpha \cdot \frac{\partial}{\partial \theta_1} J(\theta) \\ \vdots \\ \theta_n := \theta_n - \alpha \cdot \frac{\partial}{\partial \theta_n} J(\theta) \end{cases}$ <p>(α: learning rate).</p>	$J(\theta) = \frac{1}{2} (h_\theta(x) - y)^2$ $\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{2} \cdot 2 \cdot (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y)$ $= (h_\theta(x) - y) \cdot \underbrace{\frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^n \theta_i x_i - y \right)}_{\rightarrow \theta_j \text{ 을 제외한 모든 } \theta_i \text{ 는 상수}.}$ $= (h_\theta(x) - y) \cdot \cancel{\underline{x_j}}$ $\frac{\partial}{\partial \theta_j} (h_\theta(x) - y) + \cancel{\underline{(h_\theta(x) - y) \cdot x_0 + (h_\theta(x) - y) \cdot x_1 + \dots + (h_\theta(x) - y) \cdot x_n)}$ $= \cancel{\underline{a_j}}$
---	--

따라서 For a single training example

$$\theta_j := \theta_j + \alpha (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}.$$

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^m (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)} \quad (\text{for every } j).$$

}

해당 룰을 우리는 **LMS update** (least mean squares), Widrow-Hoss learning rule 이라고 함.

- error term (training data에 대한 학습 예측값과 정답 label 간의 차이) 이 클 수록 θ 가 많이 변경되고,
- error term 이 작을수록 θ 는 작은 비율로 변경됨.

LMS algorithm

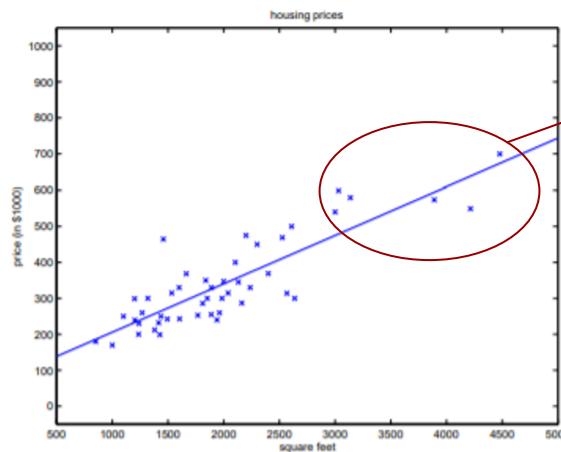
```
Repeat until convergence {  
     $\theta_j := \theta_j + \alpha \sum_{i=1}^m (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}$       (for every  $j$ ).  
}
```

- **batch gradient descent**

- 현재, 한 번의 θ 업데이트를 위해서 전체 training set 사용. 이를 batch gradient descent 라고 함.

- **stochastic gradient descent (incremental gradient descent)**

- θ 를 한 번 업데이트 하기 위하여, 전체 training dataset 를 사용하는 것이 아니라, 일부 training set (training example) 를 사용하여 업데이트.



를 결정하는 것이 **batch size**

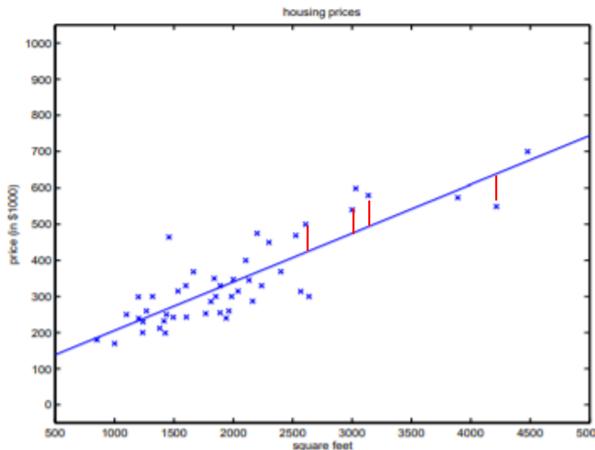
예를 들어 6개의 training example 계산을 통해서 θ 를 업데이트 한다면 => batch size = 6

참고)

- epoch : 전체 데이터 셋에 대해 한 번 학습을 완료한 상태
- batch : 한 번의 batch 마다 주는 데이터 샘플의 size
- iteration : epoch 를 나누어서 실행하는 횟수

다시 처음으로 돌아가서, 우리의 궁극적인 목표는 **model generalization**. 즉, 새로운 데이터가 입력되었을 때, 이에 대한 예측을 가장 잘 하는 모델.

- 1) θ 를 업데이트 한다는 것은 모델의 그래프를 변경하다는 뜻!
- 1) 모델의 그래프는 $J(\theta)$ (cost function) 이 작은 형태로 변경이 되어야 하는데
 - a) 중요한 것은, $J(\theta)$ 가 항상 작은 값이 될 필요는 없다! 우리에게 가장 중요한 것은 다음에 올 데이터에 대한 예측을 잘 하는 것이지, train dataset 에 대해서 예측을 잘 하는 것이 아니기 때문! 즉, model generalization 이 궁극적인 목표.



$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2.$$

θ 를 업데이트 할 때마다 training dataset 을 더 잘 표현하는 모델 그래프를 찾을 수 있음.

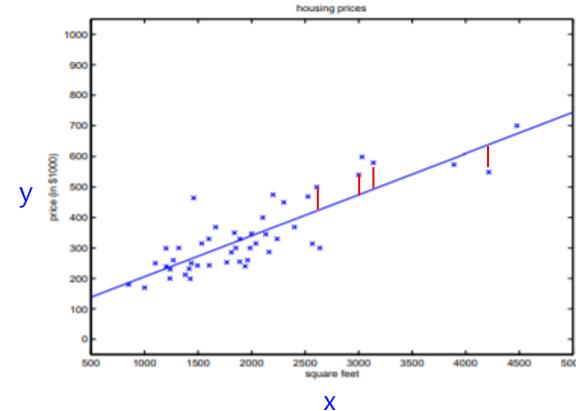
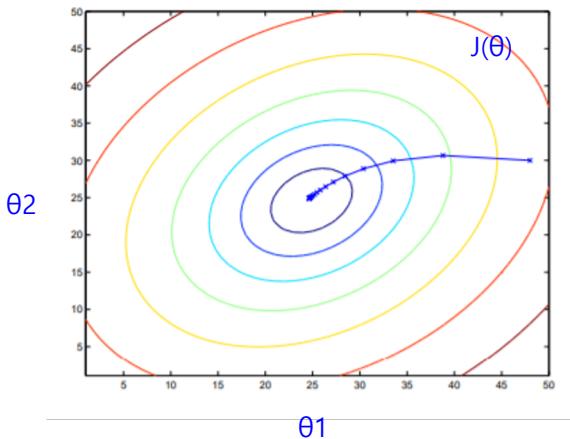
해당 그래프는 n (feature 개수) = 1.

- 왜냐하면 n 의 개수에 따라 θ 의 개수가 정해지는데 (n 개수 + 1 (intercept term)) = θ 개수)
 θ 가 3개 이상이면 그림으로써 표현 불가능. 즉, 이미지를 용이하게 보기 위해 feature 를 2개로 선정한 것. (다음 슬라이드)

Part I. Linear Regression

다시 처음으로 돌아가서, 우리의 궁극적인 목표는 **model generalization**. 즉, 새로운 데이터가 입력되었을 때, 이에 대한 예측을 가장 잘 하는 모델.

- 1) θ 를 업데이트 한다는 것은 모델의 그래프를 변경하다는 뜻!
- 1) 모델의 그래프는 $J(\theta)$ (cost function) 이 작은 형태로 변경이 되어야 하는데
 - a) 중요한 것은, $J(\theta)$ 가 항상 작은 값이 될 필요는 없다! 우리에게 가장 중요한 것은 다음에 올 데이터에 대한 예측을 잘 하는 것이지, train dataset 에 대해서 예측을 잘 하는 것이 아니기 때문! 즉, model generalization 이 궁극적인 목표.



$$h_{\theta}(x) = \theta_0 + \theta_1 x_1$$
$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2.$$

- 1) θ_1 과 θ_2 에 따른 $J(\theta)$ 의 값이고, (θ_1, θ_2 변화에 따른 $J(\theta)$)
- 2) θ_1 과 θ_2 에 따른 모델의 그래프이다. (θ_1, θ_2 로 모델이 파라미터화 된 상태)

왼쪽 이미지에서 $J(\theta)$ 값이 가장 작은 위치의 θ_1, θ_2 를 선정 했을 때, 오른쪽 이미지에서 training dataset 을 가장 잘 표현하는 모델이 결정되는 것.

2. The normal equations (정규 방정식)

Gradient descent 는 $J(\theta)$ 를 최소화 할 수 있는 하나의 방법 => 본 장에서는 최적의 θ 값을 반복되는 알고리즘 없이 한 번에 계산하는 방법 기술.

$$X = \begin{bmatrix} -(x^{(1)})^T \\ -(x^{(2)})^T \\ \vdots \\ -(x^{(m)})^T \end{bmatrix}, \quad \vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}.$$


Now, since $h_{\theta}(x^{(i)}) = (x^{(i)})^T \theta$, we can easily verify that

$$\begin{aligned} X\theta - \vec{y} &= \begin{bmatrix} (x^{(1)})^T \theta \\ \vdots \\ (x^{(m)})^T \theta \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \\ &= \begin{bmatrix} h_{\theta}(x^{(1)}) - y^{(1)} \\ \vdots \\ h_{\theta}(x^{(m)}) - y^{(m)} \end{bmatrix}. \end{aligned}$$

$$\begin{aligned} \nabla_{\theta} J(\theta) &= \nabla_{\theta} \frac{1}{2} (X\theta - \vec{y})^T (X\theta - \vec{y}) \\ &= \frac{1}{2} \nabla_{\theta} (\theta^T X^T X \theta - \theta^T X^T \vec{y} - \vec{y}^T X \theta + \vec{y}^T \vec{y}) \\ &= \frac{1}{2} \nabla_{\theta} \text{tr} (\theta^T X^T X \theta - \theta^T X^T \vec{y} - \vec{y}^T X \theta + \vec{y}^T \vec{y}) \\ &= \frac{1}{2} \nabla_{\theta} (\text{tr} \theta^T X^T X \theta - 2 \text{tr} \vec{y}^T X \theta) \\ &= \frac{1}{2} (X^T X \theta + X^T X \theta - 2 X^T \vec{y}) \\ &= X^T X \theta - X^T \vec{y} = \mathbf{0} \end{aligned}$$

Thus, the value of θ that minimizes $J(\theta)$ is given in closed form by the equation

$$\theta = (X^T X)^{-1} X^T \vec{y}.$$

3. Probabilistic interpretation (확률론적 접근)

본 섹션에서, 우리가 LSM (least-squares) cost function J 를 사용한 것에 대한 확률론인 관점으로 접근하겠습니다.

자, 이제부터 input (x) 와 target variables 이 다음의 관계를 가진다고 가정하자.

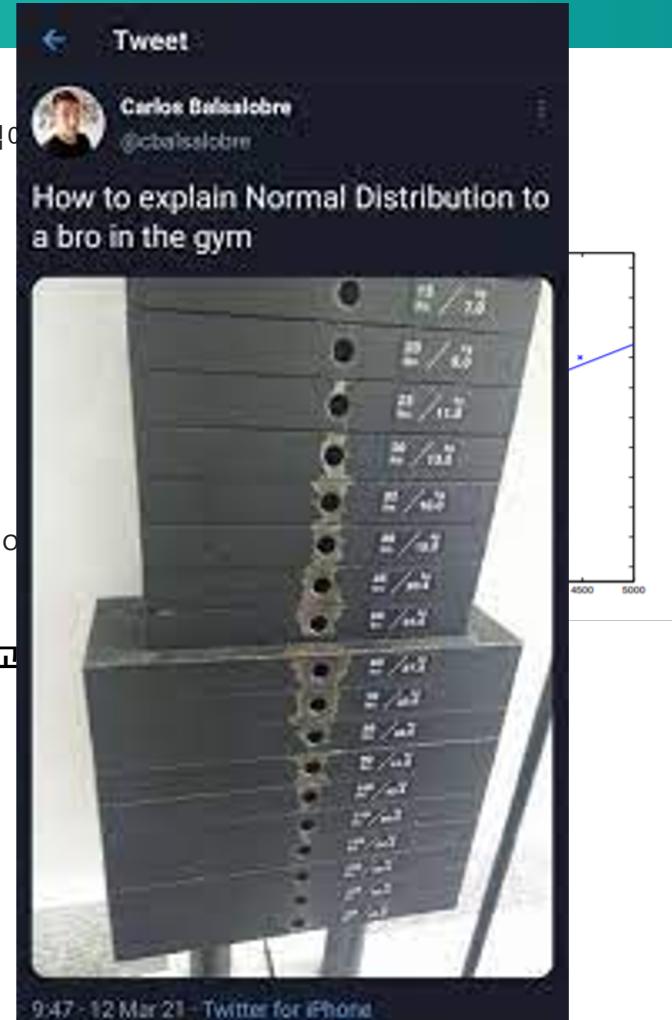
$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

ϵ : error term

- training dataset 에 대한 예측 값과 정답 label 간의 차이.
- Other things have an impact on housing prices. (i.e. some random force of nature)

ϵ 은 IID (independently and identically distributed, 독립 항등 분포) 를 따른다고 가정합니다. 그러면 ϵ 는 표준 정규 분포를 표현한 분포이기 때문.

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right).$$



3. Probabilistic interpretation (확률론적 접근)

$$p(y^{(i)}|x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right).$$
 $p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right).$

θ 로 파라미터화 된 모델에서 $x^{(i)}$ 가 주어졌을 때 (given), $y^{(i)}$ 가 나올 확률 => 정규 분포를 따른다.

$y^{(i)}$ given $x^{(i)}$ and parameterized by θ

- 정규분포 : transpose(θ) x 를 평균으로 하고 시그마를 분산으로 하는 정규분포.

$$y^{(i)} | x^{(i)}; \theta \sim \mathcal{N}(\theta^T x^{(i)}, \sigma^2).$$

x 와 θ 가 주어졌을 때의 y 의 확률적 분포를 나타 $p(\vec{y}|X; \theta)$.

=> θ 에 대한 함수로 표현한다면 다음의 likelihood function $L(\theta; X, \vec{y})$

$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X; \theta).$$

Bayesian Statistics (베이지안 통계학)

- 사건 A 가 일어났을 때의 확률을 계산함에 있어서, 사건 B 가 일어났을 때의 확률들로 표현할 수 있음.
- 즉, A가 조건으로 주어졌을 때 B의 확률에 대해서 궁금했던 것을 반대로, B가 조건으로 주어졌을 때의 A의 확률에 대해서 이야기하는 것으로 바꾸어 쓸 수 있다.

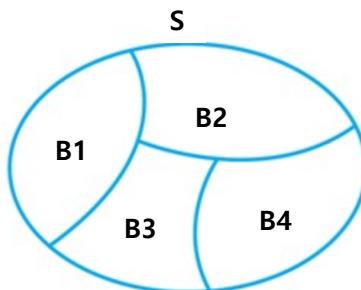
3. Probabilistic interpretation (확률론적 접근)

$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X; \theta).$$

사건 B 사건 A 사건 A 사건 B

Bayesian Statistics (베이지안 통계학)

- 사건 A가 일어났을 때의 확률을 계산함에 있어서, 사건 B가 일어났을 때의 확률들로 표현할 수 있음.
- 즉, A가 조건으로 주어졌을 때 B의 확률에 대해서 궁금했던 것을 반대로, B가 조건으로 주어졌을 때의 A의 확률에 대해서 이야기하는 것으로 바꾸어 쓸 수 있다.



Bayes' Law (베이즈 정리)

- 1) k 개의 집합 B_1, B_2, \dots, B_k 가 어떤 사건 S 의 분할. 그러면 모든 S 의 부분집합 A 에 대해서 다음의 식 성

$$A = (A \cap B_1) \cup \dots \cup (A \cap B_K)$$

- 2) 괄호 안의 사건들이 각각 서로소이기 때문에 사건 A 가 발생할 확률은 다음과 같음.

$$P(A) = P(A \cap B_1) + \dots + P(A \cap B_K)$$

- 3) 따라서, 어떤 A 라는 사건이 일어났을 때, B_j 라는 사건이 일어날 조건부 확률은 다음과 같음.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad | \quad P(B|A) = \frac{P(A \cap B)}{P(A)} \quad | \quad P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

$$P(B_j|A) = \frac{P(A|B_j)P(B_j)}{P(A)} = \frac{P(A|B_j)P(B_j)}{P(A|B_1)P(B_1) + \dots + P(A|B_K)P(B_K)}$$

3. Probabilistic interpretation (확률론적 접근)

$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X; \theta). \quad p(y^{(i)}|x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right).$$



$$\begin{aligned} L(\theta) &= \prod_{i=1}^m p(y^{(i)} | x^{(i)}; \theta) \\ &= \prod_{i=1}^m \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right). \end{aligned}$$

이제, x와 y에 대한 probabilistic model 이 성립되었고, 최적의 θ 를 선택하는 가장 좋은 방법은?

=> **maximum likelihood** : data(x, y)가 가장 높은 확률을 가지는 θ 를 선택. 즉, $L(\theta)$ 를 최대로 하는 θ .

$L(\theta)$ 를 미분하는 것보다 (최대값을 찾는 것보다) $\log L(\theta)$ 를 미분하는 편이 간편.

$$\begin{aligned} \ell(\theta) &= \log L(\theta) \\ &= \log \prod_{i=1}^m \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\ &= \sum_{i=1}^m \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \end{aligned}$$

$$\frac{1}{2} \sum_{i=1}^m (y^{(i)} - \theta^T x^{(i)})^2, \Rightarrow \text{최대가 되어야 함.}$$

즉, 우리가 LSM (least-squares) cost function J 를 사용한 것에 대해서
확률론인 관점에서 해석함! (very natural algorithm)

$$= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^m (y^{(i)} - \theta^T x^{(i)})^2. \Rightarrow \text{최소}$$

4. Locally weighted linear regression (LWR)

학습	설명	종류	알고리즘
Supervised Learning	<ul style="list-style-type: none"> 정답 (label) 이 무엇인지 알려주며 모델 학습하는 방법 정확한 input과 output 존재 	<ul style="list-style-type: none"> Classification Regression 	<ul style="list-style-type: none"> Classification : kNN, Support Vector (Part II) Regression : Linear Regression, Locally Weighted Linear (Part I)

예측할 입력 데이터 x 에 가까운 학습 데이터들에 더 많은 가중치를 둘으로써, 곡선의 적합을 수행하는 학습 방법.

일반적인 linear regression 알고리즘에서, query point x 에 대한 prediction 을 수행하기 위해서,

1. Fit θ to minimize $\sum_i (y^{(i)} - \theta^T x^{(i)})^2$.
2. Output $\theta^T x$.

반면, LWR 알고리즘에서는 다음의 방식 사용.

1. Fit θ to minimize $\sum_i w^{(i)} (y^{(i)} - \theta^T x^{(i)})^2$.
2. Output $\theta^T x$.

query point x 에 가까울 수록, 더 많은 가중치 $w^{(i)}$

$$w^{(i)} = \exp \left(-\frac{(x^{(i)} - x)^2}{2\tau^2} \right)$$

4. Locally weighted linear regression (LWR)

예측할 입력 데이터 x 에 가까운 학습 데이터들에 더 많은 가중치를 둘으로써, 곡선의 적합을 수행하는 학습 방법.

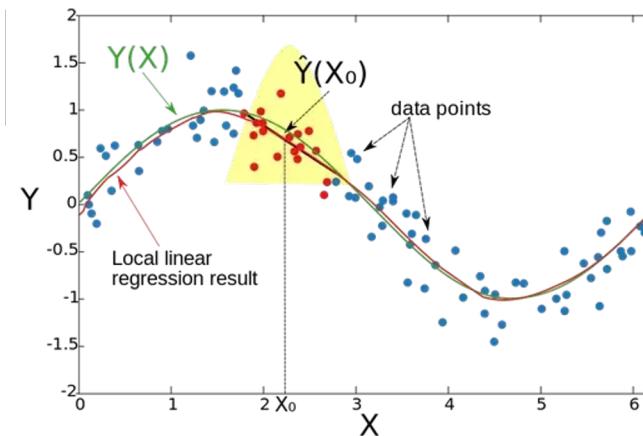
1. Fit θ to minimize $\sum_i w^{(i)}(y^{(i)} - \theta^T x^{(i)})^2$.

2. Output $\theta^T x$. query point x 에 가까울 수록, 더 많은 가중치를 둠.

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

 $w^{(i)}$

- non-negative valued weights
- 가우시안 분포 아님.



$|x^{(i)} - x|$ 이 클수록 = 거리가 멀수록
 ● $w^{(i)}$ 는 작아짐
 ● = error term 을 적게 수용함.

$|x^{(i)} - x|$ 이 작을수록 = 거리가 가까울수록
 ● $w^{(i)}$ 커짐.
 ● = error term 을 많이 수용함.

4. Locally weighted linear regression (LWR)

예측할 입력 데이터 x 에 가까운 학습 데이터들에 더 많은 가중치를 둘으로써, 곡선의 적합을 수행하는 학습 방법.

$$1. \text{ Fit } \theta \text{ to minimize } \sum_i w^{(i)} (y^{(i)} - \theta^T x^{(i)})^2.$$

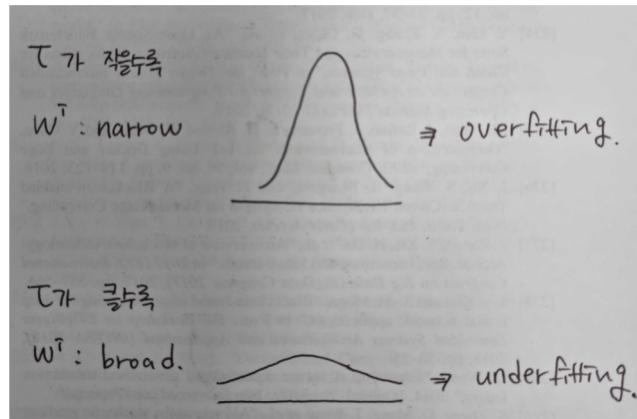
$$2. \text{ Output } \theta^T x.$$

query point x 에 가까울 수록, 더 많은 가중치를 둠.

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

bandwidth parameter

- 타우에 따라 bell-shaped 커브가 fatter, thinner 선택할 수 있음.



LWR 은 non-parametric algorithm

- 모집단의 형태와 관계없이, 주어진 데이터에서 직접 확률을 계산하여 통계적으로 검정하는 분석 방식.
- 기존의 unweighted linear regression 알고리즘은 fixed, finite 의 파라미터를 가진 **parametric learning** 알고리즘.
- 한번 θ 를 결정하면 prediction 을 위해 training data 를 저장할 필요 없는 parametric learning algorithm 과는 달리, LWR 에서 prediction 을 수행하기 위해서 전체 training data 필요함.

- Exponential family

- 다음의 형식으로 작성될 수 있다면, exponential family
구성

$$p(y; \eta) = b(y) \exp(\eta^T T(y) - a(\eta))$$

확률밀도함수가 적절한 함수 $a, b, c_i, t_i (i=1, \dots, k)$ 에 대하여,

$f(x; \theta) = a(\theta)b(x)\exp\left[\sum_{i=1}^k c_i(\theta)t_i(x)\right], \quad -\infty < x < \infty, \quad \theta = (\theta_1, \theta_2, \dots, \theta_k)$ 로 표현되면 이를 k 개의 모수 $\theta_1, \theta_2, \dots, \theta_k$ 를 가진 지수족 (Exponential Family)에 속한다고 정의한다.

- notation

- y : data
- η : **natural parameter (canonical parameter)**
- $T(y)$: sufficient statistic ($=y$)
- $b(y)$: base measure (y 에 관한 식, η 포함 X)
- $a(\eta)$: log partition function (η 와 관련된 함수, y 포함 X , 결정적 역할 X)

- Exponential family에 속하는 확률밀도함수 (PDF; probability density function)

- 베르누이 - Binary**
- 이항분포
- 포아송분포 - Count
- 지수분포
- 기하분포
- 정규분포 - Real**
- 균등분포
- 감마분포

Part III. Generalized Linear Models

$$p(y; \eta) = b(y) \exp(\eta^T T(y) - a(\eta))$$

- Exponential family에 속하는 확률밀도함수 (PDF; probability density function)
 - 베르누이 - Binary

We write the Bernoulli distribution as:

$$\begin{aligned} p(y; \phi) &= \phi^y (1 - \phi)^{1-y} \\ &= \exp(y \log \phi + (1 - y) \log(1 - \phi)) \\ &= \exp\left(\left(\log\left(\frac{\phi}{1-\phi}\right)\right)y + \log(1 - \phi)\right). \end{aligned}$$

Φ : probability of event

$$\begin{aligned} T(y) &= y \\ a(\eta) &= -\log(1 - \phi) \\ &= \log(1 + e^{-\eta}) \\ b(y) &= 1 \end{aligned}$$

$$\phi = \frac{1}{1 + e^{-\eta}}$$

sigmoid function,
logistic regression.

- 정규분포 - Real

$$\begin{aligned} p(y; \mu) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y - \mu)^2\right) \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \cdot \exp\left(\mu y - \frac{1}{2}\mu^2\right) \end{aligned}$$

μ : probability of event

$$\begin{aligned} \eta &= \mu \\ T(y) &= y \\ a(\eta) &= \mu^2/2 \\ &= \eta^2/2 \\ b(y) &= (1/\sqrt{2\pi}) \exp(-y^2/2). \end{aligned}$$

- 분산 값 (σ^2)은 Θ 와 $h(x)$ 의 선택에 영향을 미치지 않음.
 - $\sigma^2 = 1$ 로 고정.

Part III. Generalized Linear Models

$$p(y; \eta) = b(y) \exp(\eta^T T(y) - a(\eta))$$

- Exponential family 사용하는 이유: have some nice mathematical properties.

(1) MLE (Maximum Likelihood Estimation) with respect to η is concave.

- exponential family 가 natural parameter로 parameterized 되어 있을 때, optimization problem is concave.
- 즉, cost function (doing maximum likelihood) is convex.

$$(1) E[y : \eta] = \frac{\partial}{\partial \eta} \cdot a(\eta)$$

$$\text{Var}[y : \eta] = \frac{\partial^2}{\partial \eta^2} a(\eta)$$

Part III. Generalized Linear Models

we can actually **build** a lot of **many powerful models** by choosing an appropriate family in the exponential fa

- Exponential family에 속하는 확률밀도함수 (PDF; probability density function)
 - 베르누이 - Binary
 - 이항분포
 - 포아송분포 - Count
 - 지수분포
 - 기하분포
 - 정규분포 - Real
 - 균등분포
 - 김마분포

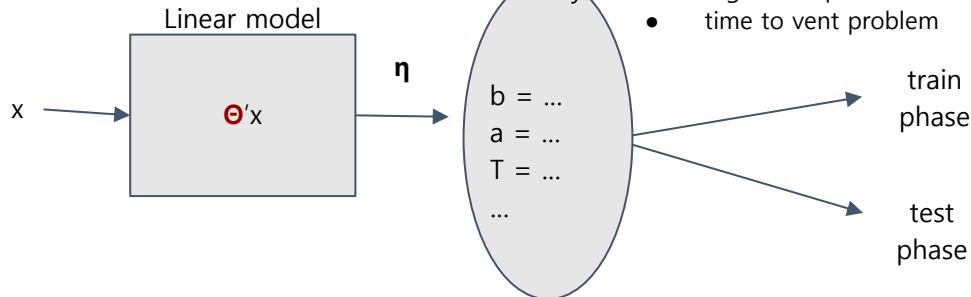
GLM

Assumptions & Design choices

$y | x; \theta \sim \text{ExponentialFamily}(\eta)$. I.e., given x and θ , the distribution of y follows some exponential family distribution, with parameter η .

The natural parameter η and the inputs x are related linearly: $\eta = \theta^T x$.
 (Or, if η is vector-valued, then $\eta_i = \theta_i^T x$.)

- (1) Test time : output $E[y|x;\Theta]$
 $\Rightarrow h\Theta(x) = E[y|x;\Theta]$



- Learning update rule

```
Loop {
    for i=1 to m, {
         $\theta_j := \theta_j + \alpha (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}$  (for every j).
    }
}
```

$$\max_{\theta} \log p(y^{(i)}; \theta^T x^{(i)})$$

$$E[y;\eta] = E[y;\theta^T x] = h\theta(x)$$

Part III. Generalized Linear Models

Softmax Regression \Rightarrow express the multinomial as an exponential family distribution

- 분류 문제

- y 값으로 k 개 존재. (e.g. 이메일 - 스팸, 개인 메일, 업무 관련된 이메일 기타 등)
- 다항 데이터에 대해 GLM 도출. \Rightarrow 다항 분포를 Exponential family 로 표현할 수 있음.

We write the Bernoulli distribution as:

Φ : probability of event

$$\begin{aligned} p(y; \phi) &= \phi^y (1 - \phi)^{1-y} \\ &= \exp(y \log \phi + (1 - y) \log(1 - \phi)) \\ &= \exp\left(\left(\log\left(\frac{\phi}{1 - \phi}\right)\right)y + \log(1 - \phi)\right). \end{aligned}$$

$$\begin{aligned} T(y) &= y \\ a(\eta) &= -\log(1 - \phi) \\ &= \log(1 + e^{-\eta}) \\ b(y) &= 1 \end{aligned}$$

$$\phi = \frac{1}{1 + e^{-\eta}}$$

sigmoid function,
logistic regression.

$\Rightarrow \Phi_1$: probability of event 1

$\Rightarrow \Phi_2$: probability of event 2

Φ 는 각각 독립적이지 않기 때문에, Φ_k 를 다음과 같이 표현.

$$\dots$$

$$\phi_k = 1 - \sum_{i=1}^{k-1} \phi_i,$$

$\Rightarrow \Phi_k$: probability of event k

기존의 $T(y) = y$ (real number) 과는 다르게 define $T(y) \in \mathbb{R}^{k-1}$

$$T(1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, T(2) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, T(3) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots, T(k-1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, T(k) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$E[(T(y))_i] = P(y = i) = \phi_i.$$

각각의 $T(1), T(2), T(3)$ 이 발생할 확률은
 Φ_1, Φ_2, Φ_3 로 표현 가능.

Part III. Generalized Linear Models

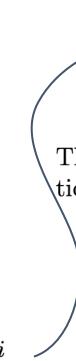
show that the multinomial is a member of the exponential family.

$$\begin{aligned}
 p(y; \phi) &= \phi_1^{1\{y=1\}} \phi_2^{1\{y=2\}} \cdots \phi_k^{1\{y=k\}} \\
 &= \phi_1^{1\{y=1\}} \phi_2^{1\{y=2\}} \cdots \phi_k^{1-\sum_{i=1}^{k-1} 1\{y=i\}} \\
 &= \phi_1^{(T(y))_1} \phi_2^{(T(y))_2} \cdots \phi_k^{1-\sum_{i=1}^{k-1} (T(y))_i} \\
 &= \exp((T(y))_1 \log(\phi_1) + (T(y))_2 \log(\phi_2) + \\
 &\quad \cdots + (1 - \sum_{i=1}^{k-1} (T(y))_i) \log(\phi_k)) \\
 &= \exp((T(y))_1 \log(\phi_1/\phi_k) + (T(y))_2 \log(\phi_2/\phi_k) + \\
 &\quad \cdots + (T(y))_{k-1} \log(\phi_{k-1}/\phi_k) + \log(\phi_k)) \\
 &= b(y) \exp(\eta^T T(y) - a(\eta))
 \end{aligned}$$

where

$$\eta = \begin{bmatrix} \log(\phi_1/\phi_k) \\ \log(\phi_2/\phi_k) \\ \vdots \\ \log(\phi_{k-1}/\phi_k) \end{bmatrix}, \quad \eta_i = \log \frac{\phi_i}{\phi_k}.$$

$$\begin{aligned}
 a(\eta) &= -\log(\phi_k) \\
 b(y) &= 1.
 \end{aligned}$$



$$\begin{aligned}
 e^{\eta_i} &= \frac{\phi_i}{\phi_k} \\
 \phi_k e^{\eta_i} &= \phi_i \\
 \phi_k \sum_{i=1}^k e^{\eta_i} &= \sum_{i=1}^k \phi_i = 1
 \end{aligned} \tag{7}$$

This implies that $\phi_k = 1 / \sum_{i=1}^k e^{\eta_i}$, which can be substituted back into Equation (7) to give the response function

$$\phi_i = \frac{e^{\eta_i}}{\sum_{j=1}^k e^{\eta_j}}$$

softmax function

Part III. Generalized Linear Models

show that the multinomial is a member of the exponential family.

$$\begin{aligned} e^{\eta_i} &= \frac{\phi_i}{\phi_k} \\ \phi_k e^{\eta_i} &= \phi_i \\ \phi_k \sum_{i=1}^k e^{\eta_i} &= \sum_{i=1}^k \phi_i = 1 \end{aligned} \tag{7}$$

This implies that $\phi_k = 1 / \sum_{i=1}^k e^{\eta_i}$, which can be substituted back into Equation (7) to give the response function

$$\phi_i = \frac{e^{\eta_i}}{\sum_{j=1}^k e^{\eta_j}}$$

softmax function

- To complete our model,

$$\begin{aligned} p(y = i|x; \theta) &= \phi_i \\ &= \frac{e^{\eta_i}}{\sum_{j=1}^k e^{\eta_j}} \\ &= \frac{e^{\theta_i^T x}}{\sum_{j=1}^k e^{\theta_j^T x}} \end{aligned}$$

This model, which applies to classification problems where $y \in \{1, \dots, k\}$, is called **softmax regression**. It is a generalization of logistic regression.

Our hypothesis will output

$$\begin{aligned} h_\theta(x) &= E[T(y)|x; \theta] \\ &= E \left[\begin{array}{c} 1\{y=1\} \\ 1\{y=2\} \\ \vdots \\ 1\{y=k-1\} \end{array} \middle| x; \theta \right] \\ &= \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{k-1} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\exp(\theta_1^T x)}{\sum_{j=1}^k \exp(\theta_j^T x)} \\ \frac{\exp(\theta_2^T x)}{\sum_{j=1}^k \exp(\theta_j^T x)} \\ \vdots \\ \frac{\exp(\theta_{k-1}^T x)}{\sum_{j=1}^k \exp(\theta_j^T x)} \end{bmatrix}. \end{aligned}$$

**End of the
Document**