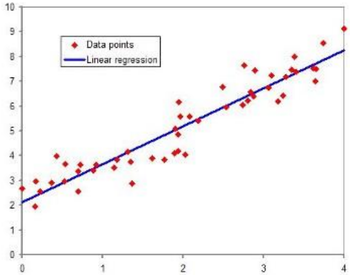
회귀분석 (Regression)

관찰된 연속형 변수들에 대해 **두 변수 사이의 모형**을 구한뒤 **적합도를 측정**해 내는 분석 방법 (출처: 위키피디아)

선형회귀분석 (Linear Regression): 종속 변수 y와 한 개 이상의 **독립 변수 (또는 설명 변수)** X와의 **선형 상관 관계를 모델링**하는 회귀분석 기법 (출처: 위키피디아)



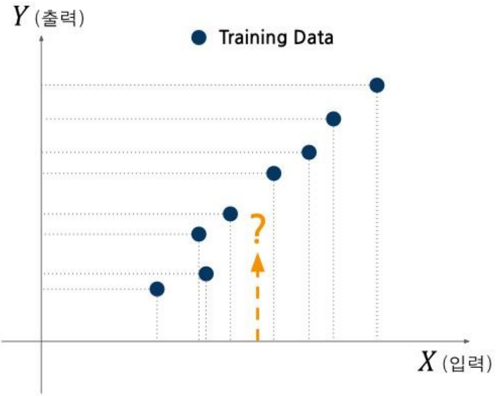
Linear Regression

Training Data

```
[1.2] → 1.1
[3.2] → 2.7
[2.8] → 2.8
[1.2] → 0.9
[4.2] → -0.1
  …
[3.2] → -0.2
```

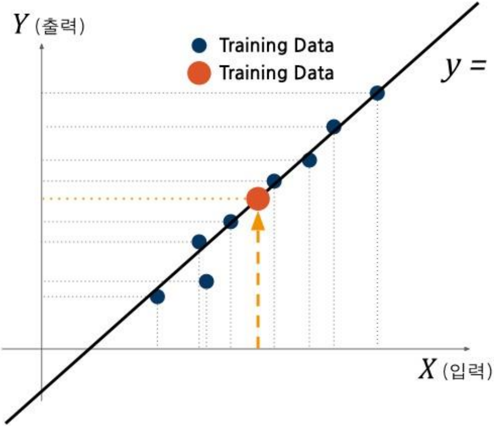
Test

```
[1.3] → ?
```



Linear Regression

이 **선**만 구할 수 있으면,
어떤 **입력**이 들어와도
출력을 **예측**할 수 있다!



$y = ax + c$

선을 구한다
= a 값과 c 값을 구한다

어떻게 구하지..?

Linear Regression

입력이 복잡한 지도학습…

Training Data

```
[1.2,  3.8, -1.4, …,  4.1] →  1.1
[3.2, -1.2, -0.2, …,  2.1] →  2.7
[2.8, -1.4, -0.3, …,  2.3] →  2.8
[1.2,  3.4, -1.5, …,  4.2] →  0.9
[4.2,  2.1,  2.8, …, -0.5] → -0.1
                   …
[3.2,  2.2,  2.2, …, -0.4] → -0.2
```

Test

```
[1.3,  3.2, -1.5, …,  4.1] → ?
```

쉬운 설명을 위해…
단순한 예제.

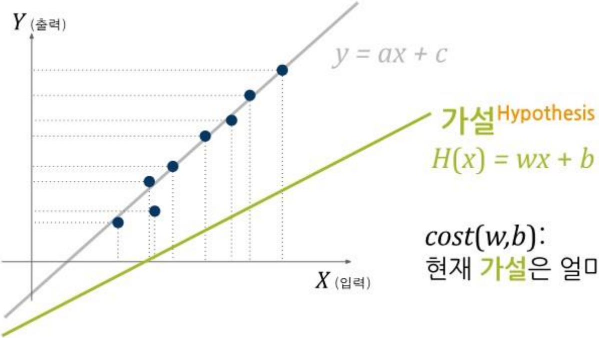
Training Data

```
[1.2] →  1.1
[3.2] →  2.7
[2.8] →  2.8
[1.2] →  0.9
[4.2] → -0.1
   …
[3.2] → -0.2
```

Test

```
[1.3] → ?
```

가설과 비용 Hypothesis and Cost function



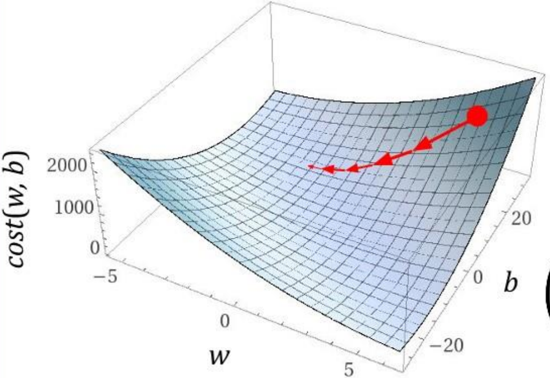
$H(x) = wx + b$

$cost(w,b)$:
현재 가설은 얼마나 잘못되었는가?

경사 하강법 Gradient Descent

우리의 목표: cost를 최소화 하자! = cost를 최소로 만드는 w, b 값을 찾자!

$$\arg\min_{w,b} cost(w,b)$$



경사 따라 내려가야하는데,
경사는 어떻게 구하지?
⇒ **편미분** 이용

경사:

$$\left(\frac{\partial cost(w,b)}{\partial w}, \frac{\partial cost(w,b)}{\partial b}\right)$$

경사 하강법 Gradient Descent

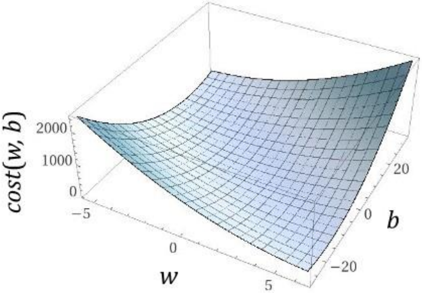
경사: $\left(\frac{\partial cost(w,b)}{\partial w}, \frac{\partial cost(w,b)}{\partial b} \right)$

업데이트:

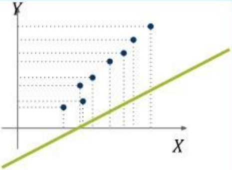
Learning Rate

$$w = w - \alpha \frac{\partial cost(w,b)}{\partial w}$$

$$b = b - \alpha \frac{\partial cost(w,b)}{\partial b}$$



Linear Regression (2)



입력이 조금 더 복잡할 때?

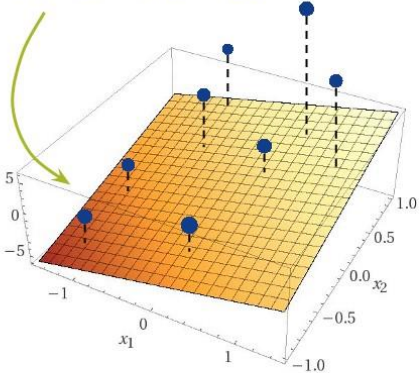
Training Data

```
[1.2,  3.8] →  1.1
[3.2, -1.2] →  2.7
[2.8, -1.4] →  2.8
[1.2,  3.4] →  0.9
[4.2,  2.1] → -0.1
          ...
[3.2,  2.2] → -0.2
```

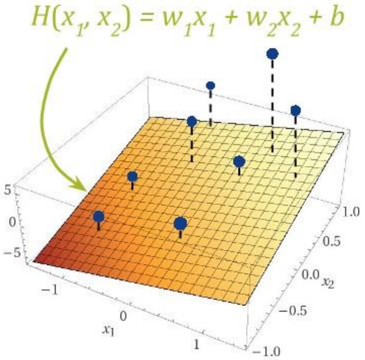
Test

```
[1.3,  3.2] → ?
```

$$H(x_1, x_2) = w_1x_1 + w_2x_2 + b$$



가설과 비용 (2) Hypothesis and Cost function (2)



$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$H(x_1, x_2) = H(\mathbf{x}) = \mathbf{w}^\mathsf{T}\mathbf{x} + b$$

$$cost(\mathbf{w}, b) = \frac{1}{n}\sum_{i=0}^{n}\left(y_i - H(\mathbf{x}_i)\right)^2$$

경사 하강법 (2) Gradient Descent (2)

우리의 목표: cost를 최소화 하자! = cost를 최소로 만드는 w, b 값을 찾자!

$$\arg\min_{\mathbf{w},b} cost(\mathbf{w}, b)$$

업데이트:

경사 (Gradient)

Learning Rate

$$\left.\begin{aligned} w_1 &= w_1 - \alpha \frac{\partial cost(\mathbf{w},b)}{\partial w_1} \\ w_2 &= w_2 - \alpha \frac{\partial cost(\mathbf{w},b)}{\partial w_2} \end{aligned}\right\} \quad \mathbf{w} = \mathbf{w} - \alpha \frac{\partial cost(\mathbf{w},b)}{\partial \mathbf{w}}$$

$$b = b - \alpha \frac{\partial cost(\mathbf{w},b)}{\partial b}$$

Linear Regression (3)

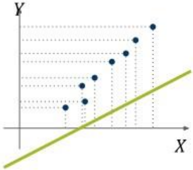
입력이 조금 더 더 복잡할 때?

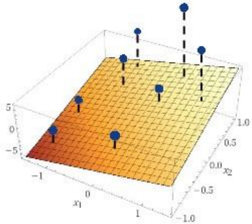
Training Data

```
[1.2,  3.8, -1.4, …,  4.1] →  1.1
[3.2, -1.2, -0.2, …,  2.1] →  2.7
[2.8, -1.4, -0.3, …,  2.3] →  2.8
[1.2,  3.4, -1.5, …,  4.2] →  0.9
[4.2,  2.1,  2.8, …, -0.5] → -0.1
                  …
[3.2,  2.2,  2.2, …, -0.4] → -0.2
```

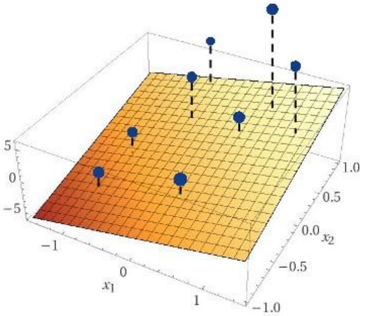
Test

```
[1.3,  3.2, -1.5, …,  4.1] → ?
```





가설과 비용 (3) Hypothesis and Cost function (3)



$$\mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$$

$$H(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

$$cost(\mathbf{w}, b) = \frac{1}{n} \sum_{i=0}^{n} \left(y_i - H(\mathbf{x}_i)\right)^2$$

경사 하강법 (3) Gradient Descent (3)

우리의 목표: cost를 최소화 하자! = cost를 최소로 만드는 w, b 값을 찾자!

$$\arg\min_{\mathbf{w},b} cost(\mathbf{w}, b)$$

업데이트:

$$\mathbf{w} = \mathbf{w} - \alpha \frac{\partial cost(\mathbf{w},b)}{\partial \mathbf{w}}$$

$$b = b - \alpha \frac{\partial cost(\mathbf{w},b)}{\partial b}$$

CSC 311: Introduction to Machine Learning

Lecture 3 - Linear Classifiers, Logistic Regression, Multiclass Classification

Roger Grosse   Chris Maddison   Juhan Bae   Silviu Pitis

University of Toronto, Fall 2020

Overview

- 분류: 이산 값 대상 예측
) 이진 분류: 이진 값 대상 예측
) 다중 클래스 분류: 이산(> 2) 값 대상 예측

- 이진 분류의 예
) 다양한 증상의 존재 또는 부재를 감안하여 환자가 질병을 앓고 있는지 예측
) 이메일을 스팸 또는 비스팸으로 분류
) 금융 거래가 사기인지 예측

이진 선형 분류

분류: D차원 입력 x ∈ RD가 주어지면 이산 값 대상을 예측합니다.

이진: 이진 대상 t ∈ {0, 1}을 예측합니다.

) t = 1인 학습 예제를 양성 예제라고 하고, t = 0인 학습 예제를 음성 예제라고 합니다. 죄송합니다.

) t ∈ {0, 1} 또는 t ∈ {−1, +1}은 계산 편의를 위한 것입니다.

선형: 모델 예측 y는 x의 선형 함수이며, 그 뒤에 임계값 r이 붙습니다.

$$z = \mathbf{w}^T\mathbf{x} + b$$

$$y = \begin{cases} 1 & \text{if } z \geq r \\ 0 & \text{if } z < r \end{cases}$$

임계값 제거

일반성을 잃지 않고(WLOG) 임계값 r = 0이라고 가정할 수 있습니다.

$$\mathbf{w}^T\mathbf{x} + b \geq r \iff \mathbf{w}^T\mathbf{x} + b - r \geq 0.$$

절편 제거

- 항상 값 1을 취하는 더미 기능 x0을 추가합니다.
  가중치 w0 = b는 편향과 동일합니다(선형 회귀와 동일).

- **Simplified model**
  Receive input $\mathbf{x} \in \mathbb{R}^{D+1}$ with $x_0 = 1$:

$$z = \mathbf{w}^T\mathbf{x}$$

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

Examples

NOT

x_0	x_1	t
1	0	1
1	1	0

- 이것이 더미 피처 x0가 포함된 우리의 훈련 세트라고 가정해 봅시다.
- w0, w1에 대한 어떤 조건이 완벽한 분류를 보장합니까?
) x1 = 0일 때, 필요: $z = w0x0 + w1x1 \geq 0 \Longleftrightarrow w0 \geq 0$
) x1 = 1일 때, 필요: $z = w0x0 + w1x1 < 0 \Longleftrightarrow w0 + w1 < 0$
- 예시 솔루션: w0 = 1, w1 = −2 이것이 유일한 솔루션입니까?
-

Examples

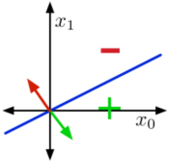
AND

x_0	x_1	x_2	t
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

z = w0x0 + w1x1 + w2x2
필요: w0 < 0
필요: w0 + w2 < 0
필요: w0 + w1 < 0
필요: w0 + w1 + w2 ≥ 0

예시 솔루션: w0 = −1.5, w1 = 1, w2 = 1

Input Space, or Data Space for NOT example



x_0	x_1	t
1	0	1
1	1	0

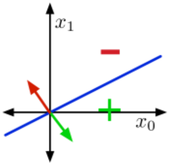
- 훈련 예제는 점입니다.
- 가중치(가설) w는 반공간으로 표현될 수 있습니다.

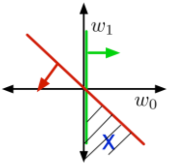
  H+ = {x : wTx ≥ 0}, H— = {x : wTx < 0}

) 이러한 반공간의 경계는 원점을 통과합니다(왜?)
- 경계는 결정 경계입니다. {x : wTx = 0}

) 2차원에서는 선이지만 고차원에서는 초평면입니다.
- 선형 결정 규칙으로 훈련 예제를 완벽하게 분리할 수 있는 경우 데이터가 선형적으로 분리 가능하다고 합니다.

Weight Space





$w_0 \geq 0$

$w_0 + w_1 < 0$

- Weights (hypotheses) $\mathbf{w}$ are points
- Each training example $\mathbf{x}$ specifies a half-space $\mathbf{w}$ must lie in to be correctly classified: $\mathbf{w}^T\mathbf{x} \geq 0$ if $t = 1$.
- For NOT example:
  - $x_0 = 1, x_1 = 0, t = 1 \implies (w_0, w_1) \in \{\mathbf{w} : w_0 \geq 0\}$
  - $x_0 = 1, x_1 = 1, t = 0 \implies (w_0, w_1) \in \{\mathbf{w} : w_0 + w_1 < 0\}$
- The region satisfying all the constraints is the feasible region; if this region is nonempty, the problem is feasible, otw it is infeasible.

Towards Logistic Regression

Loss Functions

- 대신: 손실 함수를 정의한 다음 결과 비용 함수를 최소화하려고 시도합니다.
) 기억하세요: 비용은 훈련 세트에 대한 평균(또는 합산) 손실입니다.
- 겉보기에 명백한 손실 함수: 0-1 손실

$$L_{0-1}(y, t) = \begin{cases} 0 & \text{if } y = t \\ 1 & \text{if } y \neq t \end{cases}$$

$$= \mathbb{I}[y \neq t]$$

Attempt 1: 0-1 loss

- Usually, the cost $\mathcal{J}$ is the averaged loss over training examples; for 0-1 loss, this is the misclassification rate:

$$\mathcal{J} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[y^{(i)} \neq t^{(i)}]$$

Attempt 1: 0-1 loss

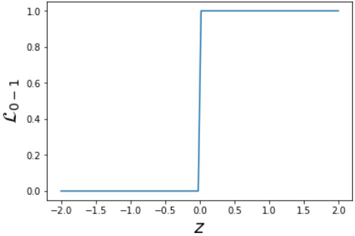
- 문제: 최적화하는 방법? 일반적으로 어려운 문제(NP-hard일 수 있음) 이는 계단 함수(0-1 손실)가 좋지 않기
-
  때문입니다(연속/부드러움/볼록 등)

Attempt 1: 0-1 loss

- 함수의 최소값은 임계점에 있을 것입니다. 0-1
- 손실의 임계점을 찾아보도록 합시다.
- 체인 규칙:

$$\frac{\partial \mathcal{L}_{0-1}}{\partial w_j} = \frac{\partial \mathcal{L}_{0-1}}{\partial z} \frac{\partial z}{\partial w_j}$$

- 하지만 $\partial \mathcal{L}_{0-1}/\partial z$는 정의된 모든 곳에서 0입니다!



) $\partial \mathcal{L}_{0-1}/\partial w_j = 0$은 가중치를 아주 작은 양으로 변경해도 손실에 아무런 영향이 없다는 것을 의미합니다.

) 거의 모든 지점의 기울기가 0입니다!

Attempt 2: Linear Regression

때때로 우리는 관심 있는 손실 함수를 최적화하기 쉬운 함수로 대체할 수 있습니다. 이를 부드러운 대리 손실 함수를 사용한 완화라고 합니다.

L0—1의 한 가지 문제: 최종 예측에 따라 정의되며, 이는 본질적으로 불연속성을 포함합니다.

대신 손실을 다음과 같이 정의하십시오. of $\mathbf{w}^T\mathbf{x}$ directly

- Redo notation for convenience: $z = \mathbf{w}^T\mathbf{x}$

Attempt 2: Linear Regression

- 우리는 이미 선형 회귀 모델을 맞추는 방법을 알고 있습니다. 대신 이것을 사용할 수 있을까요?

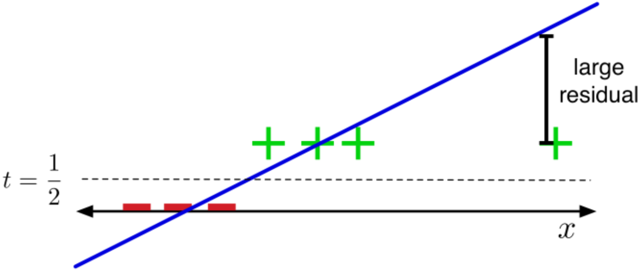
$$z = \mathbf{w}^T\mathbf{x}$$

$$\mathcal{L}_{SE}(z, t) = \frac{1}{2}(z - t)^2$$

타겟이 실제로 이진이라는 것은 중요하지 않습니다. 연속 값으로 취급하세요.
이 손실 함수의 경우 z를 1
2로 임계값을 설정하여 최종 예측을 하는 것이 합리적입니다(왜?)

Attempt 2: Linear Regression

The problem:



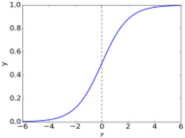
- 손실 함수는 높은 신뢰도로 정확한 예측을 할 때 싫어합니다! t = 1이면 z = 0보다 z = 10에 대해 더 불만스러워합니다.
-

Attempt 3: Logistic Activation Function

- [0, 1] 밖의 값을 예측할 이유는 분명히 없습니다. y를 이 구간에 압축해 보겠습니다.

- 로지스틱 함수는 일종의 시그모이드 또는 S자 모양 함수입니다.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



- *$\sigma^{-1}(y) = \log(y/(1-y))$를 로짓이라고 합니다.로지스틱 비선형성을 가진*
- *선형 모델은 로그선형이라고 합니다.*

$$z = \mathbf{w}^T \mathbf{x}$$

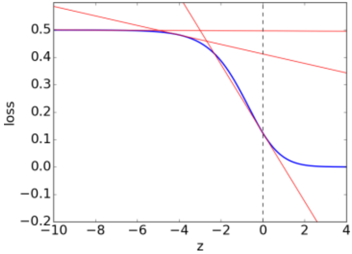
$$y = \sigma(z)$$

$$\mathcal{L}_{SE}(y, t) = \frac{1}{2}(y - t)^2.$$

- 이런 식으로 사용된 σ를 활성화 함수라고 한다.

Attempt 3: Logistic Activation Function

The problem:
(plot of L_{SE} as a function of z, assuming $t = 1$)



$$\frac{\partial L}{\partial w_j} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial w_j}$$

- z 0의 경우 σ(z) ≈ 0입니다.
- ∂z ≈ 0 (확인!) =⇒ ∂wj ≈ 0 =⇒ wj에 대한 미분이 작음
  =⇒ wj는 임계점과 같음
- 예측이 정말 틀렸다면 임계점(후보 솔루션)에서 멀리 떨어져 있어야 합니다.

Logistic Regression

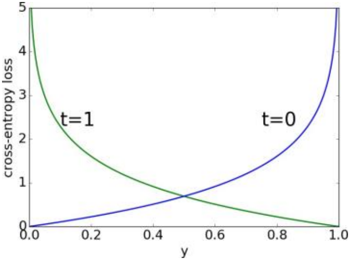
y ∈ [0, 1]이기 때문에 t = 1일 추정 확률로 해석할 수 있습니다. t = 0이면 y ≈ 1에 큰 페널티를 주고 싶습니다.

클린턴이 이길 것이라고 99% 확신했던 전문가들은 90% 확신했던 전문가들보다 훨씬 더 틀렸습니다.

크로스 엔트로피 손실(일명 로그 손실)은 이러한 직관을 포착합니다.

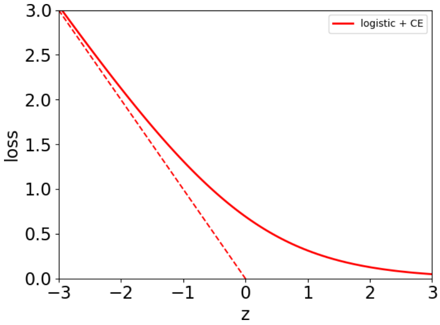
$$
\mathcal{L}_{\mathrm{CE}}(y, t) = \begin{cases} -\log y & \text{if } t = 1 \\ -\log(1 - y) & \text{if } t = 0 \end{cases}
$$

$$
= -t \log y - (1 - t) \log(1 - y)
$$



Logistic Regression:

$$z = \mathbf{w}^T \mathbf{x}$$
$$y = \sigma(z)$$
$$= \frac{1}{1 + e^{-z}}$$
$$\mathcal{L}_{CE} = -t \log y - (1 - t) \log(1 - y)$$



Plot is for target $t = 1$.

Gradient of Logistic Loss

Back to logistic regression:

$$\mathcal{L}_{\mathrm{CE}}(y, t) = -t \log(y) - (1 - t) \log(1 - y)$$

$$y = 1/(1 + e^{-z}) \quad \text{and} \quad z = \mathbf{w}^T \mathbf{x}$$

Therefore

$$\frac{\partial \mathcal{L}_{\mathrm{CE}}}{\partial w_j} = \frac{\partial \mathcal{L}_{\mathrm{CE}}}{\partial y} \cdot \frac{\partial y}{\partial z} \cdot \frac{\partial z}{\partial w_j} = \left(-\frac{t}{y} + \frac{1 - t}{1 - y} \right) \cdot y(1 - y) \cdot x_j$$

$$= (y - t) x_j$$

(verify this)

Gradient descent (coordinatewise) update to find the weights of logistic regression:

$$w_j \leftarrow w_j - \alpha \frac{\partial \mathcal{J}}{\partial w_j}$$

$$= w_j - \frac{\alpha}{N} \sum_{i=1}^{N} (y^{(i)} - t^{(i)}) \, x_j^{(i)}$$

Gradient Descent for Logistic Regression

경사 하강 업데이트 비교:

- **선형 회귀:**

$$\mathbf{w} \rightarrow \mathbf{w} - \frac{\alpha}{N} \sum_{i=1}^{N} (y^{(i)} - t^{(i)})\, \mathbf{x}^{(i)}$$

- 로지스틱 회귀:

$$\mathbf{w} \rightarrow \mathbf{w} - \frac{\alpha}{N} \sum_{i=1}^{N} (y^{(i)} - t^{(i)})\, \mathbf{x}^{(i)}$$

- 우연이 아닙니다! 이 둘은 모두 일반화 선형 모델의 예입니다. 하지만 더 자세히 설명하지는 않겠습니다.
- 평균 손실로 인한 합계 앞의 1번을 주목하세요. 이것이 비용이 손실의 합일 때(αr = α/N) 더 작은 학습률이 필요한 이유입니다.