배치 경사하강법 (BGD)

- Input : 훈련집합 $\underline{X}, \underline{Y}$, 학습률 ρ
- Output : 최적해 $\underline{\theta}^*$

1. 난수를 생성해서 초기해 $\underline{\theta}$를 결정.
2. repeat
3. $\underline{X}$ 내의 샘플에 대해 그래디언트 계산 : $\underline{\nabla}_1, \underline{\nabla}_2, \cdots, \underline{\nabla}_n$ (모든 sample)
   - $\{\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_n\}$
   - Del, nabla. ⇒ 그래디언트 계산 기호.
4. $\underline{\nabla}_{total} = \frac{1}{n}\sum_{i=1}^{n} \underline{\nabla}_i$ ← 평균값
5. $\underline{\theta} \leftarrow \underline{\theta} - \rho\underline{\nabla}_{total}$ ← 그래디언트 방향으로 update
   - θ의 변화량
6. $\underline{\theta}^* = \underline{\theta}$

(종료조건)
$\|\rho\underline{\nabla}_{total}\| < \varepsilon$ ⇒ 수치적으로 최적해는 아닐 수 있지만 수렴에 문제가 없을 정도로 계산

스토캐스틱 경사하강법 (SGD)

1. 난수를 생성해서 초기해 $\underline{\theta}$를 설정한다.
2. repeat
3. $\underline{X}$의 샘플 순서를 섞음.
4. for (i=1 to n)
5. i번째 sample에 대한 그래디언트 $\underline{\nabla}_i$ 계산
6. $\underline{\theta} \leftarrow \underline{\theta} - \rho\underline{\nabla}_i$
7. (종료조건)
8. $\underline{\theta}^* \leftarrow \underline{\theta}$

- SGD에서는 한 번에 하나씩 그래디언트를 계산하고 즉시 매개변수를 업데이트.
- 전체 샘플에 대해 다 업데이트하는 것을 1epoch이라고 부름. → 학습의 전체 횟수
- 새로운 epoch마다 샘플의 순서를 섞어서 무작위성을 부여하므로, 스토캐스틱이라고 함.

⇒ Data의 특성에 따라 오염이 심해짐

로지스틱 회귀 (Logistic Regression)

⇒ $S = \underline{w}^T\underline{x}$ 확신값

- 선형분류에서는 신호(출력) S를 ±1로 만들어서 출력으로 사용.
  → Thresholding (출력의 크기에 제약을 가함)
  → ex) $h(\underline{x}) = sign(\underline{w}^T\underline{x})$
- 선형회귀에서는 신호 S 자체를 출력으로 사용함.
  → ex) $h(\underline{x}) = \underline{w}^T\underline{x}$
- 로지스틱 회귀에서는 신호 S를 0과 1 사이의 값으로 만들어서 출력.
  → 로지스틱 함수

$$\theta(S) = \frac{e^S}{1+e^S} \in [0, 1]$$

S가 $-\infty$로 가면 0에 수렴
S가 ∞로 가면 1에 수렴

→ $h(\underline{x}) = \theta(\underline{w}^T\underline{x})$ (S자 모양 함수 사용)

Compare

- 선형분류에서는 이진분류를 통해 ±1 중 하나를 출력함.
- 로지스틱 회귀에서는 이진분류문제에서 결과를 확률로 출력함.
  - ↳ 확률로 표현.
  - ⇒ 이진 분류 문제를 로지스틱 회귀로 (확률을 이용해서 분류할 수 있음)
- 선형분류에서 사용하는 sign() 함수는 hard threshold라고 부름.
- 로지스틱 회귀에서 사용하는 logistic function은 soft threshold or Sigmoid function이라고 부름 ⇒ 다른 sigmoid 함수를 사용해도 문제가 없음.
  - S자 형태

Logistic Regression

- 학습하고자 하는 목표함수를 $f(\underline{x})$라 하자.

$$f(\underline{x}) = P[y=+1 \mid \underline{x}]$$

← 이진분류문제
(학습 모델의 함수) (실제 확률) (or -1)
x가 +1이 될 확률
or -1이 될 확률 $P[y=-1 \mid \underline{x}]$
⇒ 수학적인 함수를 사용해서 확률을 모델링

※ 의미 : 특징벡터 $\underline{x}$가 주어진 상황에서 출력값(label) y가 +1일 확률

→ $f(\underline{x})$ 목표함수는 실제로는 알 수 없고, 훈련데이터셋을 바탕으로 추정
↳ 데이터셋을 가지고 최적해를 찾아야 함 (완벽히 재현과 같을 수 없음)

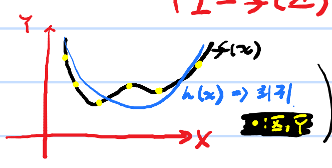
→ 추정해 낸 함수를 $h(\underline{x})$라고 쓰고, 가설(hypothesis)이라 부름.

※ 훈련데이터셋 $\underline{X}, \underline{Y}$는 어떤 확률분포 $P(y \mid \underline{x})$로부터 얻어진 것이라 생각
↳ 측정, 얻어지는 과정에서 확률분포에서 얻어진 것이라고 생각 (경향성이 있다고 생각) → 확률 P

→ Input data (특징 벡터) $\underline{x}$가 주어졌을 때, 출력에서 y가 나올 확률 (상관관계) : 예측하고자 하는 모델을 표현

$$P(y \mid \underline{x}) = \begin{cases} f(\underline{x}) & \text{for } y=+1 \\ 1-f(\underline{x}) & \text{for } y=-1 \end{cases}$$

⇒ 입력과 출력의 관계



⇒ 실제 $f(x)$를 완벽히 맞추기는 어려움
⇒ 정확하게 하기 위해 함수를 늘려야 함 (parameter가 복잡)
↳ 모든 Data point를 맞출 수 없음
※ 만약! 수학적으로 근사해서 값을 맞출 수 있음.

오차 정의하기

- $h(x)$가 목표함수 $f(\underline{x})$를 잘 모사하는지 여부를 판단하기 위해, 오차를 측정할 수 있어야 함.
  - → 벡터 중 하나.

ex) $\underline{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \underline{x}_T = \begin{bmatrix} 1.1 \\ 0.1 \end{bmatrix}$ ⇒ $\|\underline{x}_T - \underline{x}\|_2$

- 로지스틱 회귀에서 오차를 측정하는 방법 중 하나는 우도를 사용하는 것.
  - (확률적 개념)
  - ⇒ $f(x)$를 모름. $h(x)$는 알지만. ⇒ 간접적으로 찾아내야 함.

우도 (가능도, Likelihood)

- 가설 $h(\underline{x})$가 실제확률분포 $P(y \mid \underline{x})$를 표현한다고 할 때, 훈련집합의 입력데이터 $\underline{x}$를 넣었을 때 출력 데이터 y를 (실제값) 얻을 가능성 ⇒ 확률 X, 통계값
  - → $f(x)$를 $h(x)$로 사용하려고 할 때

ex) 주사위 값. 1, 1, 1이 주어졌을 때 확률모델을 예측해보자 (주사위 신뢰모름)
↳ 어떤 Dataset이 주어졌을 때 model을 추측.
⇒ model 추측 후 실제 모델과 얼마나 차이가 나는지 성능평가