기계학습

12강 순차데이터(3), 모델조합(1)

장필훈 교수

한국방송통신대학교 프라임칼리지

학습목차

한국방송통신대학교 프라임칼리지

01
순차데이터(3)

한국방송통신대학교 프라임칼리지

1-1 LDS(Linear Dynamical System)

- 선형동적 시스템
  - 선형예측이 바탕
  - 관측값과 실측값이 완전히 같지는 않다(노이즈)
- 선형예측의 예

1) 가장 최근 값들의 평균

2) 그냥 가장 최근값(노이즈가 작고 신호가 빠르게 변할때)

한국방송통신대학교 프라임칼리지

1-1 LDS(Linear Dynamical System)

- 연쇄의 길이가 길어질수록 복잡도가 증가하면 안된다.
  - 곱셈연산에 대해 닫혀있는 분포가 지수족밖에 없어서 분포는 가우시안을 쓴다.
- 잠재변수 z_n 이 마르코프 연쇄를 이룸.
- 이 추론문제를 푼 것: 칼만필터
  - hmm의 α와 같이 전진만 함

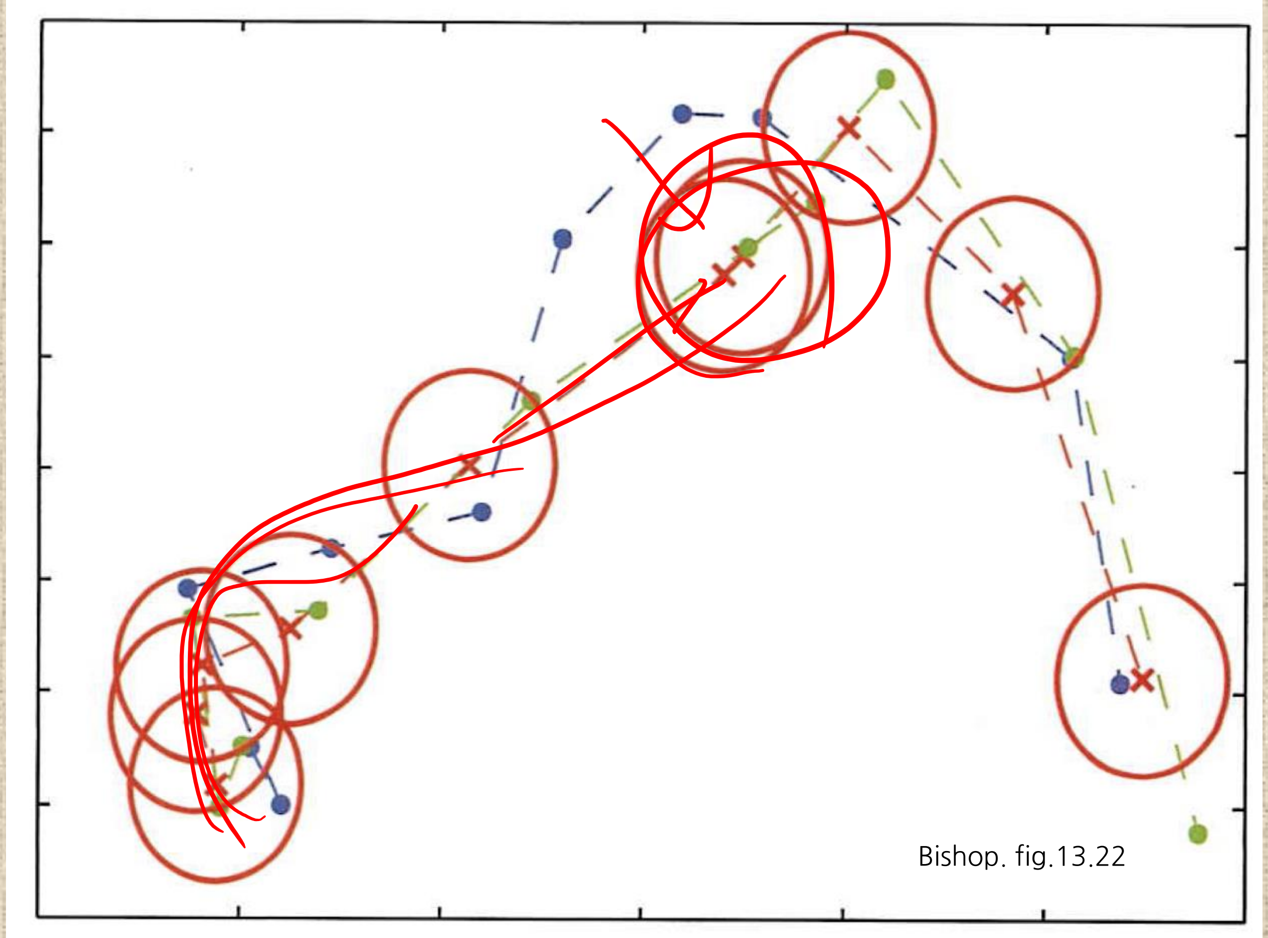
한국방송통신대학교 프라임칼리지

1-2 칼만필터

- 이전상태를 바탕으로 다음상태를 추측해내는 것.
- hmm과 같이 잠재변수와 분포를 가정한다.
- 시간적 변화에 따른 선형변환(=행렬곱으로 표현 가능)과 노이즈를 반영한다.
- 잠재변수, 노이즈 모두 가우시안을 가정.
    - 곱셈연산에 대해 닫혀있기 때문

한국방송통신대학교 프라임칼리지

1-2 칼만필터


Bishop. fig.13.22

2차원에서 물체의 실제 위치

실제위치에 노이즈가

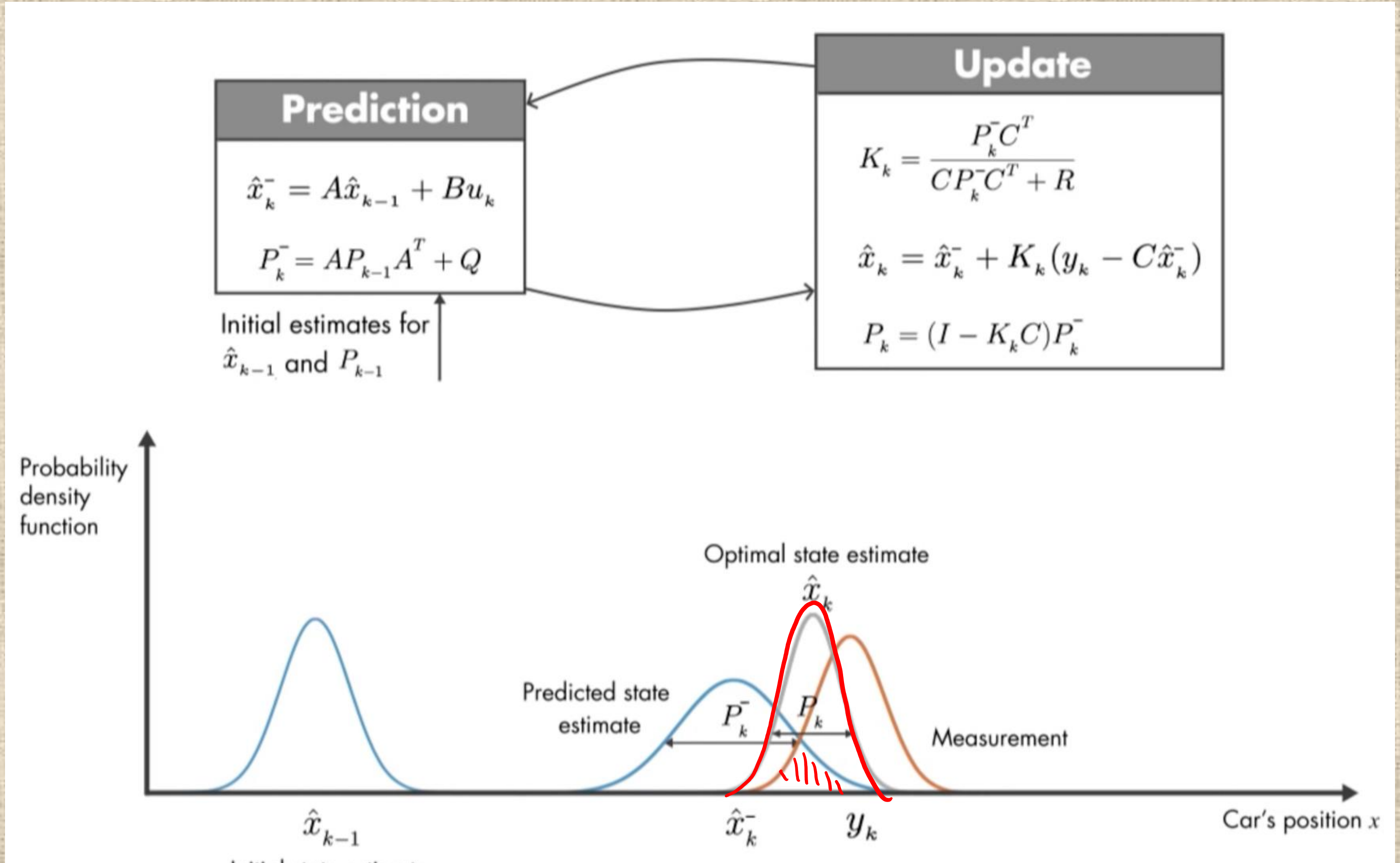
포함된 관측값

칼만필터의 예측값

(추론사후분포의 평균)

한국방송통신대학교 프라임칼리지

1-2 칼만필터



YouTube, Matlab, Understanding Kalman Filters, Part 4

02
모델조합

한국방송통신대학교 프라임칼리지

2-1 모델조합

- 베이지안 모델평균과 다르다.
  - 베이지안 모델평균은 전체 데이터집합이 단일 모델에 의해 생성된다.
  - 모델 조합은, 데이터집합의 각 포인트들이 서로 다른 잠재변수를 통해서 생성 가능하다.

예) $p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

한국방송통신대학교 프라임칼리지

2-2 bagging

- bootstrap aggregating
- 부트스트랩으로 일정수의 샘플집합들을 얻고
  각 샘플집합들에 대해 학습
- 학습된 결과를 합한다. 가장 단순하게는 voting
- 트리와 같이 불안정성이 큰 분류기에 효과가 크다
- variance를 줄이는 관점(bias도 줄어든다)

한국방송통신대학교 프라임칼리지

2-2 bagging

- 예측하고자 하는 함수 $h(\mathbf{x})$라 하면, 출력은

$$y_m(\mathbf{x}) = h(\mathbf{x}) + \epsilon_m(\mathbf{x})$$

- 제곱합 오류는

$$\mathbb{E}_{\mathrm{x}}[\{y_m(\mathbf{x}) - h(\mathbf{x})\}^2] = \mathbb{E}_{\mathrm{x}}[\epsilon_m(\mathbf{x})^2]$$

- 개별적으로 M개의 모델을 시험했을 때 평균오류는,

$$E_M = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{\mathrm{x}}[\epsilon_m(\mathbf{x})^2]$$

한국방송통신대학교 프라임칼리지

2-2 bagging

- (독립적 모델과 비교해서) bagging의 오류는

$$E = \mathbb{E}_x\left[\left\{\frac{1}{M}\sum_{m=1}^{M} y_m(x) - h(x)\right\}^2\right]$$

$$= \mathbb{E}_x\left[\left\{\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(x)\right\}^2\right]$$

$\frac{1}{M^2}$

한국방송통신대학교 프라임칼리지

2-2 bagging

- 각각의 오류가 0평균을 가지고 서로 독립이면,

$$\mathbb{E}_{\mathrm{x}}[\epsilon_m(\mathbf{x})] = 0$$

$$\mathbb{E}_{\mathrm{x}}[\epsilon_m(\mathbf{x})\epsilon_l(\mathbf{x})] = 0, m \neq l$$

이므로

$$E_{\text{bagging}} = \frac{1}{M} E_{\mathrm{M}}$$

한국방송통신대학교 프라임칼리지

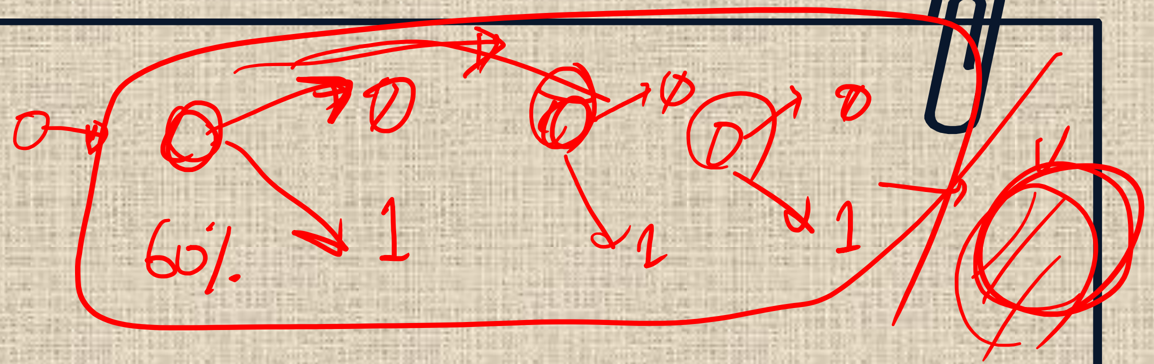
2-2 bagging

- 각각의 모델을 썼을 때의 평균에러보다 $\frac{1}{M}$에 불과하다!
  - 실제로 이런일은 잘 일어나지 않음
    - 오류는 보통 서로 높은 상관관계를 가진다
  - 하지만 줄기는 줄어든다.

$$E_{\text{bagging}} \leq E_{\text{M}}$$

한국방송통신대학교 프라임칼리지

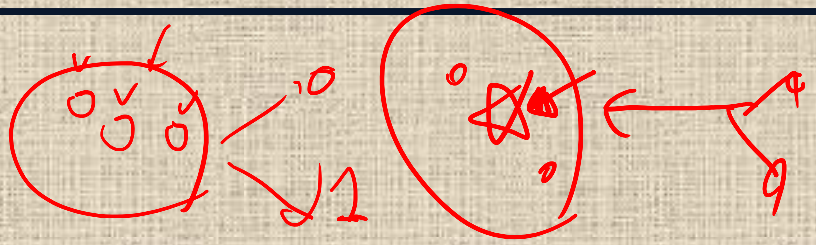
2-3 Adaboost



- adaptive boosting
- weak learner 여러개를 모아서 좋은 분류기를 만들 수 있다.
  - base classifiers
  - random보다 조금만 더 좋으면 된다.
- 회귀에도 응용 가능 v
- bias를 줄이는 관점(variance도 줄어든다.)

한국방송통신대학교 프라임칼리지

2-3 Adaboost



- base classifier들은 순차적으로 학습한다
- base classifier들은 weighted dataset으로 학습한다.
  - weighting coefficient는 앞단계의 base classifier에 의해 결정된다.
    - 앞단계에서 오분류된것들이 더 큰 가중치를 가진다.
- 학습이 끝나면 weighted sum

한국방송통신대학교 프라임칼리지

2-3 Adaboost

- binary classification 예제 : $t_n \in \{-1,1\}$, 각 데이터 포인트는 w_n을 부여받음.

1. 가중치를 초기화한다. $w_n = \frac{1}{N}$
2. 다음 오류함수를 최소화하도록 학습.

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)$$

I: Indicator. 식이 참이어야 1

한국방송통신대학교 프라임칼리지

2-3 Adaboost

3. 다음계산 반복

$$\alpha_m = \ln\frac{1-\epsilon_m}{\epsilon_m}, \qquad \epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$

$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)\}$$

4. 학습이 끝나면,

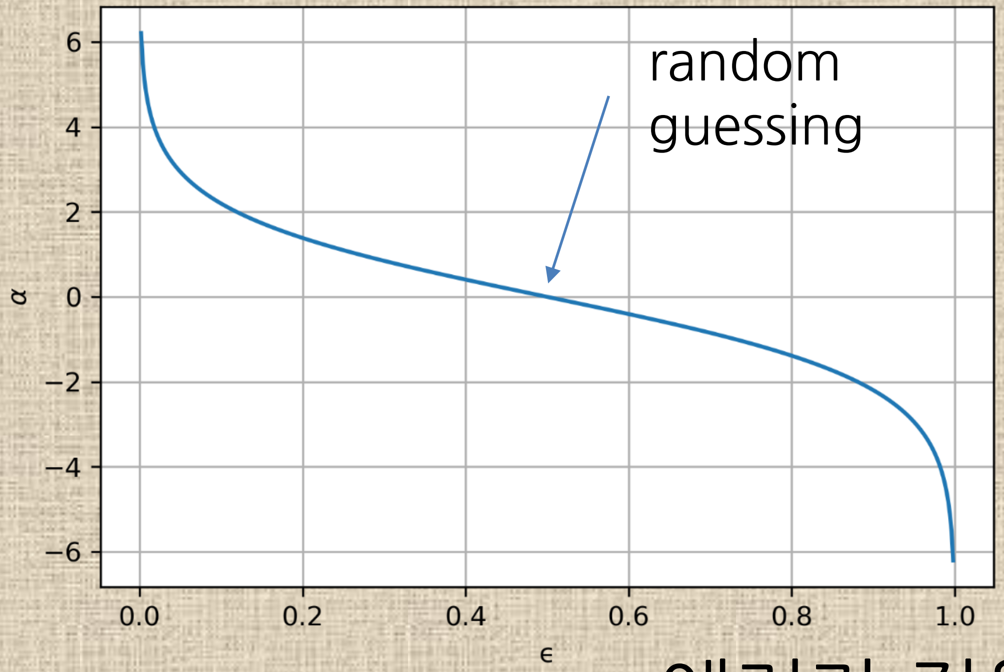
$$Y_M(\mathbf{x}) = \mathrm{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x})\right)$$

한국방송통신대학교 프라임칼리지

2-3 Adaboost

- prediction weights

$$\alpha_m = \ln \frac{1-\epsilon_m}{\epsilon_m},$$

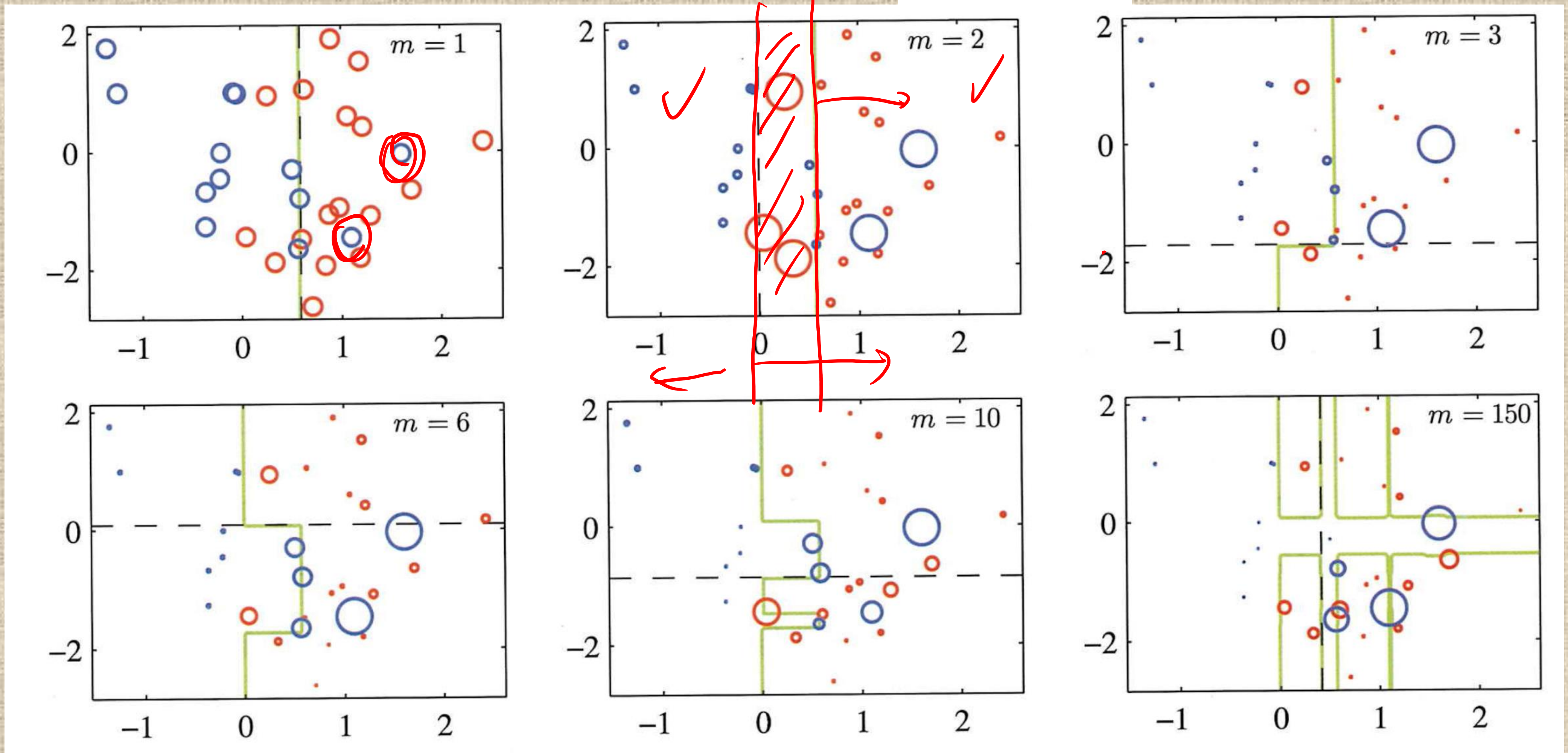


에러가 적을수록 좋음

한국방송통신대학교 프라임칼리지

2-3 Adaboost

$\sum \alpha_m y_m,\ y_m$



Bishop. Fig.14.2

한국방송통신대학교 프라임칼리지

2-3 Adaboost

- ‘지수오류함수의 연속적인 최소화’로 해석가능
- 다음 오류함수를 가정.

$$E = \sum_{n=1}^{N} \exp\{-t_n f_m(\mathbf{x_n})\}$$

$$f_m(\mathbf{x}) = \frac{1}{2}\sum_{l=1}^{m} \alpha_l y_l(\mathbf{x}), \qquad t_n \in \{-1, 1\}$$

한국방송통신대학교 프라임칼리지

2-3 Adaboost

- α_m과 $y_m(\mathbf{x})$에 대해서만 최소화하기 위해 E를 변형
  $(\alpha_1, \ldots, \alpha_{m-1}, y_1(\mathbf{x}), \ldots, y_{m-1}(\mathbf{x})$에 대해서는 고정)

$$E = \sum_{n=1}^{N} \exp\left\{-t_n f_{m-1}(\mathbf{x}_n) - \frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\}$$

$$= \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\}$$

2-3 Adaboost

- 옳게 분류한 데이터포인트를 T_m, 오분류된 포인트를 M_m

$$E = e^{-\frac{\alpha_m}{2}} \sum_{n \in T_m} w_n^{(m)} + e^{\frac{\alpha_m}{2}} \sum_{n \in M_m} w_n^{(m)}$$

$$= \left(e^{\frac{\alpha_m}{2}} - e^{-\frac{\alpha_m}{2}}\right) \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n) + e^{-\frac{\alpha_m}{2}} \sum_{n=1}^{N} w_n^{(m)}$$

한국방송통신대학교 프라임칼리지

2-3 Adaboost

- 앞의 식을 보면, 두번째 항은 $y_{m(\mathrm{x})}$와 무관하다.

  즉, 앞의 식을 $y_m(\mathbf{x})$ 에 대해 최소화 하는것은

  p18의 식을 최소화 하는것과같다.

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)$$

앞의 계수$\left(e^{\frac{\alpha_m}{2}} - e^{-\frac{\alpha_m}{2}}\right)$ 도 상수취급 가능하기 때문.

한국방송통신대학교 프라임칼리지

2-3 Adaboost

- $\frac{\partial E_m}{\partial \alpha_m} = 0$도 구해보면, 앞의 결과($\alpha_m = \ln \frac{1-\epsilon_m}{\epsilon_m}$)를 얻는다.
- α_m과 $y_m(\mathbf{x})$를 얻은 다음, $E_{(m+1)}$을 구해보면, p19의 재귀식을 유도해낼 수 있다.

한국방송통신대학교 프라임칼리지

2-3 Adaboost

$$E_m = \sum_{n=1}^{N} \exp\{-t_n f_{m+1}(\mathbf{x}_n)\}$$

$$= \sum_{n=1}^{N} \exp\left\{-t_n f_{m-1}(\mathbf{x}_n) - \frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n) - \frac{1}{2} t_n \alpha_{m+1} y_{m+1}(\mathbf{x}_n)\right\}$$

$$= \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\} \exp\left\{-\frac{1}{2} t_n \alpha_{m+1} y_{m+1}(\mathbf{x}_n)\right\}$$

$$= \sum_{n=1}^{N} w_n^{(m+1)} \exp\left\{-\frac{1}{2} t_n \alpha_{m+1} y_{m+1}(\mathbf{x}_n)\right\}$$

한국방송통신대학교 프라임칼리지

2-3 Adaboost

- 다음을 앞쪽 식에 대입하면, p19의 재귀식을 얻는다.

$$t_n y_m(\mathbf{x}_n) = 1 - 2I(y_m(\mathbf{x}_n) \neq t_n)$$

$$w_n^{(m+1)} = w_n^{(m)} \exp\left\{-\frac{\alpha_m}{2}\right\} \exp\{\alpha_m I[y_m(\mathbf{x}_n) \neq t_n]\}$$

- $\exp\left\{-\frac{\alpha_m}{2}\right\}$은 ϵ_m을 계산할 때 분모분자에서 cancel out.
  - 따라서 최솟값의 위치에 영향을 주지 않음.

한국방송통신대학교 프라임칼리지

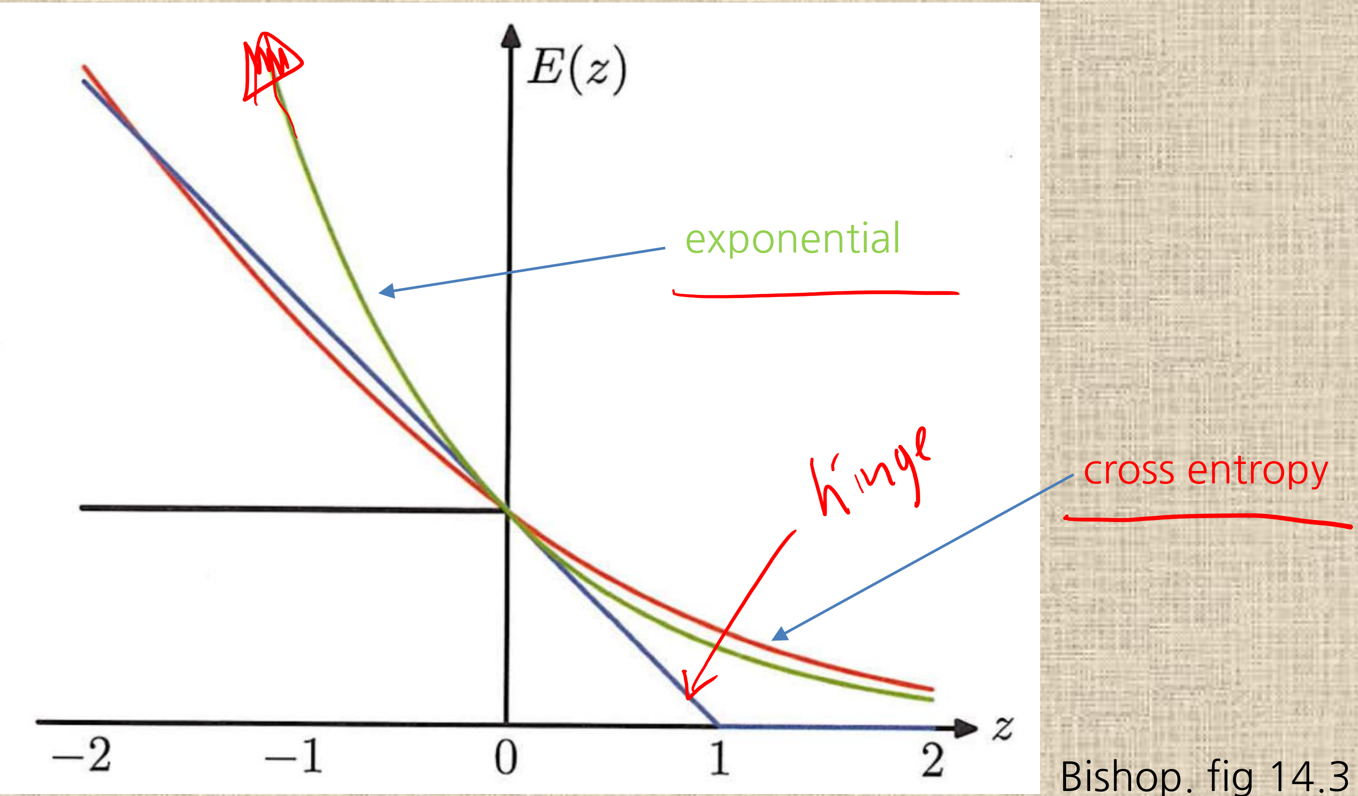
2-3 Adaboost

- 구현이 매우 쉽다
- $t_n y_m(\mathbf{x})$에서 보듯이 outlier에 민감하다.
  - 게다가 exponential하게 증가한다.
- exponential error값은 확률로 해석하기가 곤란하다.
  (cf. log likelihood)
- multiclass 분류문제에 적용하기가 까다롭다.

한국방송통신대학교 프라임칼리지

2-3 Adaboost

- 오류함수들 비교

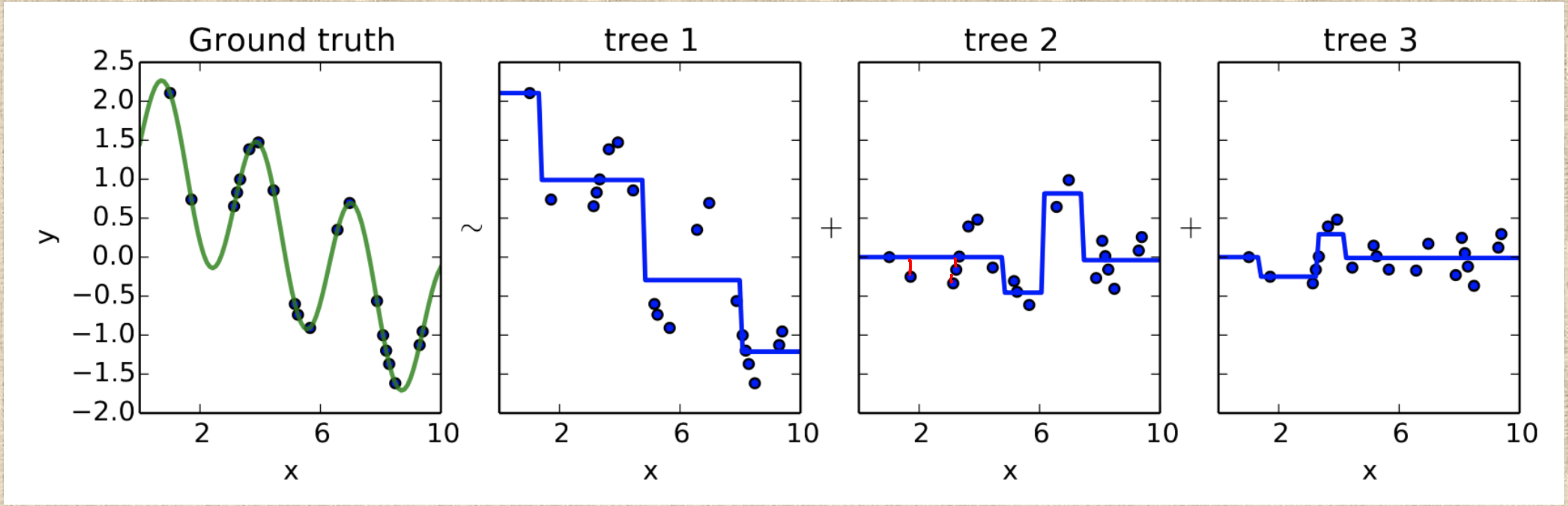


Bishop. fig 14.3

한국방송통신대학교 프라임칼리지

2-4 gradient boost

- regression, residual fitting



Peter Prettenhofer, Gradient Boosted Regression Trees.
https://github.com/pprett/pydata-gbrt-tutorial/blob/master/slides/slides.pdf

한국방송통신대학교 프라임칼리지

2-4 gradient boost

- adaboost는 stump가 연속되지만, GB는 tree가 연속됨.
  - leaf는 보통 8~32개
- tree형태의 분류기는 아무런 방법이나 써도 된다.
  - weak learner
  - Adaboost와 같음
  - regression문제에서는 leaf가 잔차(residual)

한국방송통신대학교 프라임칼리지

2-4 gradient boost

1. 초기화 $F_0 = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$

2. 모든 데이터에 대해

 1) 잔차계산 : $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$

 2) r_{im}에 대해 regression tree를 fit

 3) 오차계산 : $\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$

 4) 함수 갱신

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_m I(x \in R_{jm})$$

3. $F_M(x)$가 결과

한국방송통신대학교 프라임칼리지

2-4 gradient boost

- 예제(YouTube, StatQuest「Gradient Boost」)

height	favorite color	gender	weight
1.6	blue	male	88
1.6	green	female	76
1.5	blue	female	56

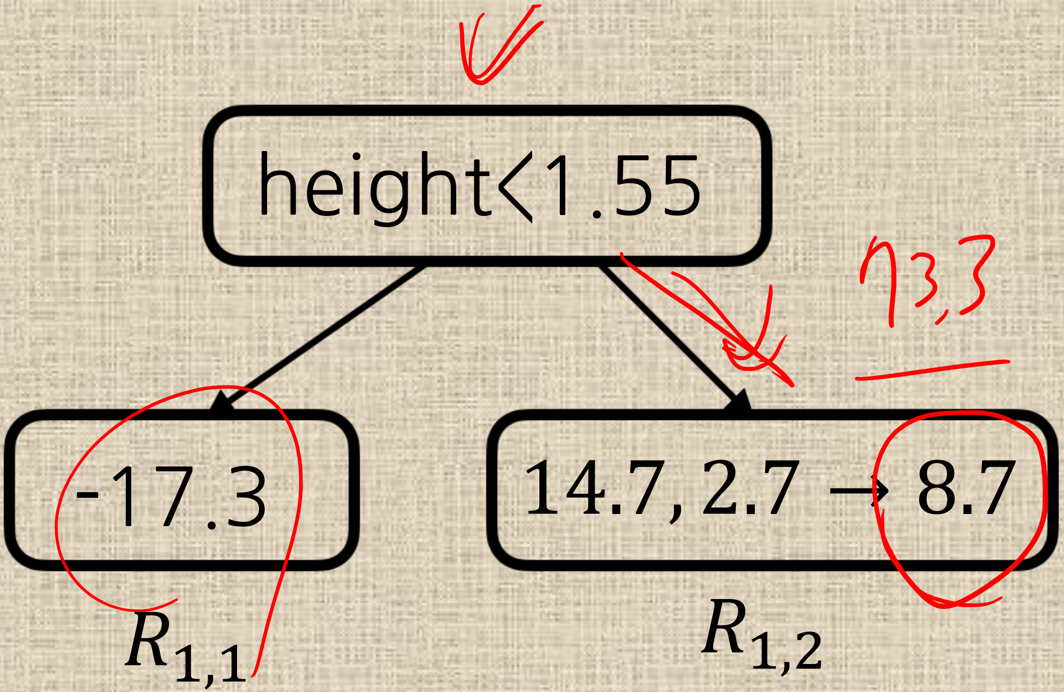
- 제곱오차를 loss로 가정. 그러면 $F_0(\mathbf{x}) = 73.3$

한국방송통신대학교 프라임칼리지

2-4 gradient boost

- step2-2,2-3

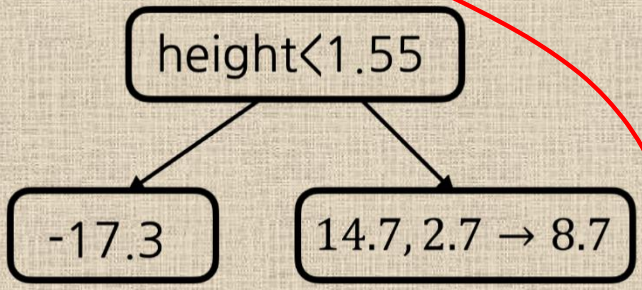
weight	$r_{i,1}$
88	88-73.3=14.7
76	2.7
56	-17.3



2-4 gradient boost

- step2-4

$$F_1(x) = F_0(x) + 0.1 \times$$



height	favorite color	gender	weight	prediction
1.6	blue	male	88	73.3+0.1×8.7=74.2
1.6	green	female	76	74.2
1.5	blue	female	56	71.6

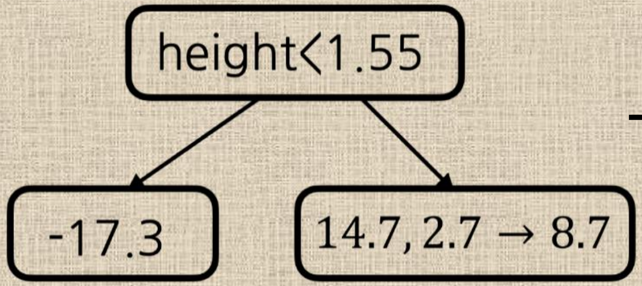
다음 iteration에서 잔차계산의 기준은 이 값

한국방송통신대학교 프라임칼리지

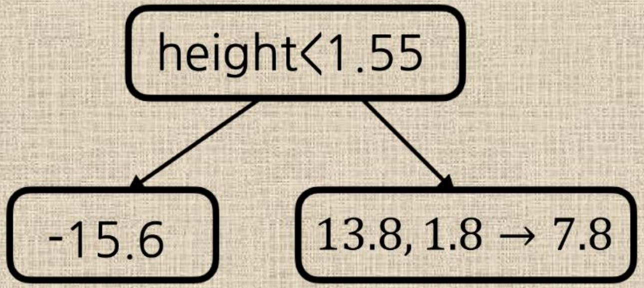
2-4 gradient boost

- $F_2(x)$

$$= F_0(x) + 0.1 \times$$



$$+0.1 \times$$



73.3

- 보통은 stump를 쓰지 않는다.(더 큰 tree를 쓴다)

2-4 gradient boost

- classification
  - regression과 동일한 과정을 거친다
  - 0~1을 출력으로 내고, 확률값으로 해석하게 한다.
  - loss

$$-\hat{y}\log\frac{p}{1-p}+\log\left(1+\exp\left(\log\frac{p}{1-p}\right)\right)$$

$\log\frac{p}{1-p}$=log odds

한국방송통신대학교 프라임칼리지

2-4 gradient boost

- classification(cont.)
  - loss미분이 어려워서 근사치를 쓴다.
  - F_M은 log odds의 합이고, 각 weak learner는 확률을 출력으로 주기 때문에 summation(Σ)하기 전에

$$\frac{\sum r}{\sum[p_i \times (1 - p_i)]}$$

r:residual, p:$F_{-1}(x)$

한국방송통신대학교 프라임칼리지

다음시간

13강

- 모델조합:tree

한국방송통신대학교 프라임칼리지