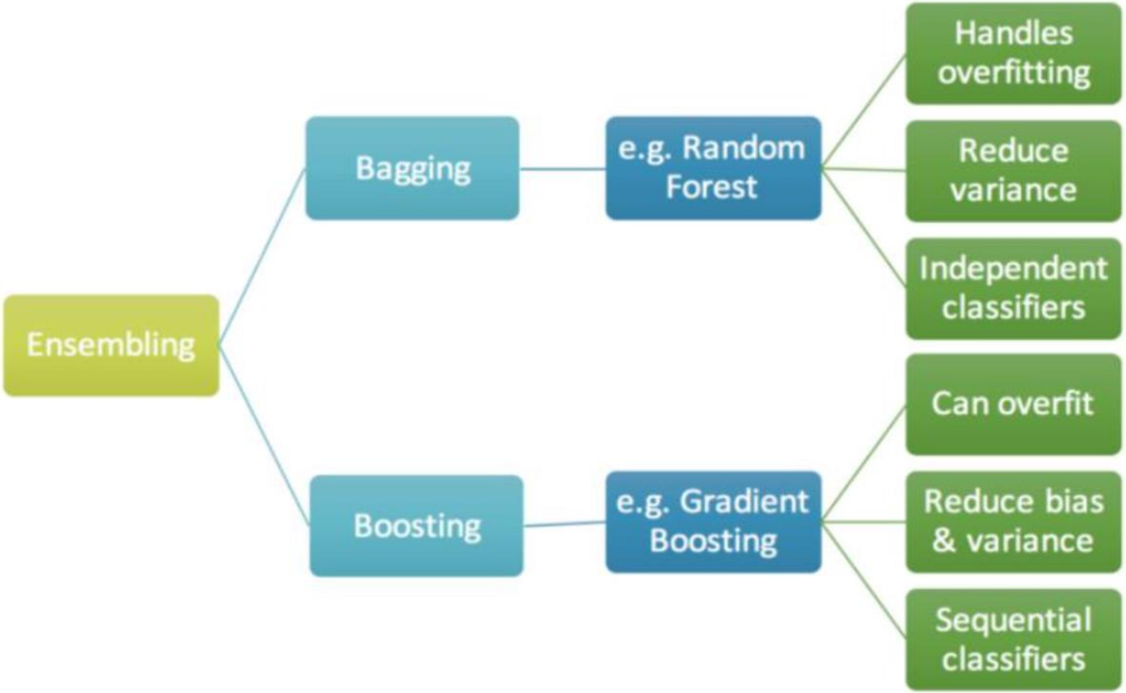
KUBIG
Data Science and Machine Learning

Week 6. Ensemble Learning

What is Ensemble Learning?

Ensemble Learning



- 병렬적 모델결합
- 독립적으로 모델 구성
- 매 sampling마다 동일 가중치 부여

- 직렬적 모델결합
- 이전 모델의 오류를 바탕으로 새 모델 구성
- 학습오류 큰 데이터에 가중치 부여
- 단일 모델의 성능 낮을 경우

Ensemble Learning – Voting Classifier

- Voting Classifier

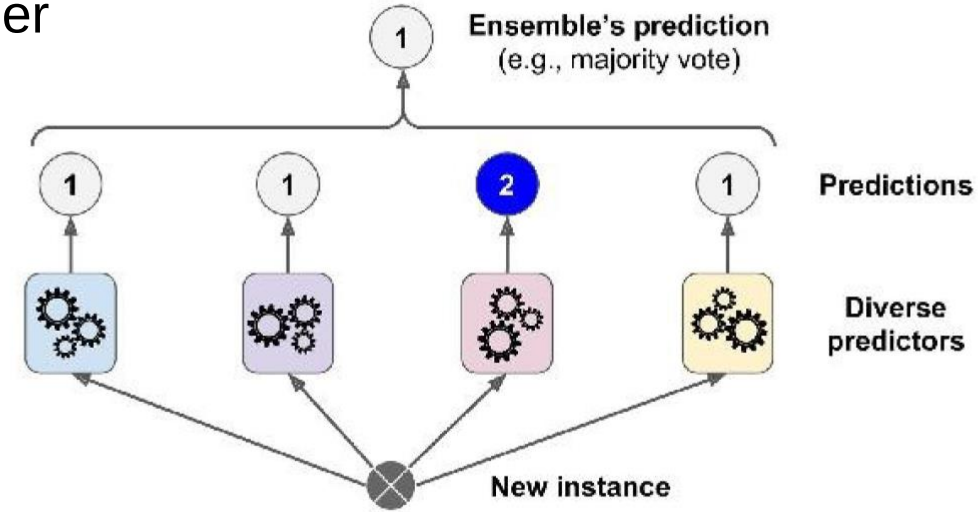


Figure 7-2. Hard voting classifier predictions

Ensemble Learning – Bagging

- Bagging (Bootstrap Aggregating)

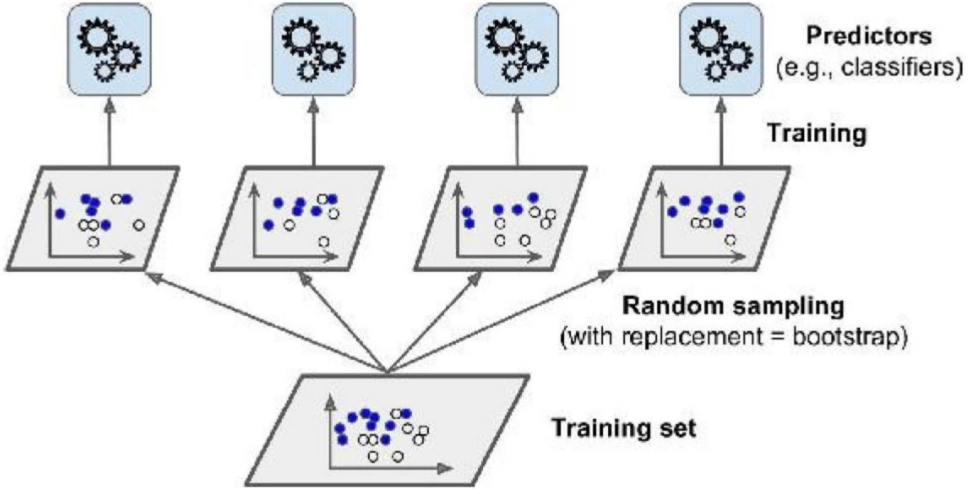


Figure 7-4. Pasting/bagging training set sampling and training

Note # 1. Bootstrap

Jackknife Estimator

- Let $\hat{\theta}_{[i]}$ denotes the “Leave-One-Out” estimator
- Jackknife pseudo-values are defined by

$$\hat{\theta}_{ps,i} = n\hat{\theta} - (n-1)\hat{\theta}_{[i]}$$

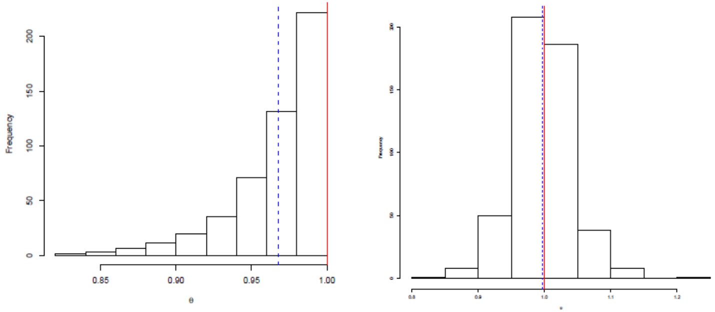
- Bias-adjusted Jackknife estimator is

$$\hat{\theta}_J = \frac{1}{n}\sum\hat{\theta}_{ps,i} = \hat{\theta} - (n-1)(\overline{\theta}_{[\mathrm{n}]} - \hat{\theta})$$

Note # 1. Bootstrap

Jackknife Estimator

- Illustration of the bias corrected version of the sample maximum $\hat{\theta}$ for $U_i \overset{iid}{\sim} (0, 1)$. (i.e. $\theta = 1$)



(a) $\hat{\theta} = U_{(n)}$ (b) Bias-Corrected $\hat{\theta}$, $\hat{\theta}_J$

Note # 1. Bootstrap

- Bootstrap is a general technique for estimating unknown quantities associated with sampling distribution of estimators such as

  - Standard Errors
  - Confidence Intervals
  - p-values

Note # 1. Bootstrap

- Suppose $F(x)$ is the true population distribution.

- We estimate the functional of F based on the sample $X_1, \ldots, X_\mathrm{n}$.
  Ex) Population expectation

$$\mu = E[X] = \int xf(x)\,dx \quad \left(= \int x\, dF(x)\right)$$

$$\hat{\mu} = \bar{X}_n = \frac{1}{n}\sum X_i \qquad \left(= \int x\, dF_n(x)\right)$$

Note # 1. Bootstrap

- $F_n(x)$ denotes the empirical distribution of $(X_1, \dots, X_{\mathrm{n}})$.

$$F_n(x) = \frac{1}{n} \sum_{i}^{n} I\,(x \leq X_i)$$

- Underlying fundamentals of this idea is

$$F_n(x) \quad \rightarrow \quad F(x)$$

Note # 1. Bootstrap

- Uncertainty / Randomness comes from

$$F(x) - F_n(x)$$

- Uncertainty quantification is not trivial since we only have a single $F_n(x)$ for unknown $F(x)$

Note # 1. Bootstrap

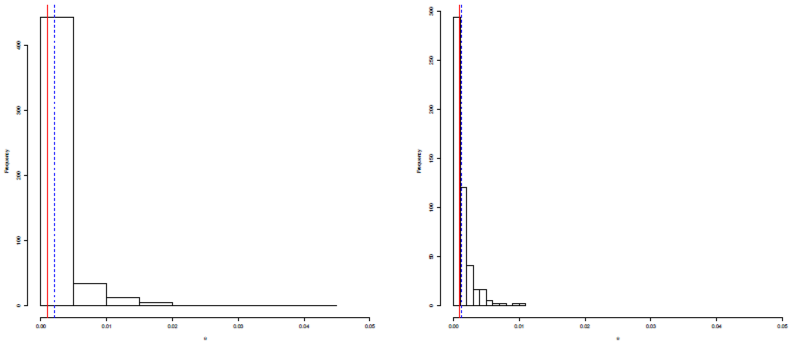
- Given a set of sample $(X_1, \dots, X_\mathrm{n})$, a bootstrap sample denoted by $(X_1^*, \dots, X_\mathrm{n}^*)$ is a random drawing samples **with replacement** from $(X_1, \dots, X_\mathrm{n})$.

- The idea of bootstrap is

$$F_n^*(x) \quad \rightarrow \quad F_n(x) \qquad \approx \qquad F_n(x) \quad \rightarrow \quad F(x)$$

Note # 1. Bootstrap

- Comparison of variance estimator for sample maximum $\hat{\theta}$ for $U_i \overset{iid}{\sim} (0, 1)$. (i.e. $\theta = 1$)



(a) Jacknife (b) Bootstrap

Figure: Histogram of variance estimator for 500 independent repetitions: Monte Carlo MSE is .00903 for the jackknife estimator and .00108 for the bootstrap estimator.

Ensemble Learning – Bagging

- Bagging (Bootstrap Aggregating)

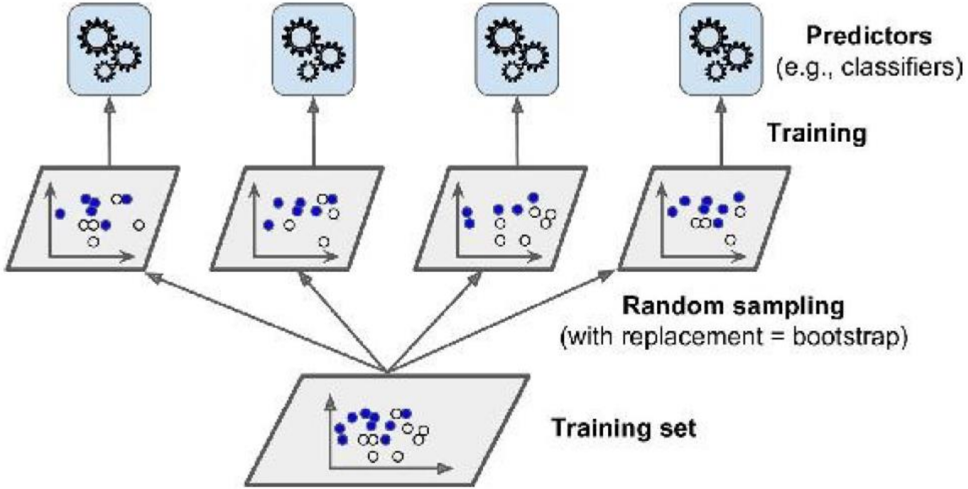
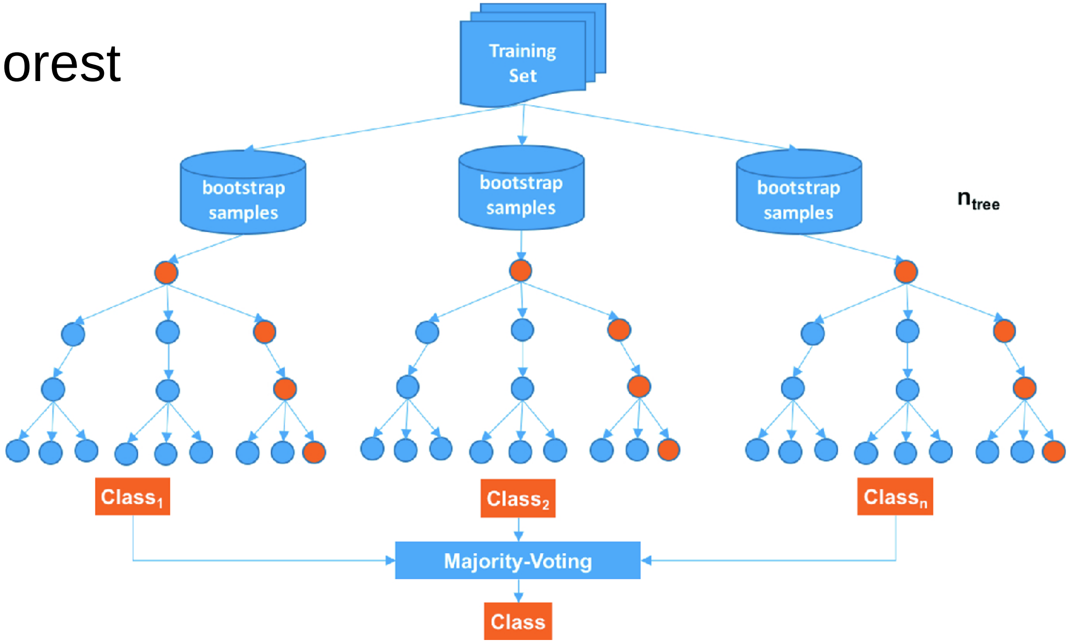


Figure 7-4. Pasting/bagging training set sampling and training

Ensemble Learning – Random Forest

- Random Forest



Ensemble Learning – Random Forest

- Random Forest

1. From $\mathbf{X}_{n\times p}$, obtain $\mathbf{X}^*_{n\times p}$ bootstrap samples.

2. For $\mathbf{X}^*_{n\times p}$, fit a decision tree by using randomly selected $k\ (\leq p\ )$ features.

   In general, $k = \sqrt{p}$.

3. Repeat 1-2 M times. (M = # of trees)

Note 2. Decision Trees

특성	로지스틱	KNN	LDA	SVM	의사결정 나무	최소제곱 선형모형	Neural network
자료 type 민감성	상	상	상	상	하	상	상
결측 자료 영향	상	중	상	상	하	상	상
이상치 민감성	상	하	상	상	하	상	상
표준화	선택	선택	선택	선택	불필요	불필요	필요
해석의 용이성	용이	난해	난해	난해	용이	용이	매우 난해
성능	중간	중간	중간	중간	중간	중간	높음

Ensemble Learning - Boosting

- Boosting

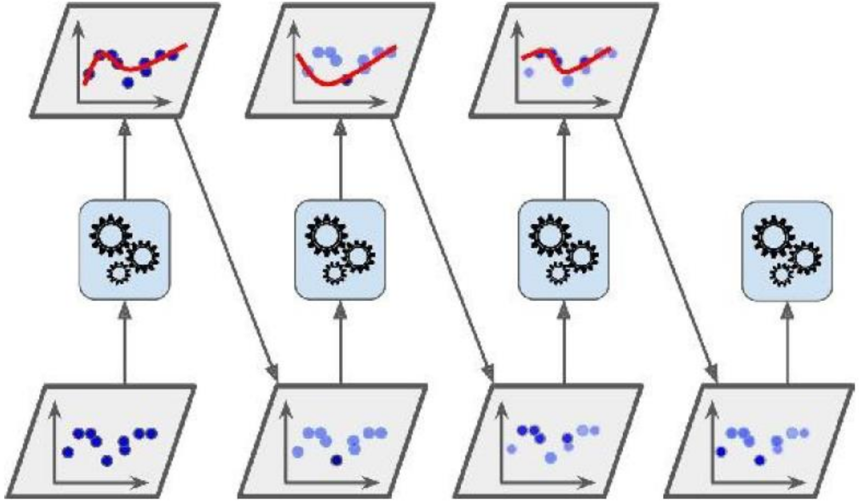


Figure 7-7. AdaBoost sequential training with instance weight updates

Ensemble Learning - Boosting

Algorithm 10.2 *Forward Stagewise Additive Modeling.*

1. Initialize $f_0(x) = 0$.
2. For $m = 1$ to M:
   (a) Compute
   $$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma)).$$
   (b) Set $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$.

Ensemble Learning - Boosting

Algorithm 10.2 *Forward Stagewise Additive Modeling.*

1. Initialize $f_0(x) = 0$.
2. For $m = 1$ to M:

   (a) Compute

   $$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta b(x_i;\gamma)).$$

   (b) Set $f_m(x) = f_{m-1}(x) + \beta_m b(x;\gamma_m)$.

Ensemble Learning - AdaBoost

1. $\mathcal{D}_1(\boldsymbol{x}) = 1/m$
2. $for\ t = 1,2,\dots,T\ do$
3. $\quad h_t = \mathfrak{L}(D, \mathcal{D}_t)$
4. $\quad e_t = P_{x\sim D_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$
5. $\quad if\ \ e_t > 0.5\ \ then\ break$
6. $\quad \alpha_t = \frac{1}{2}\ln(\frac{1-e_t}{e_t})$
7. $\quad \mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}_t(x)\exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}$
8. $end\ for$
9. $\mathcal{H}(\boldsymbol{x}) = sign(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}))$

Ensemble Learning - AdaBoost

- Change little bit…

(c) $as_m = \frac{1}{2}\log\left(\frac{1-err_m}{err_m}\right)$ → amount of say

(d) $w_i^{(m+1)} = \begin{cases} e^{-as_m} & \text{if } y_i = G_{m-1}(x_i) \\ e^{as_m} & \text{if } y_i \neq G_{m-1}(x_i) \end{cases}$ → $\sum w_i^{(m)} \neq 1$

Ensemble Learning - AdaBoost

관측치	1	2	3	4	5	6	7	8
x	5	10	15	20	25	30	35	40
y	-1	-1	1	1	1	-1	-1	1
가중치	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$

<표 11.1> adaboost를 위한 학습데이터

Ensemble Learning - AdaBoost

관측치	1	2	3	4	5	6	7	8
x	5	10	15	20	25	30	35	40
y	-1	-1	1	1	1	-1	-1	1
가중치	0.577	0.577	0.577	0.577	0.577	1.733	1.733	0.577
조정가중치	0.083	0.083	0.083	0.083	0.083	0.251	0.251	0.083

<표 11.2> 아다부스트를 위한 조정된 가중치의 계산

Ensemble Learning - AdaBoost

관측치	1	2	3	4	5	6	7	8
x	5	10	20	30	30	35	35	40
y	-1	-1	1	-1	-1	-1	-1	1
가중치	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$

<표 11.3> 두 번째 tree stump를 위한 데이터셋

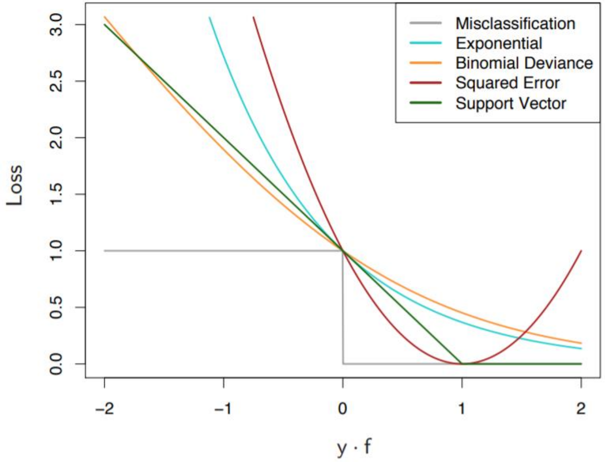
Ensemble Learning - AdaBoost

관측치	1	2	3	4	5	6	7	8
x	5	10	15	20	25	30	35	40
y	-1	-1	1	1	1	-1	-1	1
가중치	0.379	0.379	2.638	2.638	2.638	0.379	0.379	0.379
조정가중치	0.038	0.038	0.270	0.270	0.270	0.038	0.038	0.038

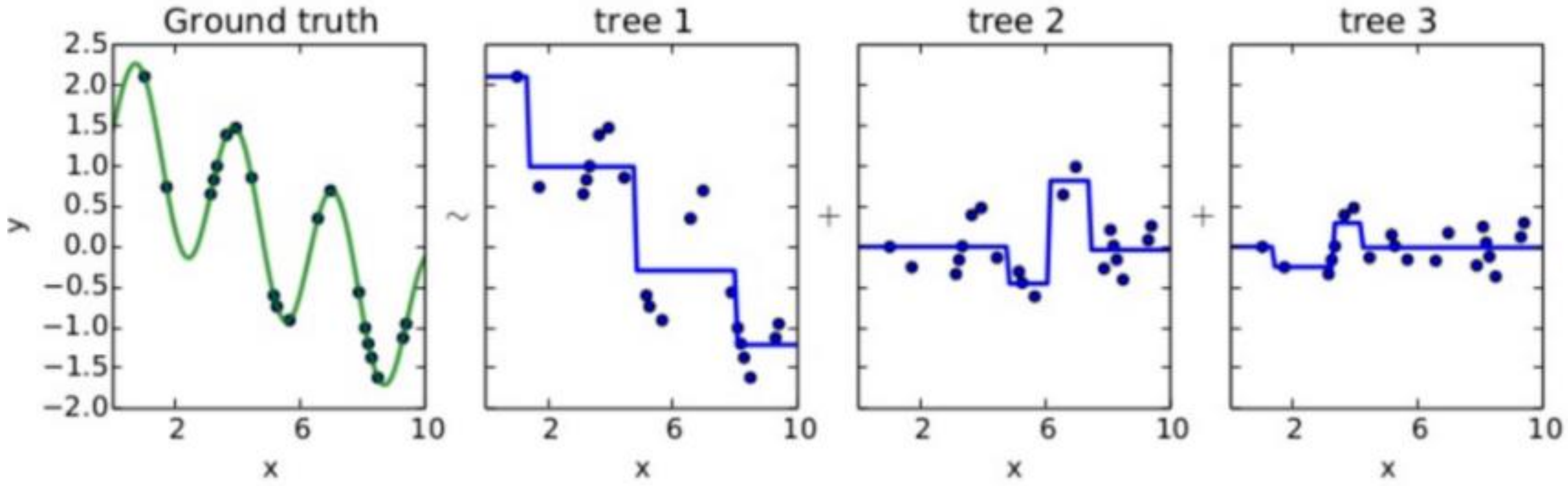
<표 11.2> 아다부스트를 위한 조정된 가중치의 계산

Ensemble Learning - AdaBoost

- AdaBoost is a special case of Forward Stagewise Additive Modeling (=Boosting) when we use Exponential Loss!



Ensemble Learning – Gradient Boosting



Ensemble Learning - Gradient Boosting

- Gradient Boosting은 임의의 differentiable loss function에 대해 Forward Stagewise Additive Model의 최적화 문제를 근사적으로 해결하는 알고리즘이다.

$$\sum_{i=1}^{n} L(y_i, f(\boldsymbol{x}_i))$$

$$f_m(\boldsymbol{x}_i) = f_{m-1}(\boldsymbol{x}_i) - \eta_m \frac{\partial L(y_i, f(\boldsymbol{x}_i))}{\partial f(\boldsymbol{x}_i)} \Big|_{f(\boldsymbol{x}_i = f_{m-1}(\boldsymbol{x}_i))}$$

$$= f_{m-1}(\underline{x_i}) - \eta_m g_{im}$$

Ensemble Learning – Gradient Boosting

- For regression..

Algorithm 10.3 *Gradient Tree Boosting Algorithm.*

---

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^N L(y_i, \gamma)$.
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \ldots, N$ compute
   $$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$
   (b) Fit a regression tree to the targets r_{im} giving terminal regions $R_{jm},\ j = 1, 2, \ldots, J_m$.
   (c) For $j = 1, 2, \ldots, J_m$ compute
   $$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$
   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.

Ensemble Learning – Gradient Boosting

- For classification..

Algorithm 10.4 *Gradient Boosting for K-class Classification.*

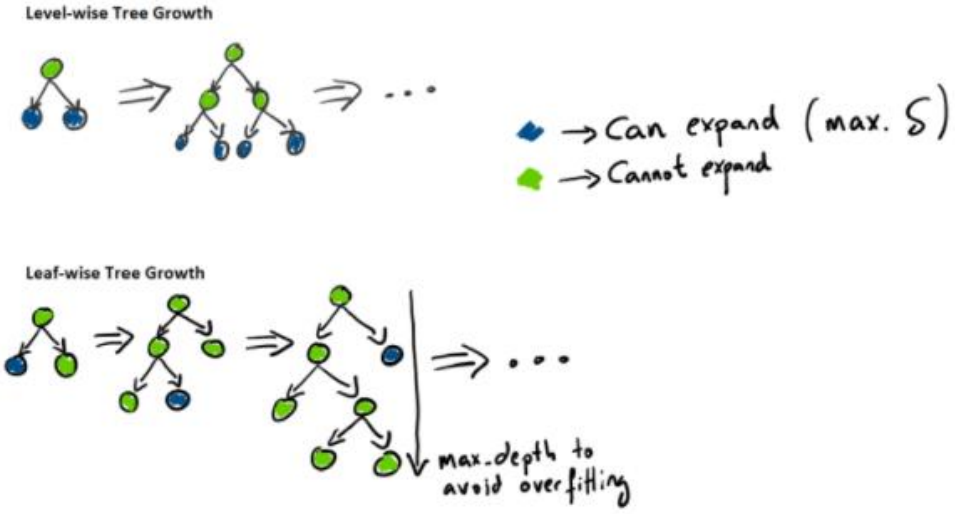
1. Initialize $f_{k0}(x) = 0,\ k = 1, 2, \dots, K$.
2. For m=1 to M:
   (a) Set
   $$p_k(x) = \frac{e^{f_k(x)}}{\sum_{\ell=1}^{K} e^{f_\ell(x)}},\ k = 1, 2, \dots, K.$$
   (b) For $k = 1$ to K:
      i. Compute $r_{ikm} = y_{ik} - p_k(x_i),\ i = 1, 2, \dots, N$.
      ii. Fit a regression tree to the targets $r_{ikm},\ i = 1, 2, \dots, N$, giving terminal regions $R_{jkm},\ j = 1, 2, \dots, J_m$.
      iii. Compute
      $$\gamma_{jkm} = \frac{K-1}{K} \frac{\sum_{x_i \in R_{jkm}} r_{ikm}}{\sum_{x_i \in R_{jkm}} |r_{ikm}|(1 - |r_{ikm}|)},\ j = 1, 2, \dots, J_m.$$
      iv. Update $f_{km}(x) = f_{k,m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jkm} I(x \in R_{jkm})$.
3. Output $\hat{f}_k(x) = f_{kM}(x),\ k = 1, 2, \dots, K$.

Ensemble Learning - XGBoost

$$L^{(m)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}^{(m-1)} + \phi_m(x_i)\right) + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

- 모델에 대한 규제화
- 결측치 학습 기능
- 특성변수를 임의로 일부만 뽑아 사용
- 자료의 크기가 작은 정형화 자료에서는 가장 강력한 머신러닝 기법

Ensemble Learning - LightGBM



- Leaf-wise 증가하는
  의사결정나무 사용

- GOSS 기법

- Exclusive feature bundling

reference

자료

19-2 STAT424 통계적 머신러닝 - 박유성 교수님

교재

파이썬을 이용한 통계적 머신러닝 (2020) – 박유성

ISLR (2013) - G. James, D. Witten, T. Hastie, R. Tibshirani

The elements of Statistical Learning (2001) - J. Friedman, T. Hastie, R. Tibshirani

Hands on Machine Learning (2017) - Aurelien Geron