Ensemble Model 1
(Bagging, Random Forest)

Jeonghun Yoon

Terms

Resampling

교차 검증

Bootstrap부트스트랩

Ensemble앙상블

Bagging

Random forest

알고리즘 성능에 대한 현상

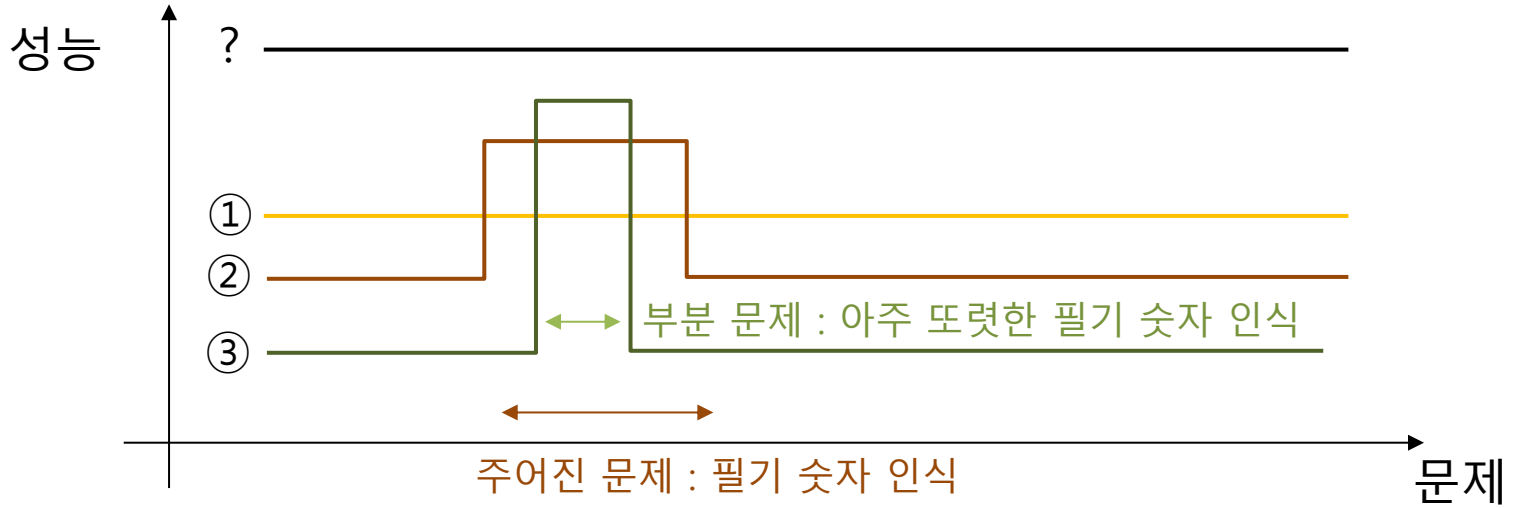
No free lunch

Bias-variance dilemma (trade-off)

No free lunch

모든 문제에 대해 다른 모든 알고리즘을 능가하는 알고리즘이 있나?

- 없다.
- 이론적으로 증명 ⇒ No free lunch(NFL) 정리
  - 어떤 특정 정책에 의해 얼핏 보면 이득을 얻는 것 같지만, 그것은 한 측면의 이득일 뿐이고 반드시 이면에 다른 측면이 있고 그 측면에서 손해가 발생한다.
  - No free lunch theorems for optimization. (David H. Wolpert)
  - 모든 알고리즘은 ①, ②, ③과 같다.



혼성 모델Hybrid model의 필요성

혼성 모델Hybrid model

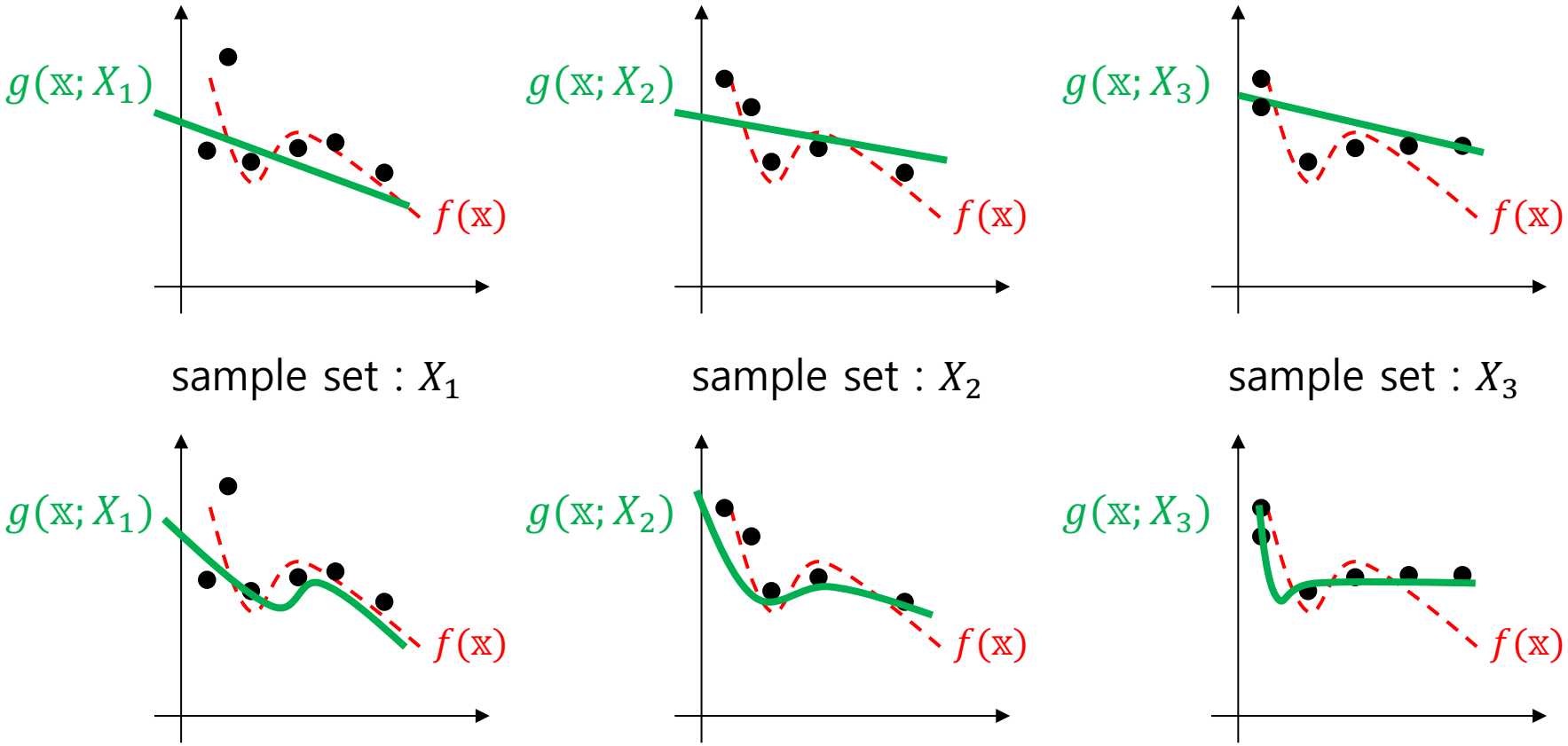
- 여러 알고리즘을 결합하는 모델
- 여러 알고리즘을 결합하면 가장 좋은 단일 알고리즘보다 결과가 좋다는 대다수의 의견

혼성 모델의 도입 배경

- 특정 문제가 주어진 상황에서 그 문제를 가장 높은 성능으로 풀 수 있는 알고리즘에 대한 필요성
- 다른 모든 방법을 능가하는 보편적으로 우수한 알고리즘의 존재에 대한 의문

Bias – Variance Trade off

- 샘플집합 X_i는 $f(\mathbb{x})$로부터 생성
- X를 사용하여 근사치 $g(\mathbb{x})$를 추정 ⇒ $g(\mathbb{x}; X)$로 표현가능
- 일반화 오차 : MSE로 표현 가능 (MSE = Bias^2 + Irreducible Error + Variance)



https://nbviewer.jupyter.org/github/jeonghunyoon/machine-learning-lecture-notes/blob/master/Lecture13_Linear_Regression.pdf?flush_cache=true

Resampling

Resampling

- 데이터의 양이 충분치 않을 때, 같은 샘플을 여러 번 사용하는 것.
- 성능 측정의 통계적인 신뢰도를 높이려는 의도

Resampling의 필요성

- 실제 상황에서는 만족할 만큼 충분히 큰 sample 집합 확보의 어려움
  - bias-variance tradeoff를 통하여, 충분히 큰 sample의 수를 확보하는 것이 중요하다는 것을 알 수 있음
    - 샘플 집합의 크기 N이 커지면 분산의 값이 감소 ⇒ MSE 감소
  - 모델 선택은 별도의 검증 집합이 필요함(큰 sample의 필요성)

교차 검증

Resampling을 이용하여, 분류기의 성능을 측정하는 방법 중 하나(without replacement)

입력 : 분류기 c, 훈련 집합 $X = \{(\mathbb{x}_1, t_1), (\mathbb{x}_2, t_2), \dots, (\mathbb{x}_N, t_N)\}$, $k(2 \le k \le N)$
출력 : 성능 q

알고리즘 :

X를 k개의 부분 집합으로 등분하고 그들을 X_1, X_2,..., X_k라 한다.

for $(i = 1$ to $k)$ {

$X' = \bigcup_{j=1, i \neq j}^{k} X_j$ 로 c를 학습

X_i로 c의 성능을 측정(q_i)

}

return $q = \frac{1}{k} \sum_{i=1}^{k} q_i$

$X = \bigcup_{i=1}^{k} X_i$, X_is are mutually disjoint.

Bootstrap

Statistical term for “roll n-face dice n times”

Resampling을 이용하여, 분류기의 성능을 측정하는 방법 중 하나(with replacement)

Sample이 두 번 이상 뽑히는 경우, 한 번도 안 뽑히는 경우 발생

입력 : 분류기 c, 훈련 집합 $X = \{(\mathbb{x}_1, t_1), (\mathbb{x}_2, t_2), \dots, (\mathbb{x}_N, t_N)\}$, 반복횟수 T
출력 : 성능 q

알고리즘 :

for ($i = 1$ to T) {

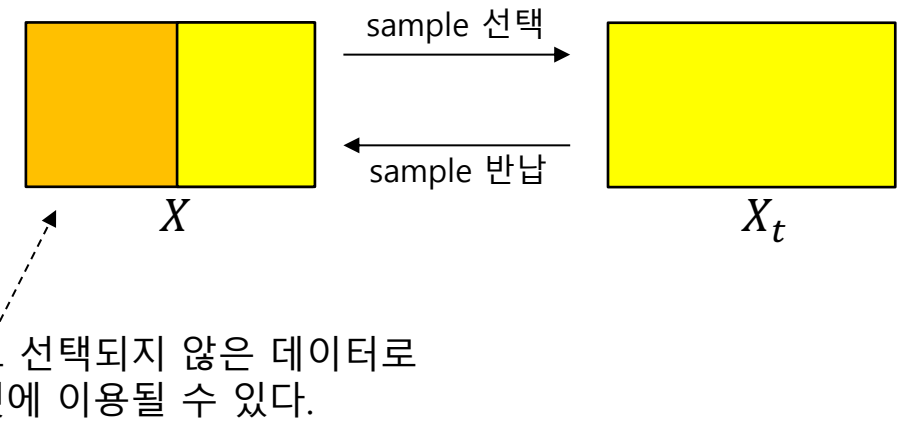
X에서 임의로 N개의 샘플을 뽑아 X_t라 한다. (With replacement)

X_t로 분류기 c를 학습.

$X - X_t$로 c의 성능을 측정하여 q_t라 한다.

}

return $q = \frac{1}{k}\sum_{i=1}^{k} q_i$



Ensemble

혼성 모델 중에 가장 활발히 연구되는 주제

모델의 합집합, 모음

- 같은 문제에 대해 서로 다른 여러 알고리즘이 해를 구하고, 결합 알고리즘이 그들을 결합하여 최종 해를 만드는 방식
- 문제와 유사한 여러 하위 문제들에 대해 하나의 알고리즘이 해를 구하고, 결합 알고리즘이 그들을 결합하여 최종 해를 만드는 방식

Ensemble앙상블

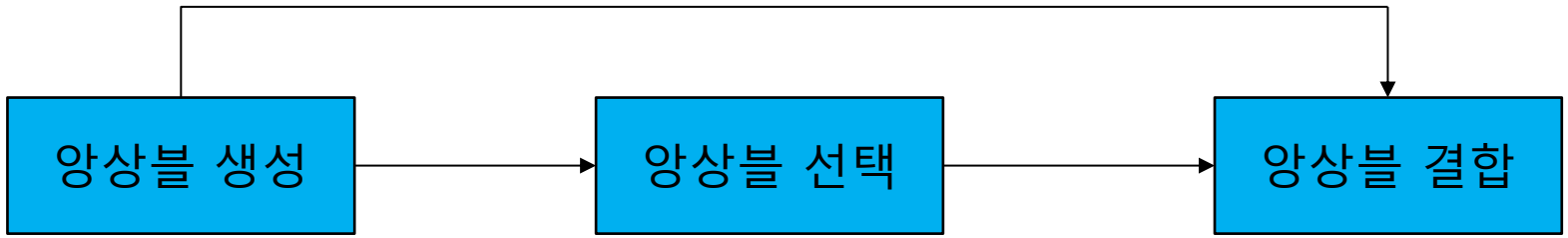
동기

- 여럿의 의견을 종합할 때의 힘
  - 어느 도시에서는 소를 광장에 매어 놓고 참가자들에게 체중을 추정하여 적어 내게 하고 실제 체중에 가장 가까운 사람에게 상품을 주는 대회가 있다고 한다. 수백 명이 참가하는데 그들이 적어낸 숫자들을 평균해 보면 답과 아주 근사하다고 한다.
- 사람들은 중요한 의사 결정을 할 때 여러 사람의 의견을 듣는 습성이 있다.
- 사실 그룹의 중지를 모으는 것이 독립적인 개개인의 지식을 능가할 수 있다는 이 개념은 통계학이 아닌 다른 분야에서도 이미 많이 사용되는 개념이다.

Ensemble

다양성(diversity)

- 앙상블에 참여한 분류기가 모두 같은 분류 결과를 출력한다면, 결합에 의해 얻는 이득은 전혀 없다.
- 어떤 분류기가 틀리는 샘플을 다른 분류기는 맞추는 그런 경향을 내포하는 다양성이 보장되어야 한다.

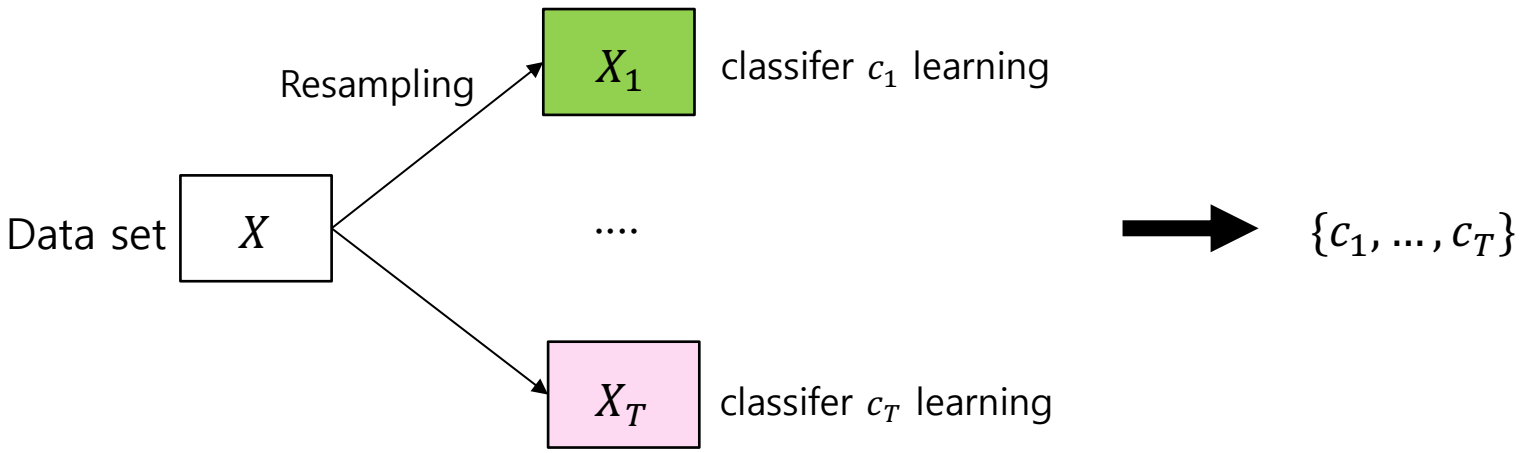


분류기 앙상블 시스템(Classifier Ensemble System)

Ensemble 생성

앙상블 생성(분류기 앙상블 시스템)

- Resampling을 통하여 생성된 샘플 집합들을 이용하여, 분류기들을 각각 훈련
  - Bagging
  - Boosting
- 특징 벡터의 부분 공간을 이용하여, 샘플 부분 집합을 생성 후 분류기를 훈련
  - Random forest
- 앙상블을 구성하는 분류기 : 요소 분류기(component classifier), 기초 학습기(base learner)



Ensemble 결합

요소 분류기(기초 학습기)들의 출력을 결합하여 하나의 분류 결과를 만드는 과정

요소 분류기의 출력

- 부류 표지(class label)
- 부류 순위(class ranking)
- 부류 확률(class probability)

Class label

부류 : M개 / 분류기 : T개 / 1 sample에 대한 분류기의 class vector : $L_1, \dots, L_T$ where $L_i = (l_{i1}, \dots, l_{iM})^T$ and $l_{ij} \in \{0, 1\}$ / α_t : 기본 분류기의 가중치

Majority vote(다수 투표)

$$q = \arg \max_{j=1,\dots,M} \sum_{t=1}^{T} l_{tj}$$

Weighted majority vote(다중 다수 투표)

$$q = \arg \max_{j=1,\dots,M} \sum_{t=1}^{T} \alpha_t l_{tj}$$

- Adaboost에서 사용할 수 있는 방법

Behavior knowledge space(BKS/행위지식공간)

- 분류기가 부류들에 대해서 어떻게 행동하는지를 이용하는 것

Majority vote 예제

샘플 $\mathbb{x}$에 대하여 총 5개의 binary 기본 학습기의 결과가 다음과 같다.

$L_1 = (0, 1)$
$L_2 = (0, 1)$
$L_3 = (1, 0)$
$L_4 = (0, 1)$
$L_5 = (0, 1)$

Class 0에 대하여 : $\sum_{i=1}^{5} l_{0i} = 0 + 0 + 1 + 0 + 0 = 1$
Class 1에 대하여 : $\sum_{i=1}^{5} l_{1i} = 1 + 1 + 0 + 1 + 1 = 4$

다수결 투표에 의하여 class 1로 결론이 난다.

Behavior knowledge space(BKS)

다중 분류기의(c_1, c_2, c_3) 결과 : (l_1, l_2, l_1)

우리는 l_1으로 분류되는 것이라고 확신할 수 있을까?

만약 c_2가 w_2를 99.9999%의 확률로 분류한다면?

$(\mathbb{x}_1, l_2) \rightarrow (l_2, l_2, l_2)$
$(\mathbb{x}_2, l_2) \rightarrow (l_1, l_2, l_1)$
…
$(\mathbb{x}_7, l_2) \rightarrow (l_1, l_2, l_1)$
$(\mathbb{x}_8, l_2) \rightarrow (l_1, l_2, l_1)$
$(\mathbb{x}_9, l_1) \rightarrow (l_1, l_2, l_1)$
$(\mathbb{x}_{10}, l_2) \rightarrow (l_1, l_2, l_1)$
$(\mathbb{x}_{11}, l_3) \rightarrow (l_3, l_3, l_3)$
…
$(\mathbb{x}_{20}, l_2) \rightarrow (l_1, l_2, l_1)$
$(\mathbb{x}_{21}, l_2) \rightarrow (l_1, l_2, l_1)$
$(\mathbb{x}_{22}, l_1) \rightarrow (l_1, l_2, l_1)$
$(\mathbb{x}_{23}, l_3) \rightarrow (l_1, l_2, l_1)$
…

입력 샘플의 참 부류 빈도

BKS 표

다중 분류기 결과	l_1 / l_2 / l_3 (실제)
l_1, l_1, l_1	
l_1, l_1, l_2	
l_1, l_1, l_3	
l_1, l_2, l_1	2 / 6 / 1 (실제)
…	

input sample의 출력 결과가(l_1, l_2, l_1) 이면
l_2에 속한다고 결론을 내릴 수 있음

Class ranking

Borda 계수

- T개 분류기로부터 ranking vector $\mathbb{R}_1, \dots, \mathbb{R}_T$를 얻는다.
  - $\mathbb{R}_t = (r_{t1}, \dots, r_{tM})$
- 이들을 score vector $\mathbb{S}_1, \dots, \mathbb{S}_T$로 변환한다.
  - $\mathbb{S}_t = (s_{t1}, \dots, s_{tM})$
  - 단순하게, $s_{ti} = M - r_{ti}$ 또는 $s_{ti} = \frac{1}{r_{ti}}$를 부여. for $i = 1, \dots, M$
- T개의 점수 벡터를 모두 더한 후 가장 큰 값을 갖는 부류 w_q로 최종 분류

$$q = \arg \max_{j=1,\dots,M} \sum_{t=1}^{T} s_{tj}$$

Softmax

기본 학습기들의 출력값이, 부류, 순위, 확률이 아닌 실수 벡터인 경우

- MLP와 SVM은 부류 별로 실수 값을 출력 (Bayesian classifier : 엄밀한 확률 벡터)
  - 출력 벡터 $\mathbb{O} = (o_1, \ldots, o_M)^T$
- o_i는 확률은 아니고, 단지 느슨한 의미에서 i번째 부류에 속한 신뢰도로 해석 가능
- 신뢰도는 확률로 볼 수 없으므로, 적절한 변환을 거쳐, 확률로 간주
- Softmax

$$p_i = \text{softmax}(o_i) = \frac{e^{o_i}}{\sum_{j=1}^{M} e^{o_j}}$$

where $0 \leq p_i \leq 1, \sum_{i=1}^{M} p_i = 1$

Bagging

부트스트랩을 다중 분류기 생성 기법으로 확장 : **B**ootstrap **agg**regat**ing**

Bootstrapping된 샘플 집합에서 훈련 후, 입력 값에 대하여, 분류기들의 **평균**값 또는 다수결 투표를 취함

- 반복적인 복원 추출 (Bootstrap)
- 결과를 모두 종합 (Aggregation)

Bias↓ variance↑인 분류기에 사용 ⇒ variance↓ (How? average)

- 트리 분류기와 같이 불안정성을 보이는 분류기에 큰 효과를 발휘
- 훈련 집합이 달라지면 차이가 큰 트리가 생성 ⇒ 다양성 확보
- Bias를 변화시키지 않고 variance를 감소시킨다.

Bagging

분산을 감소시키기 위하여 훈련 데이터에서 많은 표본을 취해(bootstrapping) 각 표본별로 별도의 의사결정 트리를 구축한 후,

- 회귀 모형 : 각 의사결정 트리의 결과의 평균
- 분류 문제 : 각 의사결정 트리 결과의 최빈값

을 구하여 하나의 저분산 모델을 얻는다.

Bootstraping 표본 $X_1, X_2, \dots, X_B$에 대한, 각 의사결정 트리 분류기의 예측 : $\hat{f}_1(X_1), \hat{f}_2(X_2), \dots, \hat{f}_B(X_B)$

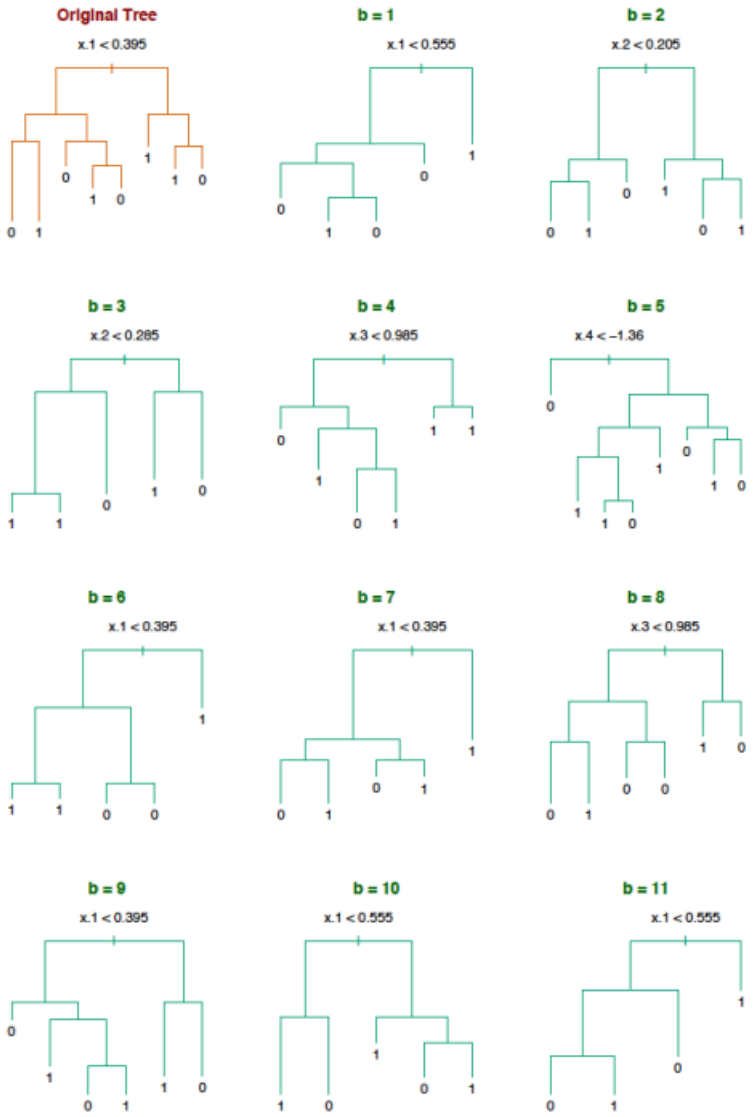
$$\hat{f}_{avg}(X) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_B(X_1)$$

(회귀의 경우)

Example

The Elements of Statistical Learning (8.7.1)

$n = 30$ training data points, 5 features, 2 classes



Bagging Algorithm

입력 : 훈련 집합 $X = \{(\mathbb{x}_1, t_1), (\mathbb{x}_2, t_2), \ldots, (\mathbb{x}_N, t_N)\}$, 샘플링 비율 $\rho(0 < \rho \leq 1)$

출력 : 분류기 앙상블(Classifier Ensemble)

알고리즘 :

$t = 0$, $C = \emptyset$

repeat {

$t = t + 1$

X에서 임의로 ρN개의 샘플을 뽑아 X_t라 한다.(with replacement)

X_t로 분류기 c_i를 학습한다.

$C = C \cup \{c_i\}$

} until (멈춤 조건)

return C

c_i는 서로 독립

When Bagging works

Bagging typically helps

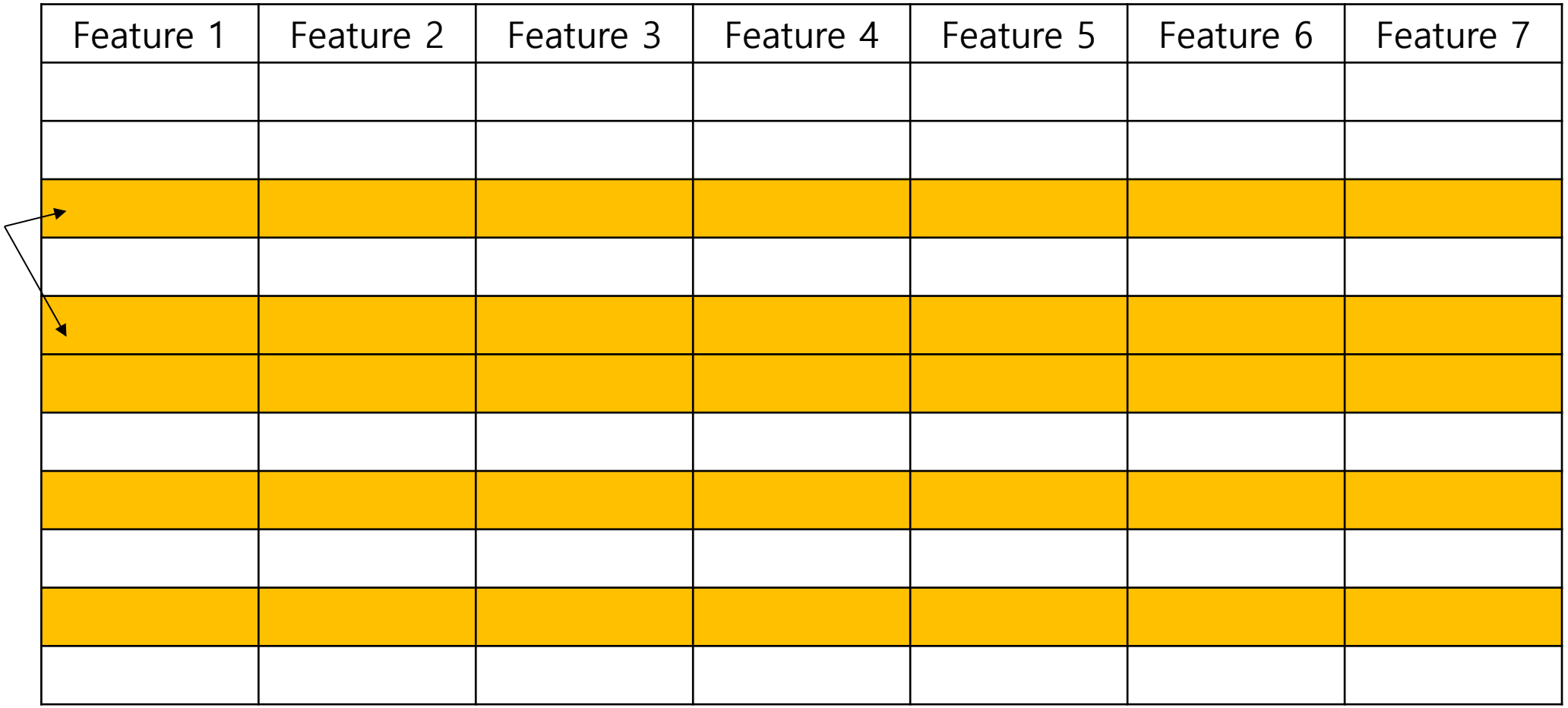
- Over-fitted base model을 사용할 때
- Training data에 높은 의존성을 가진 모델을 사용할 때

Bagging does not help much

- High bias base model을 사용할 때
- Base model이 training data를 변경하는 것에 대해 robust 할 때

Bootstrapping samples

선택된
Bootstrapped
data



Out-of-Bag (OOB) Error Estimation

부트스트랩 샘플은 전체 training observations의 약 2/3($\approx 63.2\%$)를 차지한다.

부트스트랩되지 않은 샘플, 즉 학습에 사용되지 않은 training observations은 **out-of-bag observations**이라고 한다.

이 경우, 부트스트랩 샘플은 학습에 사용하고 OOB에 속하는 샘플을 테스트용(검증용)으로 사용하면 된다.

OOB estimate of test error

- 부트스트랩 샘플을 이용하여 개별 학습기를 학습한 후, OOB에 속하는 샘플들에 대한 예측값을 모두 구한다.
- OOB의 실제 라벨값과 OOB의 예측값을 이용하여 OOB error를 구한다.
- 모든 부트스트랩 샘플 sets에 대하여 위의 과정을 반복하면, 샘플 sets 수 만큼의, errors를 모을 수 있다.
- OOB errors의 평균값을 이용하여 bagging 모델의 최종 테스트 error를 계산한다.

https://www.quora.com/Why-is-the-limit-1-frac-1-n-n-equal-to-frac-1-e

Weakness of Bagging

모든 컬럼을 선택

가장 영향력이 큰 feature라면?

선택된
Bootstrapped
data

Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7

Weakness of Bagging

Bagging은 열을 모두 선택하고, 행을 random하게 선택하는 것이다.

열을 모두 선택하게 되면, 대다수 트리의 결과가 비슷해질 수 있는 문제점이 발생한다.

Why? 대부분의 중요한 변수(컬럼)들이 트리의 초기 분기 때, 모든 표본에 그대로 존재하게 된다. 따라서 주로 중요 변수들로 분기가 이루어질 가능성이 매우 높다. 이것이 반복되면 결국, 트리 간의 상관관계가 형성되어 분산 감소 단계에서 그다지 좋은 결과를 얻지 못하게 될 수 있다.

Weakness of Bagging

$x_1, \dots, x_n$: IID / $E[X] = \mu$ / $Var[X] = \sigma^2$

$$Var\left[\frac{1}{n}\sum_{i=1}^{n} x_i\right] = \frac{\sigma^2}{n}$$

$x_1, \dots, x_n$: correlated / $\forall i \neq j, Corr(x_i, x_j) = \rho$

$$Var\left[\frac{1}{n}\sum_{i=1}^{n} x_i\right] = \rho\sigma^2 + \frac{1-\rho}{n}\sigma^2$$

n이 커지면 $\rho\sigma^2$에 영향을 크게 받게 되고, 결국 averaging의 이점이 사라지게 된다.

Random Forest

Reference : https://www.stat.berkeley.edu/~breiman/RandomForests/

데이터 과학 커뮤니티에서 가장 선호받는 알고리즘 중 하나이다.
여러 대회 우승이나 실제 업계의 문제를 해결하는 데 있어서 많이 사용된다.

Random Forest

Tree 모델에 **bagging**과 **subspace sampling**을 적용한 기법

- 훈련 데이터에서 bootstrapping 표본을 추출 (With replacement)
- 노드 분기시 전체 변수를 모두 사용하지 않고 일부 변수만 사용

Tree간 correlation을 줄임(de-correlated trees' collection) - using randomness

- 평균을 구할 때, 분산을 줄여주는 역할을 한다.

예측

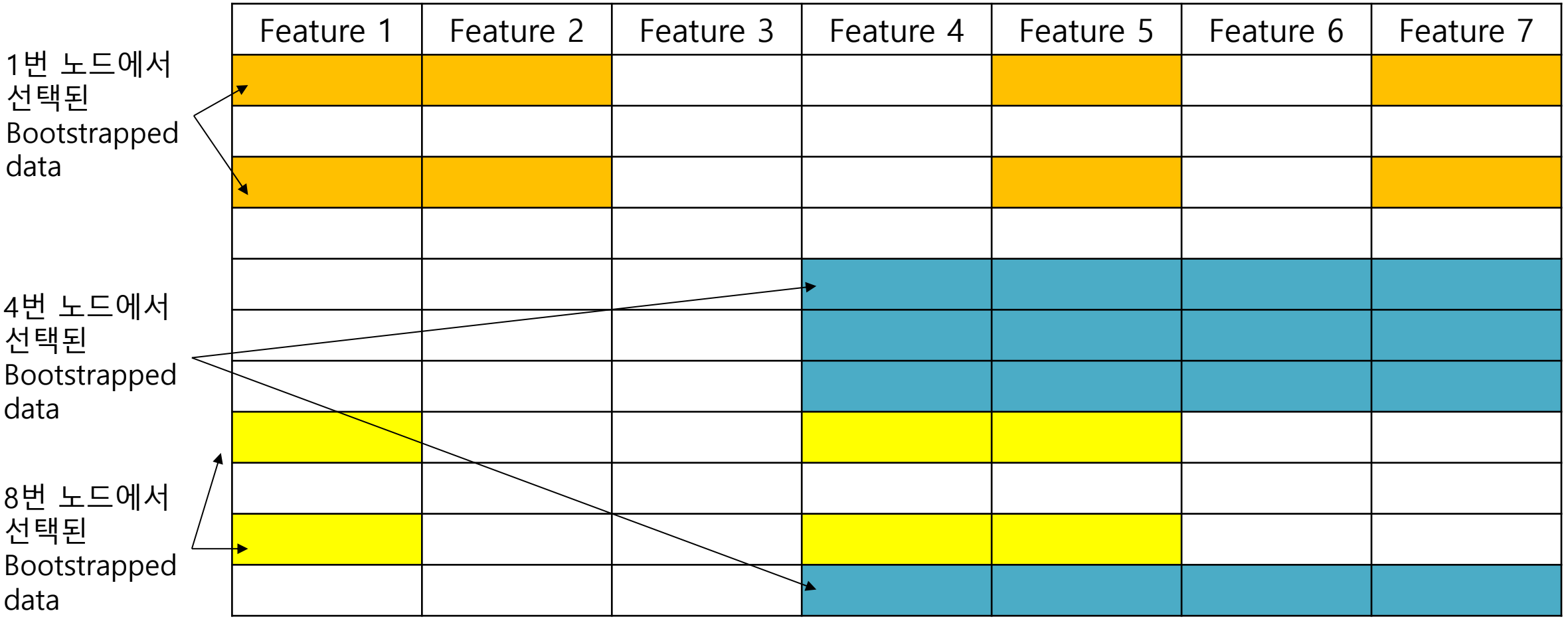
- (Regression) 결과값들의 평균
- (Classification) majority vote

Random Forest

Subspace sampling

- 전체 변수를 p개라고 하면, 분류 문제는 경험적으로 $m = \sqrt{p}$를 사용
  - $m = p$이면, bagging이다.
- 회귀 문제는 경험적으로 $m = \frac{p}{3}$을 사용

Random Forest



Random Forest

입력 : 훈련 집합 $X = \{(\mathbb{x}_1, t_1), (\mathbb{x}_2, t_2), \ldots, (\mathbb{x}_N, t_N)\}$, 샘플링 비율 $\rho(0 < \rho \leq 1)$

출력 : 분류기 앙상블(Classifier Ensemble)

알고리즘 :

$t = 0$, $C = \emptyset$

repeat {

$t = t + 1$

X에서 임의로 ρN개의 샘플을 뽑아 X_t라 한다.(replacement)

(X_t에서 분류기를 학습)

노드에서 split할 때, sample의 p feature 중 random하게 m feature를 선택한다.

선택된 feature들로 생성된 부분 공간 샘플을 후보로 하여 학습한다.

$m \leq \sqrt{p}$ or $\frac{p}{3}$

$C = C \cup \{c_i\}$

} until (멈춤 조건)

return C