

ENSEMBLE LEARNING

Soomin Park

2019. 10. 15



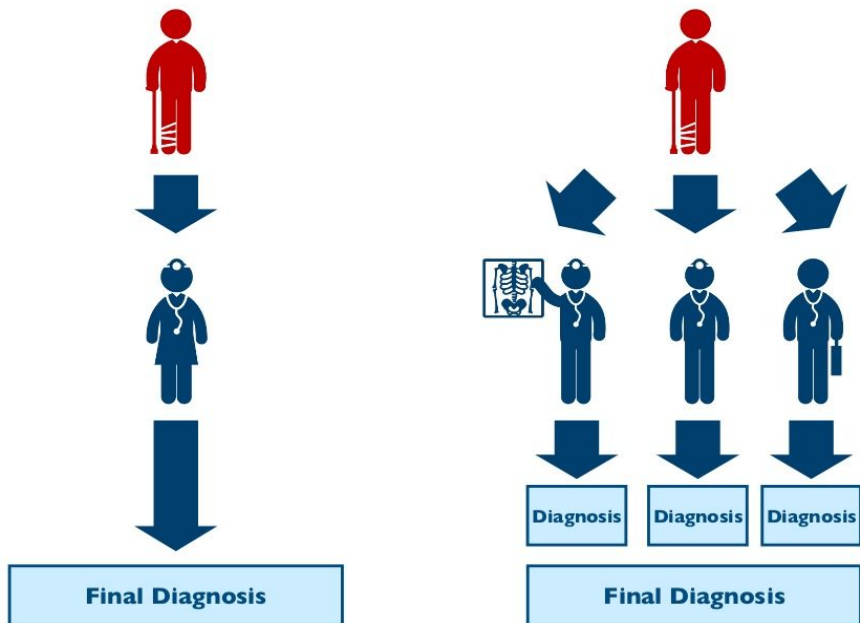
Outline

- 정의
- 종류
- 활용분야
- 실습

?? 앙상블 학습

> 지도학습의 한가지로,
여러개의 모델을 학습, 조합하여
최종 예측모델을 구현한다.

<https://youtu.be/c62uTWdhhMw?t=280>



ENSEMBLE LEARNING

- 다수의 weak learner(단순 구조)들이 상호보완하여 최종 분류 및 예측
 - 의사결정나무(Decision Tree), 경사하강법(Gradient descent), 로지스틱 회귀(Logistic reg), SVM 등
 - 개별 모델 학습에서 일어나는 편향(bias)과 분산(variance) 최소화

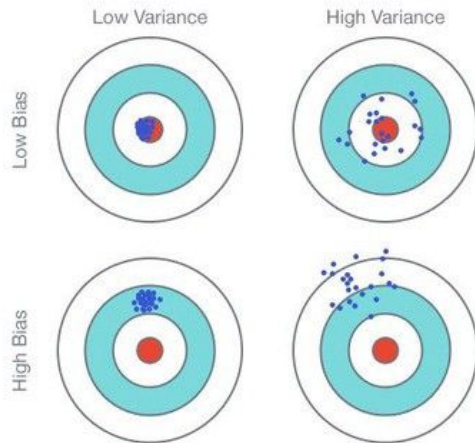
Bias

: 모델의 예측값과 기댓값(실제)의 차이 (under-fit)

Variation

: 새로운 데이터에 대한 예측정도 (over-fit)

$\text{Error}(x) = \text{Bias} + \text{Variance} + \text{error}$

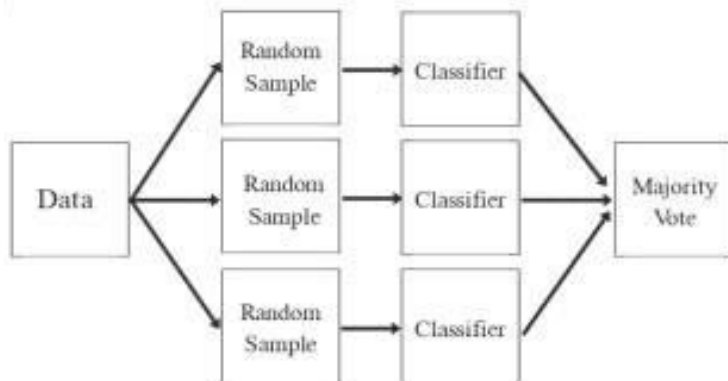


종류

Bagging	Boosting	Stacking
<ul style="list-style-type: none">❑ 병렬 (각 모델은 독립적)❑ Variance감소 (overfit 방지)❑ 동일 알고리즘 <p>Ex. Random Forest</p>	<ul style="list-style-type: none">❑ 직렬 (이전 모델 오류 고려)❑ Bias감소 (underfit 방지)❑ 동일 알고리즘 <p>Ex. AdaBoost</p>	<ul style="list-style-type: none">❑ 병렬 -> 직렬❑ 예측률 증가❑ 다양한 알고리즘

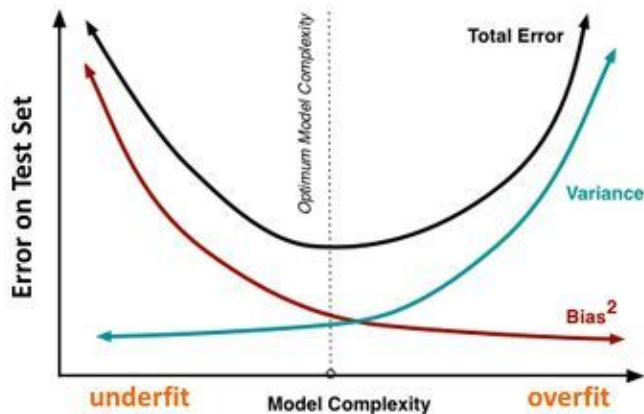
- **Bagging (Bootstrap Aggregating)**

- Bootstrap
 - : n 개의 train 데이터에서 m 개의 동일 알고리즘 learner 학습, 매번 랜덤샘플링을 실시한다.
 - : 서로 다른 m 개의 모델 생성(동일 learner이지만 각각 다른 데이터)
- sum / average / vote 등의 방법을 통해 최종 예측값 생성



- **Bagging (Bootstrap Aggregating)**

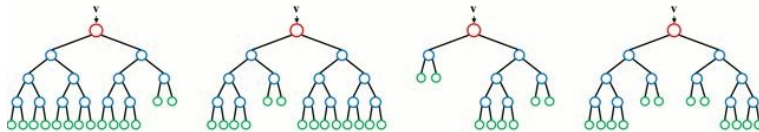
- Variance 최소화 (over fitting 감소)
 - : 가장 뛰어난 것을 뽑거나(voting) 모든 결과를 합/평균 등으로 가공하면서 (sum/average), 분산 오차가 감소
 - : 분산오차가 큰 편인 learner들을 사용 (ex. 의사결정나무, 선형 회귀 등)
 - : 데이터셋 내 분포가 고르지 않을 때 활용 (ex. 99 dogs & 1 cats)



Bagging의 대표 예시

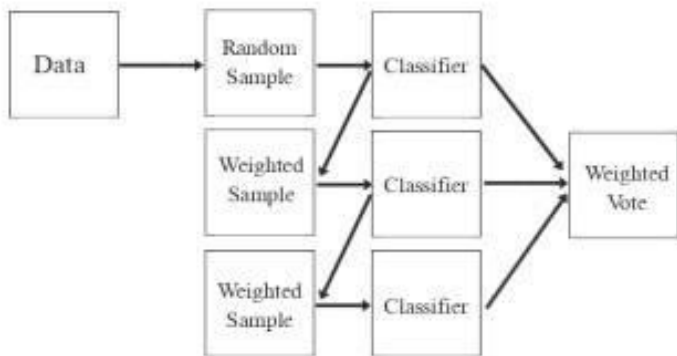
Random Forest

- Bagging의 데이터 랜덤샘플링에 feature 랜덤샘플링을 추가한것
- 의사결정나무 알고리즘 사용 -> 나무(모델)마다 독립변수가 달라진다
ex. A (나이, 성별, 거주지, 수입) -> A' (나이, 성별, 수입) & A'' (나이, 거주지, 수입)
- 개별 의사결정나무의 경우, variance가 매우 크나 (bias낮음).
다양한 조건의 다수의 나무를 훈련하여, 최종적으로 variance도 낮아짐.



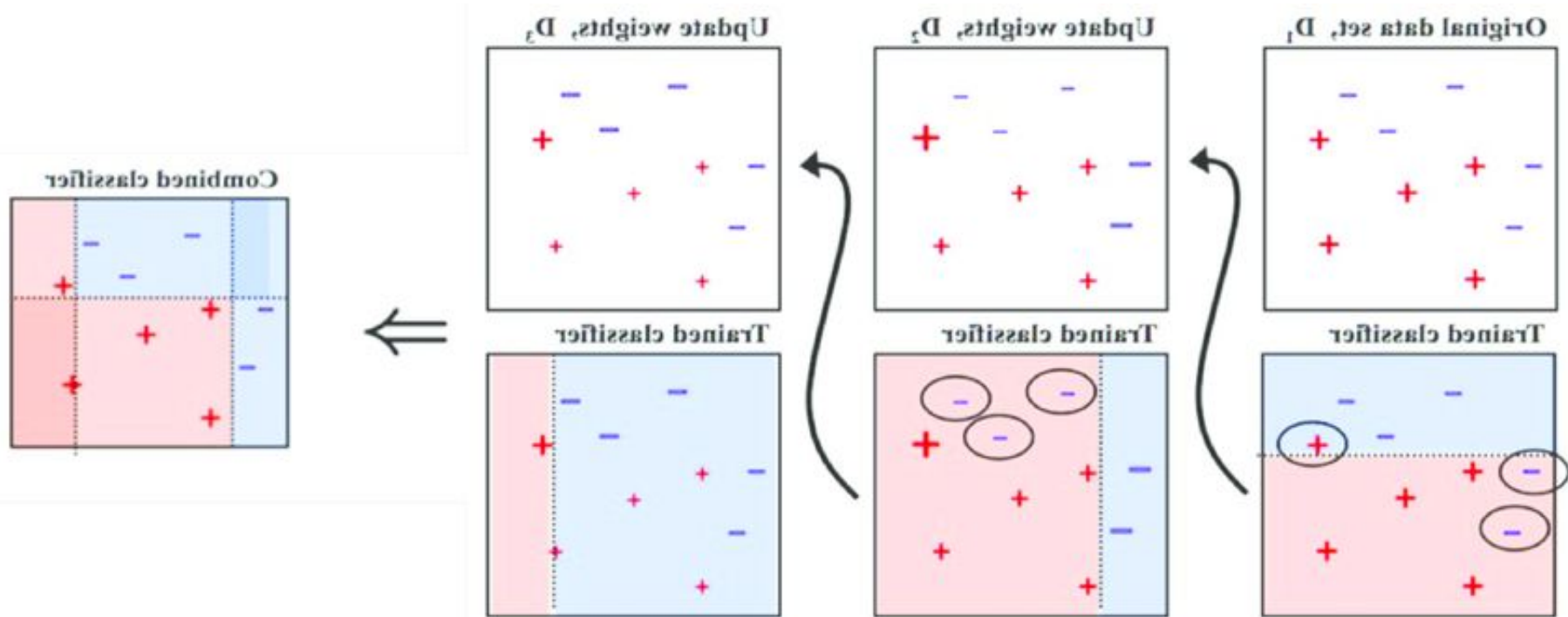
- **Boosting**

- 순차적 진행
- Bias 최소화에 초점
 - : 초반 중요도가 낮았던 feature를 후반부에 가중치를 두어 확인,
 - : 앞의 모델들에서 생기는 오차를 수정하는데에 중점 (overfit 가능성)
- 데이터 랜덤샘플링 -> 1st model
 - > 예측 에러가 있었던 데이터에 weight 추가 -> 2nd model -> ...



BOOSTING

Boosting의 대표 예시

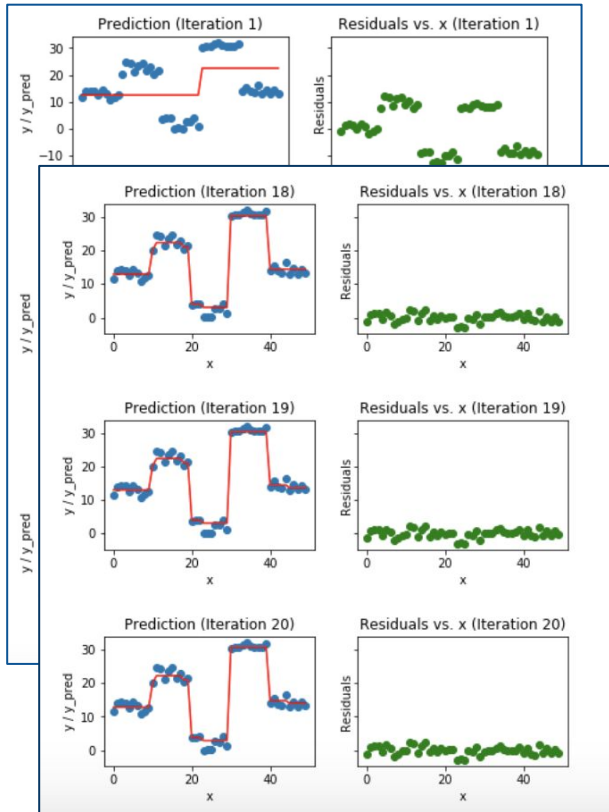


Boosting의 대표 예시

GB (Gradient Boosting)

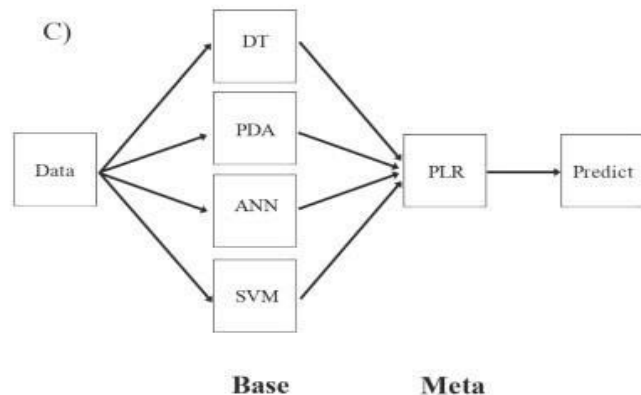
- AdaBoost와 거의 일치 (Boosting 개념에 충실)
- 이전 모델의 잔여 오차에 gradient descent를 사용하는 새로운 learner 적용, 궁극적으로 residual를 0으로 만들고자 한다.

: $err_1 = f(x) + err_2 = f(x) + g(x) + err_3$ 을 통해 error크기를 줄여나갈 수 있음에 착안.



- **Stacking**

- 데이터셋을 train, validation, test로 구분 (k-folds와 비슷)
- Base learners & Meta learner
- train에 대해 다양한 base learners(DT, LR, RF, ANN 등) 학습
 - > validation으로 base models 검증예측
 - > 예측된 validation으로 meta learner 학습 -> test 예측



REFERENCE

https://www.slideshare.net/pierluca.lanzi/dmtm-lecture-10-classification-ensembles?next_slideshow=1

<https://lsjsj92.tistory.com/515?category=853217>

<https://eda-ai-lab.tistory.com/117>

<https://hoony-gunputer.tistory.com/144?category=742600>