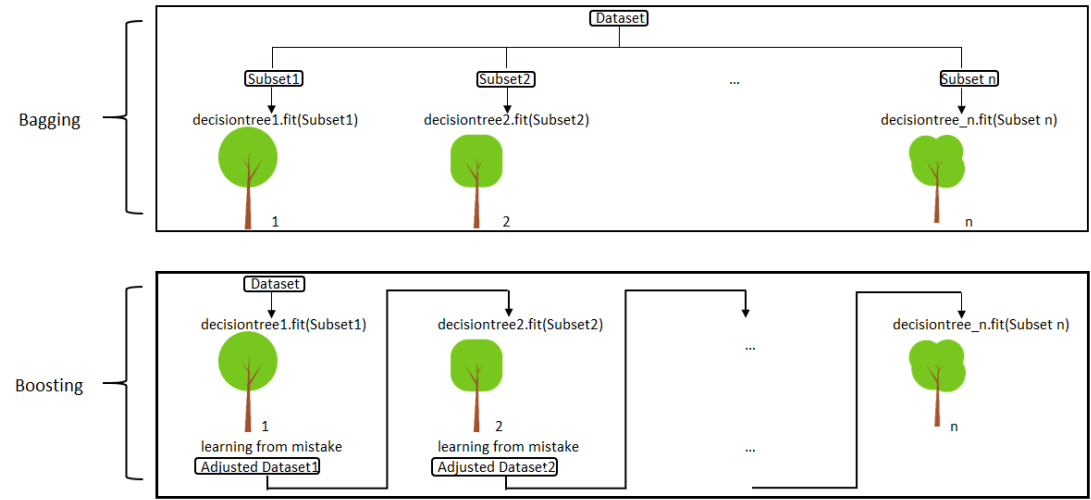
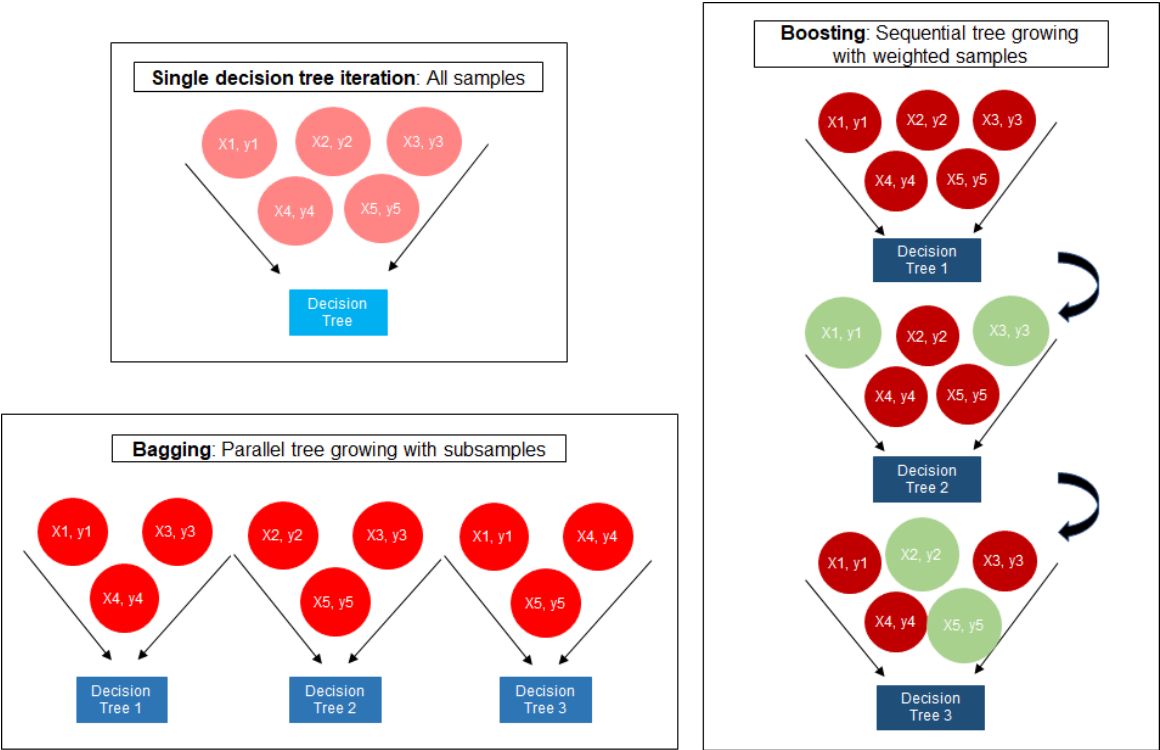


Boosting

- 여러개의 성능이 약한학습기(Weak Learner)를 학습/예측하며, 잘못 예측한 데이터에 가중치를 부여해 오류를 개선해 나가며 강한 학습기(Strong Learner) 를 만드는 방법이다.
- 순차적으로 값을 예측하여 결과를 도출한다
- 가중치 Update 하는 방법을 이용해 앞의 Model을 보완해 나간다.
 - Update 한 가중치를 이용해 Sample을 다시 추출한다.

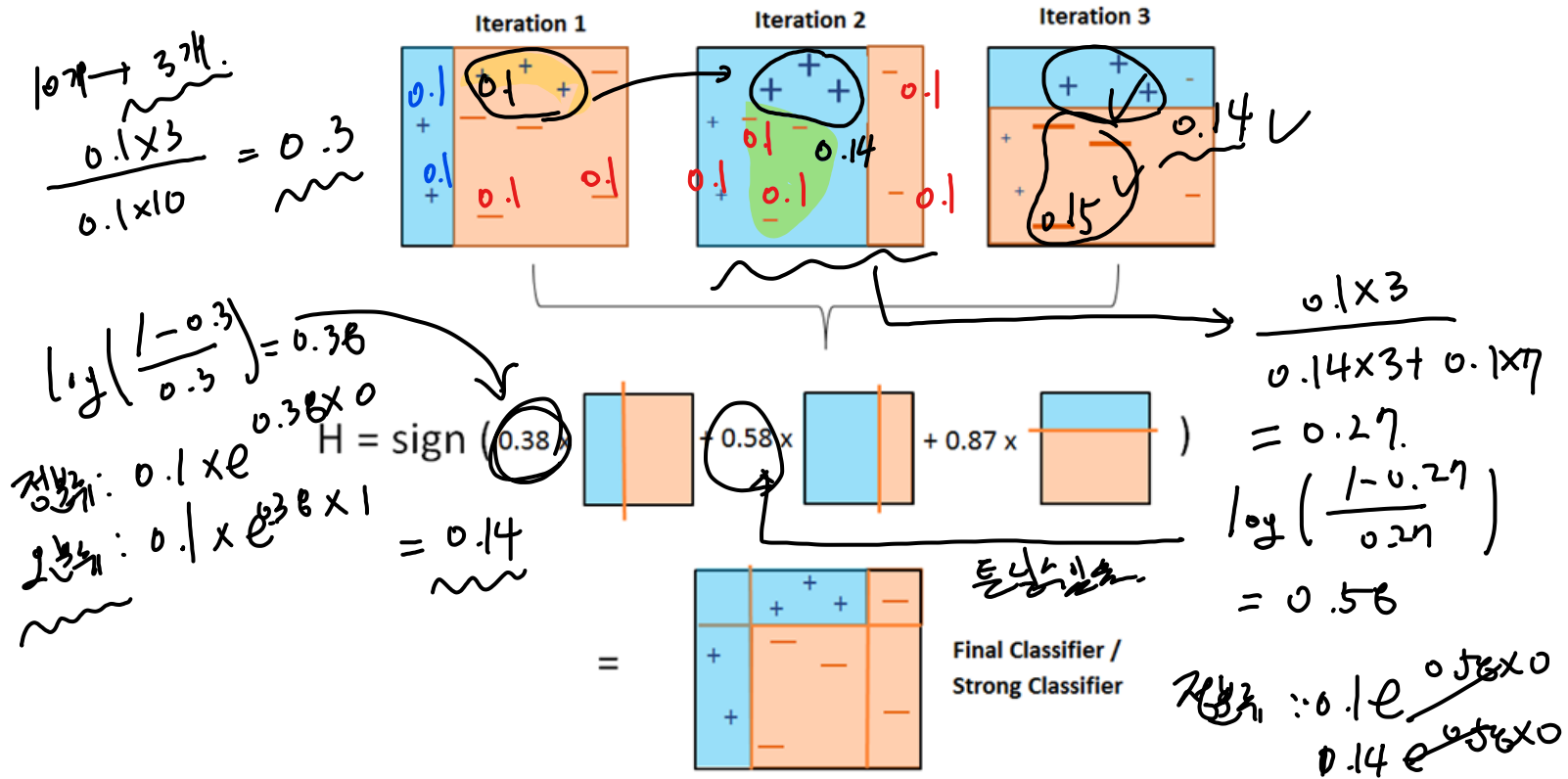


종류

- Adaboost
- GradBoostMachine(GBM) → XGBoost, LightGBM 으로 발전 (정형데이터 분류, 회귀모델에서 GBM 계열이 각광받는다)

AdaBoost

AdaBoost Classifier Working Principle with Decision Stump as a Base Classifier



- **Iteration1** : 첫 번째 Weak Learner가 첫번째 분류기준(D1)으로 + 와 - 를 분류
 - 잘못 분류된 Sample Data에 대해 가중치를 부여(두 번째 그림에서 커진 + 표시)
- **Iteration2** : 두 번째 Weak Learner가 두번째 분류기준(D2)으로 +와 - 를 다시 분류
 - 잘못 분류된 데이터에 대해 가중치를 부여(세 번째 그림에서 커진 - 표시)
- **Iteration3** : 세 번째 Weak Learner가 세번째 분류기준으로(D3) +와 -를 다시 분류해서 오류 데이터를 찾음
- 마지막으로 분류기들에게 가중치를 부여하여 최종 예측 수행

≡ 파라미터 명	Aa 설명
base_estimators	- 학습에 사용하는 알고리즘 - Default = None → DecisionTreeClassifier(max_depth=1)가 적용
n_estimators	- 생성할 약한 학습기의 갯수를 지정 - Default = 50
learning_rate	- 학습을 진행할 때마다 적용하는 학습률(0~1)- Weak learner가 순차적으로 오류 값을 보정해나갈 때 적용하는 계수 - Default = 1.0

GBM(Gradient Boost Machine)

- AdaBoost와 유사하지만, 가중치 업데이트시 경사하강법(Gradient Descent) 를 사용한다.
- $h(x) = y - f(X)$ 와 같은 오류식의 최소화 하기위해 가중치를 업데이트 한다.
 - y : 정답 Label
 - X : input data
 - $f(X)$: 학습을 통한 예측값

Why gradient?

- $L = \frac{1}{2} \sum_{i=1}^n \{y_i - f(x_i)\}^2$

- $\frac{\partial L}{\partial f(x_i)} = -\{y_i - f(x_i)\}$
- Reidual = Negative Gradient

Original		Iteration1		Iteration2	
x ₁	y ₁	x ₁	y ₁ -f ₁ (x ₁)	x ₁	y ₁ -f ₁ (x ₁)-f ₂ (x ₁)
x ₂	y ₂	x ₂	y ₂ -f ₁ (x ₂)	x ₂	y ₂ -f ₁ (x ₂)-f ₂ (x ₂)
x ₃	y ₃	x ₃	y ₃ -f ₁ (x ₃)	x ₃	y ₃ -f ₁ (x ₃)-f ₂ (x ₃)
x ₄	y ₄	x ₄	y ₄ -f ₁ (x ₄)	x ₄	y ₄ -f ₁ (x ₄)-f ₂ (x ₄)
x ₅	y ₅	x ₅	y ₅ -f ₁ (x ₅)	x ₅	y ₅ -f ₁ (x ₅)-f ₂ (x ₅)
x ₆	y ₆	x ₆	y ₆ -f ₁ (x ₆)	x ₆	y ₆ -f ₁ (x ₆)-f ₂ (x ₆)
x ₇	y ₇	x ₇	y ₇ -f ₁ (x ₇)	x ₇	y ₇ -f ₁ (x ₇)-f ₂ (x ₇)
x ₈	y ₈	x ₈	y ₈ -f ₁ (x ₈)	x ₈	y ₈ -f ₁ (x ₈)-f ₂ (x ₈)
x ₉	y ₉	x ₉	y ₉ -f ₁ (x ₉)	x ₉	y ₉ -f ₁ (x ₉)-f ₂ (x ₉)
x ₁₀	y ₁₀	x ₁₀	y ₁₀ -f ₁ (x ₁₀)	x ₁₀	y ₁₀ -f ₁ (x ₁₀)-f ₂ (x ₁₀)


$y = f_1(x)$
 $y - f_1(x)$

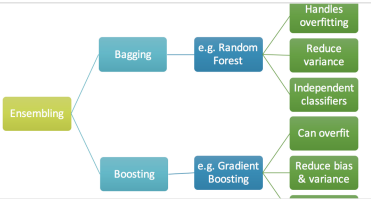
$y - f_1(x) = f_2(x)$
 $y - f_1(x) - f_2(x)$

$y - f_1(x) - f_2(x) = f_3(x)$
 $y - f_1(x) - f_2(x) - f_3(x)$

Gradient Boosting from scratch


Simplifying a complex algorithm Although most of the Kaggle competition winners use stack/ensemble of various models, one particular model that is part of most of the ensembles is some variant of Gradient Boosting (GBM) algorithm. Take for example the winner of the latest

 <https://blog.mlreview.com/gradient-boosting-from-scratch-1e317ae4587d>



```

graph LR
    Ensembling --> Bagging
    Ensembling --> Boosting
    Bagging --> RF[e.g. Random Forest]
    Boosting --> GB[e.g. Gradient Boosting]
    RF --> HOF[Handles overfitting]
    RF --> RV[Reduce variance]
    RF --> IC[Independent classifiers]
    GB --> CO[Can overfit]
    GB --> RBV[Reduce bias & variance]
    
```

Machine-Learning/01_Gradient_Boosting_Scratch.ipynb at master · groverpr/Machine-Learning
You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.
 https://github.com/groverpr/Machine-Learning/blob/master/notebooks/01_Gradient_Boosting_Scratch.ipynb

groverpr/**Machine-Learning**

Notes for machine learning

2Contributors

0Issues

201Stars

167Forks

XGBoost(eXtra Gradient Boost)

- CPU 병렬처리, GPU 지원 → 빠른 수행시간
- Early Stopping(추가)
 - 특정 반복 횟수만큼 더이상 cost function 이 감소하지 않으면 학습을 조기에 종료한다.