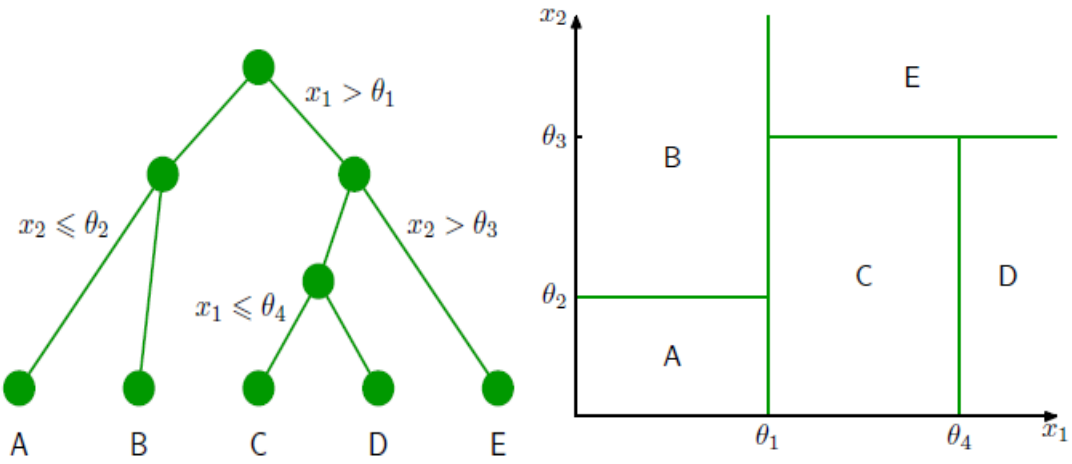
CHAPTER 8 Tree-Based Methonds

1. Decision Tree

Decision tree 데이터를 기준으로 어떤 현상을 예측/분류할 때 사용되는 방법론

Input space를 어떤 기준에 따라 여러 개의 공간으로 분할하는 것



사람이 의사결정 하는 과정과 비슷하기 때문에 해석력이 좋다는 장점

나무가 deep 해질수록 over-fitting 하기 쉬워진다는 단점

Decision tree를 생성하는 알고리즘에는 CART, ID3, C4.5, Quinlan 등 여러 종류가 있지만,

대중적으로 사용되는 것은 CART(Classification And Regression Tree)이다.

이름에서 알 수 있듯이regression 문제와 classification 문제 둘 다 해결 가능.

Regression 문제에서의 척도

-> residual sum of squares

$$y_\tau = \frac{1}{N_\tau} \sum_{\mathbf{x}_n \in \mathcal{R}_\tau} t_n$$

<optimal prediction for region>

$$Q_\tau(T) = \sum_{\mathbf{x}_n \in \mathcal{R}_\tau} \{t_n - y_\tau\}^2$$

<Residual sum of squares>

leaf nodes are indexed by $\tau = 1, \ldots, |T|$

region $\mathcal{R}_\tau$

input space having N_τ data points

Classficiation 문제에서의 척도

-> cross-entropy or Gini index

$$Q_\tau(T) = \sum_{k=1}^{K} p_{\tau k} \ln p_{\tau k}$$

<cross-entropy>

$$Q_\tau(T) = \sum_{k=1}^{K} p_{\tau k} (1 - p_{\tau k})$$

<Gini index>

$p_{\tau k}$ to be the proportion of data points in region $\mathcal{R}_\tau$ assigned to class k,

CART 에서 decision tree를 구성하는 방법

-> greedy algorithm

모든 경우의 수를 계산해서 최적의 해를 얻으면 좋겠지만 현실적으로 불가능

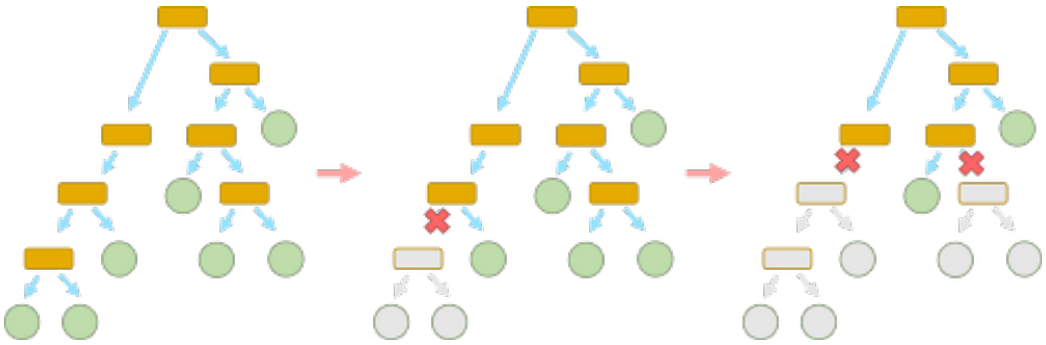
매순간 최적이라고 생각되는 것을 선택해 나가는 방식으로 진행하여

최종적인 최적해에 도달하는 기법

Decision tree 에서는 각 노드에서 어떤 변수를 기준으로, 어떤 값을 기준으로 할 지 결정

그렇다면 Decision tree를 얼마나 깊게 구성해야 할까?

->모르겠다. 일단 모델을 깊이 만든 후 줄여 나가는 방식 Pruning (가지치기)



$$C(T) = \sum_{\tau=1}^{|T|} Q_\tau(T) + \lambda|T|$$

<pruning criterion>

regularization parameter λ

Error 와 모델의 복잡도는 trade-off 관계에

이 cost function을 최소로 하는 지점에서 pruning이 일어남.

Decision tree의 장단점 정리

장점

인간의 사고 과정과 비슷해 해석력이 뛰어나다.

Regression 문제와 classification 문제 둘 다 다룰 수 있다.

단점

Over-fitting 되기 쉽다.

데이터의 특성이 특정 변수에 수직/수평적으로 구분되지 못할 때 분류율이 떨어진다.

2. Ensemble Learning

Decision Tree의 높은 variance를 가짐.

이를 보완하기 위한 방법이 Ensemble Learning

-> 하나의 모델에 여러가지 데이터를 학습시키자!

어떻게 데이터 셋을 만들어 내는지에 따라서

Bagging, Random Forest, Boosting 세 가지 방법

2-1 Bagging(Bootstrap aggregating)

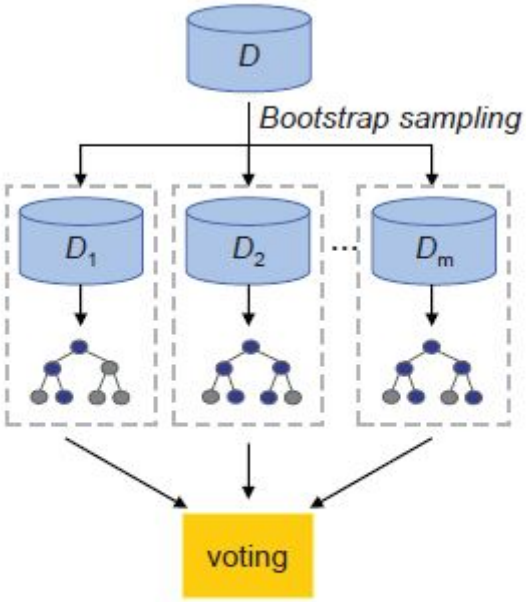
Input data를 bootstrap을 이용해 여러 데이터 셋을 만든다.

각각의 데이터 셋에 대해 Decision Tree를 생성한다.

Regression 문제 에서는 예측 값들의 평균

Classification 문제 에서는 제일 많이 나온 class

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x).$$



(a) bagging

bagging 의 단점

tree에 활용되는 변수 선정에 대한 고민이 없다.

만약 영향력이 강한 한 변수가 있으면 decision tree 간에 높은 correlation 생긴다.

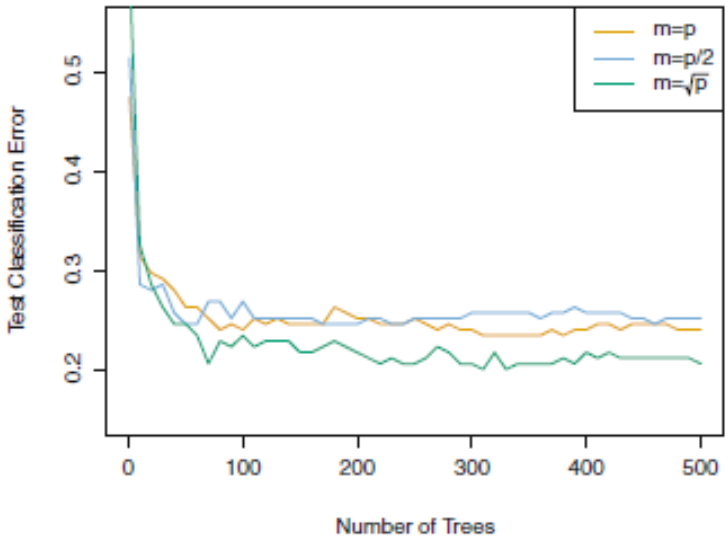
2-2 Random Forest

Bagging 기초적인 방법은 비슷함

Decision Tree를 구성할 때 변수를 다 이용하지 않고 변수 중 m개만 이용하자!

그렇다면 몇 개의 변수를 이용해야 할까?

->총 변수의 제곱근개 만큼 이용



Radom forest의 단점

보통 예측력은 높아지지만 해석력이 떨어진다.

2-3 Boosting

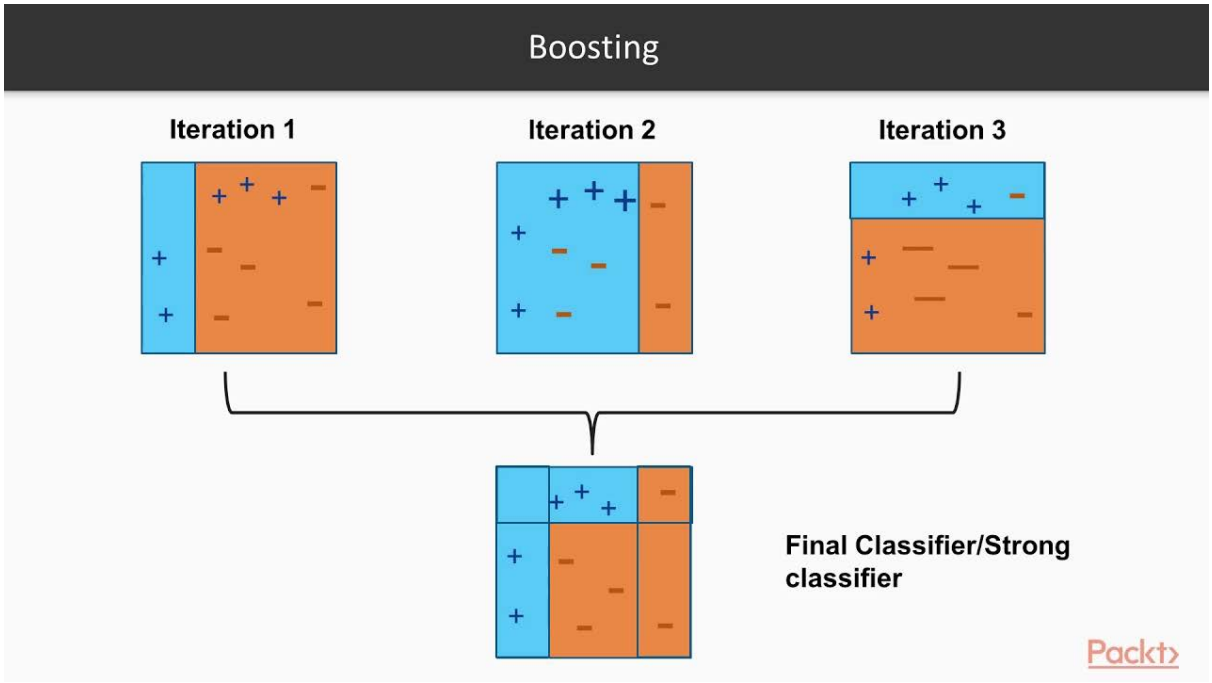
sequential한 weak learner들을 여러 개 결합하여 예측 혹은 분류 성능을 높이는 알고리즘

bagging의 경우 각각의 학습기들이 학습시에 상호 영향을 없이 학습이 끝난 후 그 결과를 종합하는 기법이었다면, Boosting은 이전 학습기의 학습 결과를 토대로 다음 학습기의 학습 데이터의 샘플가중치를 조정해 학습을 진행하는 방법

AdaBoost, Gradient Boost가 가장 많이 쓰임

AdaBoost

오류 데이터에 가중치를 부여하면서 부스팅을 수행



Iteration 1과 같이 +를 분류 실패하면 Iteration 2에서 +의 크기를 키워(가중치를 키워) 다음 약한 학습기가 더 잘 분류할 수 있도록 진행

Adaboost는 위와 같이 약한 학습기가 순차적으로 오류 값에 대해 가중치를 부여한 예측 결정 기준을 모두 결합해 예측을 수행

Gradient boost

Adaboost와 유사한데, 어떻게 가중치를 부여하는지가 다르다.

가중치를 부여하는데 있어서 Gradient Descent를 사용.

조금 더 자세한 내용이 궁금하면 참고하면 좋을 것 같습니다!

https://hyunlee103.tistory.com/25