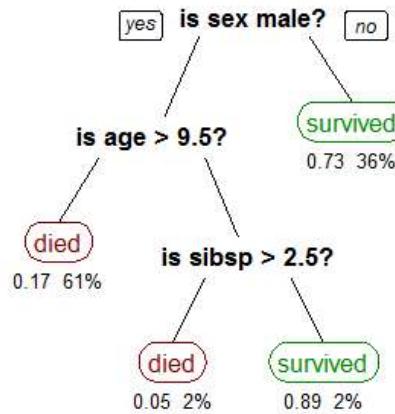


Contents ↗ *

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1966)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Alg
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

Decision Tree

1 Function approximation



2 Decision Tree Model

2.1 Function approximation

- target과 features 사이의 관계를 tree 구조로 구현한 모형이다.
- Nonparametric 함수로 이론적으로는 이산집합을 정의역으로 하는 모든 형태의 함수를 표현할 수 있다.
- Tree 모형의 목적은 target 함수를 가장 잘 묘사하는 approximation을 구하는 것이다.
- Tree 모형에선 각 x_k 축에 직각인 초평면으로 X 를 분할한다.
- x_k 가 boolean이라면 decision region의 수(가능한 전체 leaf nodes의 수)는 $|X| = 2^k$ 이다.
- 각 leaf nodes에 $Y = 0, 1$ 을 대응시킬 수 있으므로 모형의 가짓수는 $2^{|X|} = 2^{2^k}$ 가 된다.
- 따라서 모든 feature 조합에 다른 값을 대응시킬 수 있다. 이러한 구조로 인해 tree model은 information entropy와 좋은 궁합을 이룬다.
- Tree로 함수를 근사하기 위해서는 중복되지 않은 표본들 만큼의 leaf nodes가 필요하지만 충분한 수의 quality data를 얻는 것이 불가능한 경우가 많으므로 tree 구조에 대한 제약을 사용한다.

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic approach
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boundary
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1986)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

2.1.1 decision/data tree의 구조

- Node: 구조를 표시하는 기본 단위. 정보를 포함하거나 다른 node와의 관계를 통해 의사결정 구조를 표시
 - Root node: tree의 제일 첫 (위) node
 - Internal nodes: attribute가 배치되고 가지가 나누어지는 node
 - Leaf nodes (leaves, terminal nodes): class label을 할당하는 node로 children node가 없다.
 - Parent node: 바로 직전 node
 - Children node: 바로 직후 node
- Branches, edges: nodes를 연결하는 선분
- Depth: Root와 leaf node들 사이 edge 수 중에서 가장 큰 수
tree에 제약을 가할 때 depth와 number of leaves를 흔히 사용한다.

2.1.2 Feature and target spaces

- tree 모형의 이상적인 자료형태는 categorical variable이다.
- 연속변수는 discretize 시켜 사용하지만 정보의 손실이 많고 breakpoint 결정도 자명하지 않다.
- 표본이 무한히 크지 않다면 최적 threshold를 정확히 결정하기 어렵다.
- Leaf node가 지나치게 많아질 수 있으며 overfitting 문제의 통제가 쉽지 않다.

2.1.3 Top-down greedy heuristic approach

- Top-down approach는 분석을 단계별로 진행하면서 각 단계마다 해결할 문제를 조그만 문제로 나누어가면서 답을 구하는 방식이다.
- Tree의 구성
 1. Node에 배치할 최선의 attribute A를 선택
 2. A의 값에 따라 새로운 children node를 생성
 3. Leaf node에 표본을 배치
 4. 성과를 높일 여지가 남아 있다면 각 leaf node에서 위 과정을 반복
- Feature space의 분할(partition)
 - Partition $\{R_1, \dots, R_p\}$ 의 각 $R_j \subset X$ 를 각 target class에 대응시킨다.
 - 분할은 R_p 을 순차적으로 나누어 가면서 진행한다.
 - 모형에 따라 한 번에 여러 개의 영역으로 분할하기도 하고 두 개로만 분할하기도 한다.
- Greedy algorithm은 매 단계마다 주어진 상황에서 최적의 해를 찾아가는 방법이다. 전역적인 극대점을 찾는 것이 아니므로 초기 선택에 따라 완성된 tree가 다를 수 있다.
- Heuristic method(from Greek "eurisko", "find, search, discover")는 효율성을 위해 optimality를 어느정도 포기하고 실용적인 관점으로 문제를 해결하는 방법이다.

Contents ↗

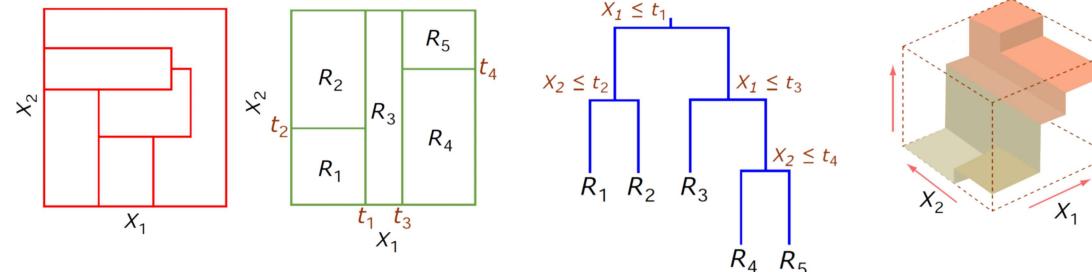
- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boundary
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1966)
 - 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - 4.2.5 Error-Based (post) Pruning Alg
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

2.2 Training a tree model

2.2.1 Decision region, decision boundary, decision surface

- 분류모형은 X 공간을 각 class에 대응하는 decision region으로 나누는 것이다.

Figure 3. Decision Region



source: Mihaela van der Schaar (2017) lecture note

2.2.2 Tree 설계

- sklearn.tree.DecisionTreeClassifier에는 tree의 구조를 결정하는 여러 keyword option이 있다.
- Decision tree의 구조
 - feature의 배열 순서 (splitter)
 - tree의 길이
- 경계값 선택 (threshold value, breakpoint, 연속변수에만 해당)
- 각 leaf의 output 결정 (function value, label 지정)

2.2.3 Overfitting

- 분기分歧 split, 분할 partition
 - 분기split할 node와 이에 적용할 attribute 선택
 - attribute를 다수의 영역으로 분할하고 추가적인 분기를 할 것인지 결정
 - 필요할 경우 가지들을 통합
- 가지치기 pruning
 - 완성한 tree를 validation data으로 검증하여 기준에 맞지 않는 가지를 제거한다.
 - rule setting을 이용한 방법을 주로 사용한다.
- 분기와 가지치기의 결정
 - Mean Squared Error: regression 과 classification
 - Gini Impurity: Classification

$$\sum_{i \in R_j} (y_i - \text{prediction}_i)^2$$

$$\sum_{i \in R_j} \{p_c \times (1 - p_c)\}$$

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision bound
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1966)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - 4.2.5 Error-Based (post) Pruning Alg
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

- 분할 횟수(number of depth)의 제한
 - 각 leaf에 도달하기 위해 사용하는 attribute의 수나 분할 횟수를 사전에 결정하여 사용
 - 한 leaf에 속한 표본의 수가 사전에 정한 임계값보다 작을 때
 - 분기 후 leaf 사이의 표본들이 통계적으로 서로 다르지 않을 때

2.2.4 Feature importances

- Tree의 특징을 생각해보면 어떤 node와 해당 children node 사이의 정보량에 별 차이가 없다면 그 node에 할당된 attribute은 적어도 그 위치에서는 분류에 기여하는 바가 없다고 볼 수 있다.
- 각 attribute는 모형에 따라 tree에서 한번만 사용하기도 하고 반복해서 사용할 수도 있다.
- Feature importance는 tree의 이러한 특징들을 고려하여 상대적인 중요성을 계산한다.
- Children nodes가 두 개일 경우 해당 node의 중요도는 다음과 같이 계산할 수 있다. (Sklearn Criterion의 `importance_improvement` method)

$$f_{kj} = \begin{cases} \frac{n_j}{n} \left(g_j - \frac{n_{jL}}{n_j} g_{jL} - \frac{n_{jR}}{n_j} g_{jR} \right) & \text{if } x_k \text{ is assigned to node } j \\ 0 & \text{otherwise} \end{cases}$$

- f_{kj} : node j 에 배정된 feature k 의 중요성
- g_j : node j 의 Gini impurity
- n_j : number of observations at node j
- 하침자 j_L, j_R 은 각각 node j 의 왼쪽과 오른쪽 child node
- Feature k 가 지정된 모든 node들의 중요도를 더해 feature k 의 중요도를 계산한다.

$$f_k = \frac{\sum_j f_{kj}}{\sum_k \sum_j f_{kj}}$$

- Decision tree를 비롯해 tree를 기반으로 한 몇몇 ensemble 모형에서는 각 attribute의 상대적 중요성을 확인할 수 있다.
- Feature selection 용도로 사용하기도 한다.

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic approach
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boundary
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - ▼ 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1983)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity pruning
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

3 분기의 결정

- 추가되는 정보의 양이 많은 feature를 먼저 배치한다.
- 추가되는 정보의 양이 일정 수준 이상되는 경우 leaf를 증가시킨다.
- 평가 기준
 - sum of squared error
 - likelihood
 - goodness-of-fit
 - information gain
 - Gini impurity
 - misclassification error
 - prediction accuracy
- 표본의 값에 대응하는 information entropy는 각 class 값에 해당하는 확률질량함수의 로그변환에 (-1)을 곱한 후 모두 더해 계산한다.

$$\text{information entropy} = H(X) = - \sum_{c=1}^C \Pr(x_c) \log_b \Pr(x_c) = -E \log_b \Pr(X_c)$$

- 이때 $0 \times \log 0 = \log 0^0 = 0$ 로 정의한다.

A property of information entropy

- entropy는 불확실성의 척도로서 개별 사건에 대한 정보를 전달하기 위해 필요한 평균적인 정보의 양을 측정한다.
- 4가지 종류의 전투기가 동일한 비율로 있을 때 다음과 같은 code를 사용할 수 있다.

| type | probability | code | no. of bits |
|------|-------------|------|-------------|
| 전투기 | 0.25 | 11 | 2 |
| 훈련기 | 0.25 | 10 | 2 |
| 폭격기 | 0.25 | 01 | 2 |
| 수송기 | 0.25 | 00 | 2 |

- 각 유형별 항공기의 정보를 표시하기 위해서는 평균 2 bits가 필요하다.

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - ▼ 2.1 Function approximation
 - 2.1.1 decision/tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - ▼ 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - ▼ 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1983)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

- 최적 code는 $-\log_2 \Pr(x)$ bits를 사용하는 것이다.

| type | probability | code | no. of bits |
|------|-------------|------|-------------|
| 전투기 | 0.50 | 1 | 1 |
| 훈련기 | 0.25 | 01 | 2 |
| 폭격기 | 0.125 | 001 | 3 |
| 수송기 | 0.125 | 000 | 3 |

$$-\frac{1}{2} \times 1 + \frac{1}{4} \times 2 + 2 \times \frac{1}{8} \times 3 = 1.75$$
$$\text{entropy} = -\sum_{x \in \text{values}(X)} \Pr(x) \log_2 \Pr(x)$$

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic approach
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boundary
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - ▼ 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1966)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - 4.2.5 Error-Based (post) Pruning
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

3.1 예제: Decision Tree for PlayTennis (Tom Mitchell)

- 예제를 이용한 tree 모형의 설명은 Quinlan의 ID3, C4.5 algorithm에 해당한다.

| Day | Outlook | Temperature | Humidity | Wind | PlayTennis |
|-----|----------|-------------|----------|--------|------------|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

- 자료에는 두 개의 class가 있으며, positive와 negative가 각각 9개와 5개이므로

$$\text{entropy} = - \left(\frac{9}{14} \log_2 \frac{9}{14} + \frac{5}{14} \log_2 \frac{5}{14} \right) = 0.4098 + 0.5305 = 0.9403$$

Contents ↗

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic approach
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boundary
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1986)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

3.2 Information gain

- Information gain은 추가정보로 인한 entropy의 변화를 의미하며 정보이득이 크다면 entropy는 크게 감소한다.
- Attribute A 의 information gain은 A 이외의 다른 attributes를 동일하게 유지한 상태에서 A 가 entropy 감소(불확실성의 감소)에 기여한 바를 표시한 것이다.
- S 를 표본이라고 하면 attribute A 의 정보이득은 다음과 같다.

$$\text{Information Gain}(S, A) = \text{IG}(S, A) = H(S) - H(S|A)$$

- Conditional entropy의 정의를 생각해보자.

$$H(Y|X) = - \sum_x p(x) \sum_x H(Y|X=x)$$

- 여기서 $p(x)$ 는 표본비율을 사용하여 표본의 조건부 entropy는 다음과 같다.

$$H(S|A) = \sum_{a \in \text{supp } A} \frac{|S_A(a)|}{|S|} H(S_A(a))$$

- $S_A(a)$ 는 전체 표본에서 attribute A 의 값이 a 인 자료를 의미하며,
 $\frac{|S_A(a)|}{|S|}$ 은 전체 표본에서 attribute A 의 값이 a 인 표본의 비율,
 $H(S_A(a))$ 는 해당 표본의 entropy이다.
- Entropy의 극소화 문제와 information gain의 극대화 문제는 동일한 해를 갖는다.

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - ▼ 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1983)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Alg
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

3.2.1 Information gain

- 아래 예는 Temperature를 분류에 사용할 경우 정보이득을 계산하는 예이다. 동일한 방식으로 다른 feature들의 정보이득을 계산하여 크기를 비교한다.
- Overfitting을 방지하기 위해서는 사전에 결정한 threshold와 비교해서 결정한다.

| | Temp | target | Hot | Mild | Cool |
|----|------|--------|-----|------|------|
| D1 | Hot | No | No | | |
| D2 | Hot | No | No | | |
| D3 | Hot | Yes | Yes | | |
| D4 | Mild | Yes | | Yes | |
| D5 | Cool | Yes | | | Yes |
| D6 | Cool | No | | | No |
| | ... | | | | |

- Attribute Temperature에 대한 표본의 조건부 entropy는 그 값이 각각 Hot, Mild, Cool일 때 entropy의 가중평균이다.
- 제일 왼쪽 열의 entropy와 나머지 세 열에 해당하는 entropy의 가중평균과의 차이가 Temperature의 information gain이다.

1. Temperature 정보를 사용하지 않을 경우 information entropy:

$$H(S) = 0.940$$

1. Temperature 정보를 사용할 경우 information entropy:

- $IG(S, x) = H(S) - H(S|x)$ 이고,

$$H(S|\text{Temperature}) = \sum_{T \in \{\text{Hot}, \text{Mild}, \text{Cool}\}} \frac{|S_{\text{Temp}}(T)|}{|S_{\text{Temp}}|} H(S_{\text{Temp}}(T))$$

이므로 각 값에 대한 조건부 entropy를 먼저 계산한다.

- $(Y|\text{Hot}) = (2+, 2-), (Y|\text{Mild}) = (4+, 2-), (Y|\text{Cool}) = (3+, 1-)$ 이므로

$$H(S|\text{Hot}) = - \left(\frac{1}{2} \log \frac{1}{2} + \frac{1}{2} \log \frac{1}{2} \right) = 1$$

$$H(S|\text{Mild}) = - \left(\frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3} \right) = 0.918$$

$$H(S|\text{Cold}) = - \left(\frac{1}{4} \log \frac{1}{4} + \frac{3}{4} \log \frac{3}{4} \right) = 0.811$$

$$H(S|\text{Temperature}) = \frac{4}{14} \times 1 + \frac{6}{14} \times 0.918 + \frac{4}{14} \times 0.811 = 0.911$$

- 따라서

$$\text{Information Gain}(S, \text{Temperature}) = 0.940 - 0.911 = 0.029$$

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1983)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

- Information Gain(S , Outlook) =
- Information Gain(S , Humidity) =
- Information Gain(S , Wind) =

3.3 Gini impurity (Gini Index)

- Impurity는 말그대로 주어진 node가 얼마나 순수한지를 비교하기 위한 척도이다.
- Gini impurity는 무작위로 추출한 집합의 원소에 같은 분포를 따르는 딱지들을 무작위로 대응시켰을 때 표시가 잘못될 확률이다.
- 세 개의 빨간 공과 한 개의 파란 공이 들어있는 주머니에서 공을 하나 꺼내 빨간 딱지 세 개와 파란 딱지 하나 중에서 무작위로 선택한 딱지를 하나 붙인다고 생각해 보자.

| 사건 (ball/tag) | $p = \Pr(\text{ball color})$ | $q = \Pr(\text{tag color})$ | probability |
|---------------|------------------------------|-----------------------------|-------------|
| red & red | 3/4 | 3/4 | 9/16 |
| red & blue | 3/4 | 1/4 | 3/16 |
| blue & red | 1/4 | 3/4 | 3/16 |
| blue & blue | 1/4 | 1/4 | 1/16 |

- 같은 색의 딱지를 붙일 확률은 10/16, 다른 색을 붙일 확률은 6/16이다. 따라서 이 상태의 impurity는 3/8이다.
- 쉽게 확인할 수 있듯이 하나의 공을 바르게 대응시킬 확률은 $p \times q$ 이고 잘못 대응시킬 확률은 $p(1 - q)$ 이다.
 $p = q$ 이므로 잘못 대응시킬 확률은 $p(1 - q) + q(1 - p) = 2p(1 - p)$ 가 된다.
- Class c 의 비율이 π_c 라고 하면 $(1 - \pi_c)$ 는 잘못된 label을 붙일 확률이다.

$$\text{Gini}(\pi) = \sum_{c=1}^C \left(\pi_c \left(\sum_{i \neq c} \pi_i \right) \right) = \sum_c (\pi_c \times (1 - \pi_c)) = \sum_{c=1}^C \pi_c - \sum_{c=1}^C \pi_c^2 = 1 - \sum_{c=1}^C \pi_c^2$$

Gini Gain = Gini impurity before split - Gini impurity after split

- 분기 이후의 impurity는 각 node의 impurity를 해당 node들에 포함된 표본 수로 가중평균한 값을 사용한다.

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1983)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

3.4 Classification Error

- 주어진 표본을 무작위로 분류했을 경우 잘못된 딱지를 불일 최소한의 오차률이다. 가장 단순한 기준이지만 class가 두 개일 경우에는 다른 기준들과 별다른 차이가 없다.

$$\text{Classification Error} = 1 - \max_c p_c$$

- 오류는 표본분포에 따라 딱지를 준비하고 무작위로 표본과 딱지를 하나씩 뽑았을 때 class가 일치하지 않는 사건
 - Class c 인 표본을 바르게 구분할 확률은 전체 표본에서 해당 class의 비율이므로 p_c 이다.
 - 바르게 구분할 확률이 가장 큰 경우는 표본에서 차지하는 비중이 가장 큰 class에서 표본이 뽑혔을 경우이다.
 - Classification error는 무작위분류가 잘못될 확률 중 가장 작은 값이 된다.

3.5 Binary decision trees에서 분할기준의 비교

- Information entropy, information impurity :

$$-(p \log p + (1-p) \log(1-p))$$

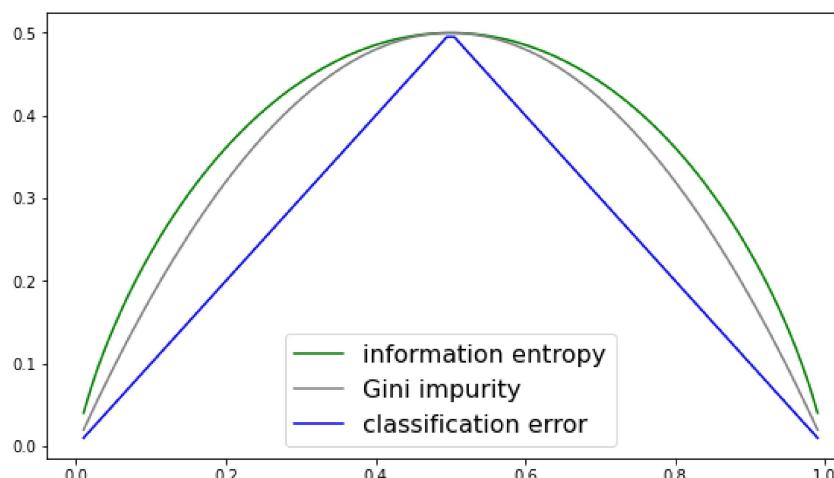
- Gini impurity :

$$2p(1-p) = 1 - (p^2 + (1-p)^2)$$

- Classification/Misclassification error :

$$1 - \max(p, 1-p)$$

Figure 7. Impurity Measures



Contents ↗

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1986)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
 - 5 Regression tree
 - 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
 - 7 Titanic dataset
 - ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
 - 9 Regression Tree
 - 10 분류모형의 비교

4 Algorithms

4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1986)

- 앞의 모형 선택을 설명하면서 사용한 algorithm이 ID3이다.
- 장점: 직관적이고 쉬운 결과 해석
- 단점: tree의 분기를 언제 중지해야 할지를 명시적으로 고려하지 못한다.
 1. 연속적인 attribute를 다루지 못한다.
 2. attributes에 missing value가 있으면 안된다.
 3. 의미없는 가지가 남아있다.
- 4. overfitting의 가능성이 높고 따라서 예측 정확도가 낮다
- 5. K 가 크면 information gain에 편이가 발생한다.
- 6. C 가 크면 curse of dimensionality가 발생한다.
- Occam's razor: 가능하면 단순한 형태의 모형이 바람직하다.
상대적으로 단순한 모형이 자료를 잘 설명한다면 우연으로 보기 어렵다.
- Properties of ID3 (Tom Michell's lecture note)
 1. Hypothesis space is complete
 2. outputs a single hypothesis

"You can't play 20 questions with nature and win", (Newell, 1973)

 3. No back tracking - possible local optimum
 4. Statistics-based search - robust to noise
 5. Prefer short tree

Contents ↗

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1986)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
 - 5 Regression tree
 - ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
 - 7 Titanic dataset
 - ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
 - 9 Regression Tree
 - 10 분류모형의 비교

4.2 C4.5 (Ross Quinlan, 1993)

- C4.5 algorithm은 ID3의 단점을 보완하기 위해 고안
- ID3와 C4.5의 python code는 [여기서](https://brianbartoldson.wordpress.com/2018/10/10/id3-and-c4-5-decision-trees/) (<https://brianbartoldson.wordpress.com/2018/10/10/id3-and-c4-5-decision-trees/>) 구할 수 있다.
- 최근 발표한 [C5](https://www.rulequest.com/download.html) (<https://www.rulequest.com/download.html>)는 C4.5에 비해 메모리 사용량과 사전정의된 규칙의 수가 작은 상업용

4.2.1 gain ratio

- 분할에는 information gain을 표준화한 gain ratio를 사용한다. 예를 들어 범주가 2개인 이산변수와 10개인 이산변수의 분기를 information gain이란 동일한 기준으로 비교하는 것은 적합하지 않으므로 정보의 양으로 gain을 정규화한 것이다.

$$\text{GainRatio}(S, A) = \frac{\text{Gain}(S, A)}{\text{SplitInfo}(S, A)}$$

$$\text{SplitInfo}(S, A) = - \sum_{a \in \text{supp}A} \frac{|S_A(a)|}{|S|} \log_2 \frac{|S_A(a)|}{|S|}$$

4.2.2 Continuous attributes

- Attributes가 이산변수이거나 연속변수인 경우에 모두 적용할 수 있다.
- Attribute의 개별값이나 Q1, Q2, Q3, 혹은 output class의 값이 변하는 지점을 breakpoint로 사용한다.

4.2.3 Missing value

- missing value를 포함한 자료에 적용할 수 있다.
- missing value가 포함된 attribute A가 node j에 할당된 경우
 - missing value를 제외하고 node j의 information gain을 계산한 후 (1-proportion of missing value)을 곱해서 구한 weighted information gain으로 분기 결정
 - 분기를 할 경우 다음 중 한 가지 방법으로 missing value를 대체한다.
 1. Node j에 포함된 A의 값 중 가장 흔한 값으로 missing value를 대체
 2. Node j에 포함된 표본 중 missing value를 포함한 관찰값의 target과 동일한 표본의 가장 흔한 값으로 A값을 대체
 3. A의 각 값에 확률 p_i 를 할당하고 descendant node에 p_i 의 확률로 해당 표본을 적용
- 결정한 값으로 tree를 계속 훈련

4.2.4 Attribute weights

- 연구의 목적에 따라 attribute마다 다른 가중값을 적용하여 분석할 수 있다.

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - ▼ 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - ▼ 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - ▼ 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1983)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

4.2.5 Error-Based (post) Pruning Algorithm

- ID3나 CART는 algorithm 자체에서 pruning을 지원하지 않는다.
- C4.5에선 한 node의 '정규화한' classification error가 children nodes의 '정규화' classification error보다 작으면 해당 children nodes를 제거하는 방식으로 pruning을 결정한다. '정규화'는 아래 설명을 참고
 - (9+, 5-) → ((4+, 2-), (1+, 1-), (4+, 2-))
- Regression tree는 태생적으로 비선형함수에 대해 취약하므로 비선형성이 심할 경우나 종속변수의 유형이 2개 이상일 경우 pruning 단계를 포함하고 있는 C4.5을 사용하면 overfitting 문제를 '간편하게' 처리할 수 있다.

4.2.5.1 Reduced-Error Pruning

- 전체 표본은 training, pruning, test set, 셋으로 나누어 training set으로 훈련을 마친 tree로 validation set을 test하여 그 성과가 가지치기를 한 tree보다 좋지 않을 경우 해당 가지 전체를 leaf node로 바꾼다.
- Pruning data로 계산한 parent node의 error rate과 children nodes error rate의 가중평균을 비교해 parent node의 error rate이 작으면 해당 branch를 가지치기한다.
- 간단하고 속도가 빠르다.

4.2.5.2 statistical pruning of C4.5

- 앞의 pruning 방법을 사용하면 훈련에 사용할 수 있는 표본의 수가 줄어든다.
- C4.5에선 전체 표본을 훈련에 사용할 수 있도록 신뢰구간을 이용한다.

1. Training data로 tree를 완성한다.

2. Node j 의 실제 예측오차 e_j 를 이용해 '실제' 예측오차의 신뢰구간을 구한다.

$$\text{error rate}_j = e_j \pm z_\alpha \sqrt{\frac{e_j(1-e_j)}{n_j}}$$

3. children nodes 신뢰구간의 오른쪽을 구한다. $\text{error rate}_j = \frac{1}{n_j} \left[e_j + z_\alpha \sqrt{\frac{e_j(1-e_j)}{n_j}} \right]$

- $\frac{1}{n_j} \left[e_j + z_\alpha \sqrt{\frac{e_j(1-e_j)}{n_j}} \right]$

1. Children nodes 신뢰구간의 오른쪽이 parent node 신뢰구간의 오른쪽보다 크다면 (신뢰구간이 더 넓으므로) 해당 branch를 제거한다.

- z 는 z-score로 C4.5의 default는 유의수준 0.50에 해당하는 0.67이다.
- 다음과 같은 regularized version을 사용하기도 한다.

$$\text{error rate}_j(\alpha) = \frac{e_j + \frac{z_\alpha^2}{2n_j} + z_\alpha \sqrt{\frac{e_j(1-e_j)}{n_j} + \frac{z_\alpha^2}{4n_j}}}{1 + \frac{z_\alpha^2}{n_j}}$$

4.2.5.3 Rule post pruning

- 1. 생성한 tree를 set of rules로 표현
- 2. 각 rule을 독립적으로 평가하고 필요할 경우 제거
- 3. 선택한 rule을 적당한 순서로 정리하여 사용

Contents ↗ *

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1983)
 - 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - 4.2.5 Error-Based (post) Pruning Alg
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression tree)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

4.3 CART, C&RT (classification and regression tree, Breiman et al., 1984)

- C4.5와 차별되는 특징
 - Gini impurity (criterion)
 - binary tree
 - no rule sets
 - consider multiple features for split (max_features)
 - missing의 경우 mean imputation 과 유사하게 instances가 많은쪽으로 구분

5 Regression tree

- Regression tree에선 각 leaf에 속한 관찰값과 평균과의 차이 제곱을 기준으로 분기를 하며,
- 설정한 깊이에 달하거나 새로운 분기로 loss를 감소시키지 못할 때 분기를 종료한다.

$$MSE = \sum_{j=1}^J \sum_{i \in R_j} (y_i - \text{avg}(y_{R_j}))^2$$

- Parent node의 MSE, MSE_j 와 children node의 weighted MSE, $MSE_{j,\text{children}}$ 를 비교하여 눈에 띠는 개선이 없을 때 분기를 중지한다.

$$\begin{aligned} MSE_j &= \sum_{i \in R_j} (y_i - \text{avg}(y_{R_j}))^2 \\ MSE_{j,\text{children}} &= \frac{n_{jL}}{n_j} \times MSE_{j,\text{left}} + \frac{n_{jR}}{n_j} \times MSE_{j,\text{right}} \end{aligned}$$

6 Sklearn Implementation

6.1 DecisionTreeClassifier, CART

- [sci-kit learn \(<https://scikit-learn.org/stable/modules/tree.html>\)](https://scikit-learn.org/stable/modules/tree.html)의 tree algorithm은 CART를 optimize 한 version이다.
- Sklearn에선 Gini impurity와 함께 information entropy (log_loss)를 loss function으로 지원한다.
- Information gain과 Gini impurity는 [이론적 \(\[https://www.unine.ch/files/live/sites/iml/files/shared/documents/papers/Gini_index_fulltext.pdf\]\(https://www.unine.ch/files/live/sites/iml/files/shared/documents/papers/Gini_index_fulltext.pdf\)\)](https://www.unine.ch/files/live/sites/iml/files/shared/documents/papers/Gini_index_fulltext.pdf)으로는 큰 차이가 없다. 실제 응용에서도 C4.5와 CART를 비교하여 보면 어느 한 algorithm의 성과가 다른 모형에 비해 의미있는 차이가 없는 것으로 알려져 있다.

Contents ⚙

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - ▼ 2.1 Function approximation
 - 2.1.1 decision/tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic approach
 - ▼ 2.2 Training a tree model
 - 2.2.1 Decision region, decision boundary
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - ▼ 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1966)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

6.2 DecisionTreeClassifier

```
sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, ccp_alpha=0.0)
```

- Scikit-learn에서 제공하는 criteria ("gini", "entropy"):
 - Information entropy는 이산 변수에 특화되어 있고 설명변수의 해석에 장점이 있다.
 - Gini impurity는 연속적인 attribute에 적합하며 상대적으로 misclassification 비율이 낮다고 알려져 있다.
- **max_depth**: int, default=None

The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
- max_depth 를 지정하지 않으면 동일한 feature에 다른 target이 대응되지 않는 한 완벽한 분류가 가능하다.
- Overfitting을 줄이기 위한 pruning 용도로 사용할 수 있는 keyword argument들을 제공하고 있다.

6.2.1 Pre-pruning

- cost_complexity_pruning_path method는 pruning 과정에서 계산한 ccp_alpha의 값을 보여준다. 이 값을 이용하여 최적 parameter 값을 결정할 수 있다.
- Pre-pruning은 사전에 결정한 규칙에 따라 분기를 중단하는 방법이다. 다음 keyword들로 값을 설정할 수 있다.
- max_depth, min_samples_leaf, min_samples_split, max_leaf_nodes, min_impurity_decrease, min_impurity_split

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic approach
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boundary
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1966)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity Pruning
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

6.2.2 Post-pruning: Cost-Complexity Pruning (CART)

- Post-pruning은 전체 tree를 구성한 후 terminal node부터 차례로 진행하는 가지치기 방법이다.
- 이 방법은 각 node를 root로 하는 subtree의 cost complexity가 complexity parameter `ccp_alpha` (Cost Complexity Pruning- Alpha)보다 작은 값을 삭제한다.
- Cost-complexity measure는 다음과 같이 정의한다.

$$R_\alpha(T) = R(T) + \alpha|T|$$

- T 를 전체 tree, t 는 node, T_t 는 node t 를 root로 하는 tree라고 하자.
- $|T|$ 는 tree T 의 terminal nodes의 수, $R(T)$ 는 tree T 의 misspecification rate이다.
- Lasso와 유사하게 terminal node의 수에 penalty를 가하는 형태이다.
- Pruning 여부는 해당 branch의 가치, 즉 $R(T_t)$ 와 $R(t)$ 를 비교하여 결정한다.
- 보통은 하나의 node로 이루어진 tree t 의 impurity $R(t)$ 는 $R(T_t)$ 보다 크지만 penalty가 있어 $R_\alpha(T_t)$ 가 $R_\alpha(t)$ 보다 커질 수 있다.
- Pruning 여부는 `ccp_alpha` 과 R_α 의 gain을 비교하여 결정한다.

$$R_\alpha(t) - R_\alpha(T_t) = R(t) - R(T_t) + \alpha - \alpha|T_t| = R(t) - R(T_t) + \alpha(1 - |T_t|) = 0$$

- Threshold α 는 다음과 같다.

$$\alpha^* = \frac{R(t) - R(T_t)}{1 - |T_t|}$$

- $gain$ or `ccp_alpha` 보다 커질 때까지 가지치기를 진행하며, 최적 `ccp_alpha`는 cross validation으로 결정하거나 아래 설명하는 방식으로 계산한다. 참고 (<https://scikit-learn.org/stable/modules/tree.html#minimal-cost-complexity-pruning>).

$$\begin{aligned} R_\alpha(T - T_t) - R_\alpha(T) &= R(T - T_t) - R(T) + \alpha|T - T_t| - \alpha|T| \\ &= R(t) - R(T - T_t) + \alpha(1 - |T_t|) \\ R_\alpha(T - T_t) - R_\alpha(T) &= R_\alpha(t) - R_\alpha(T_t) \\ &= R(t) - R(T_t) + \alpha - \alpha|T_t| \\ &= R(t) - R(T_t) + \alpha(1 - |T_t|) \\ \alpha' &= \frac{R(t) - R(T - T_t)}{|T_t| - 1} \end{aligned}$$

1. $\alpha_0 = 0$ 으로 하고 $R(T)$ 를 극소화하는 tree T^0 를 선택한다.
2. $i = 1, 2, \dots, t$ 는 다음 극소화문제의 해이며 $T^i = T^i - T_{t_i}^i$

$$\alpha_i = \min_t \frac{R(t) - R(T - T_t^i)}{|T_t^i| - 1}$$

3. 이 과정을 반복하여, 모든 결과를 정리하면

$$T^0 \supseteq T^1 \supseteq T^2 \supseteq \dots, \quad \alpha_1 \leq \alpha_2 \leq \dots$$

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic approach
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boundary
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1966)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

6.2.3 Categorical variable

- Categorical variable을 입력으로 사용하지 않는다.
- LabelEncoder \rightarrow OneHotEncoder로 정수로 바꾸어야 한다.
- LabelEncoder는 categorical variable을 $(0, 1, \dots, C)$ 와 같이 정수로 변환하므로 tree 모형은 이 변수를 ordered integer로 취급한다. Ordinal 변수가 아니라면 LabelEncoder는 적합한 변환방법이 아닐 수 있다.
- OneHotEncoder를 사용하는 것이 categorical variable의 정보를 정확히 반영하는 방법이다. Category의 수가 많으면 계산이 부담스러워질 수 있다.

7 Titanic dataset

- 변수명을 일일히 지정하지 않고 자료유형에 따라 다른 변환을 적용할 수도 있다.

```
from sklearn.compose import make_column_selector as selector

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, selector(dtype_exclude="category")),
        ('cat', categorical_transformer, selector(dtype_include="category"))])
```

- make_column_selector를 이용할 때는 사전에 자료 유형을 확인하고 필요하면 변환해야 한다.

```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(transformers=[('num', FunctionTransformer(),
                                                   ['age', 'sibsp', 'parch',
                                                   'fare']),
                  ('cat', OneHotEncoder(drop=[3, 'male',
                                              'S'],
                                         sparse=False),
                   ['pclass', 'sex',
                     'embarked'])])),
              ('classifier', DecisionTreeClassifier(random_state=42))])
```

- Training dataset과 test dataset의 예측 정확성은 f1-score 기준으로 22%의 차이
- Tree의 depth가 20으로 overfitting 문제가 상당히 심각하다.

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1983)
 - 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

depth of the tree: 20
number of the leaves: 216

DecisionTreeClassifier - training dataset

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| died | 0.97 | 0.99 | 0.98 | 474 |
| survived | 0.98 | 0.95 | 0.97 | 309 |
| accuracy | | | 0.97 | 783 |
| macro avg | 0.98 | 0.97 | 0.97 | 783 |
| weighted avg | 0.97 | 0.97 | 0.97 | 783 |

DecisionTreeClassifier - test dataset

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| died | 0.81 | 0.79 | 0.80 | 333 |
| survived | 0.64 | 0.67 | 0.66 | 189 |
| accuracy | | | 0.75 | 522 |
| macro avg | 0.73 | 0.73 | 0.73 | 522 |
| weighted avg | 0.75 | 0.75 | 0.75 | 522 |

- 적당한 depth를 결정하기 위해 cross-validation을 적용한다.
- GridSearchCV의 best_estimator_는 random_state에 상당한 영향을 받는다.
- 주의: Tree 모형은 random_state에의 값에 따라 최적 hyperparameter값이 다르게 나타날 수 있다.

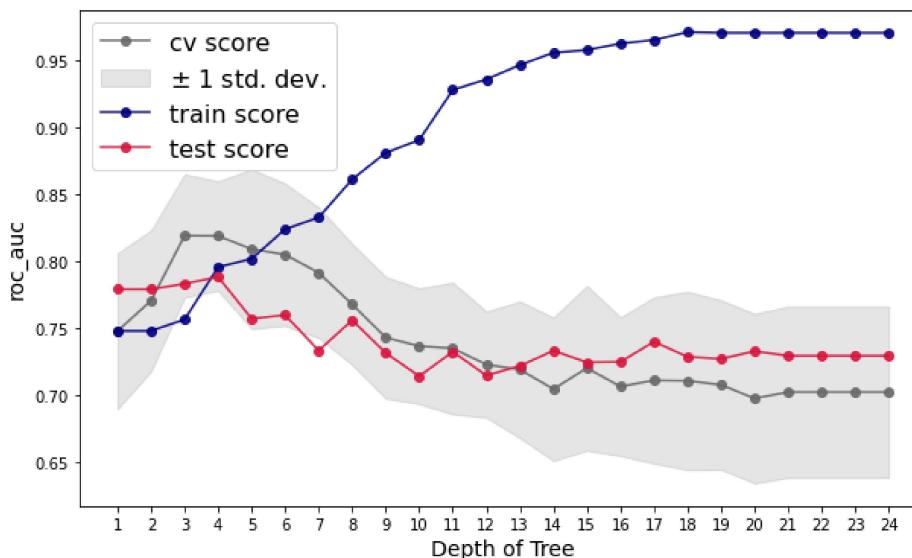
best model: DecisionTreeClassifier(max_depth=3, random_state=42)

```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(transformers=[('num', FunctionTransformer(),
                                                   ['age', 'sibsp', 'parch',
                                                   'fare']),
                                              ('cat',
                                               OneHotEncoder(drop=[3, 'male',
                                                                     'S'],
                                                             sparse=False),
                                               ['pclass', 'sex',
                                                 'embarked'])])),
              ('classifier', DecisionTreeClassifier(random_state=42))])
```

Contents ⚙

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1966)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Alg
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

Cross-validation for best depth



number of the leaves: 8

Best of Decision Trees - training dataset

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| died | 0.80 | 0.83 | 0.82 | 474 |
| survived | 0.73 | 0.68 | 0.70 | 309 |
| accuracy | | | 0.77 | 783 |
| macro avg | 0.76 | 0.76 | 0.76 | 783 |
| weighted avg | 0.77 | 0.77 | 0.77 | 783 |

Best of Decision Trees - test dataset

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| died | 0.84 | 0.87 | 0.85 | 333 |
| survived | 0.75 | 0.70 | 0.72 | 189 |
| accuracy | | | 0.81 | 522 |
| macro avg | 0.79 | 0.78 | 0.79 | 522 |
| weighted avg | 0.80 | 0.81 | 0.80 | 522 |

Contents ↗

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision bound
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1986)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

8 추정 모형에 대한 세부정보

8.1 visualization

- `plot_tree` 를 사용하면 `tree`의 전체 구조와 각 leaf의 구성을 쉽게 확인할 수 있다.

```
sklearn.tree.plot_tree(decision_tree, *, max_depth=None, feature_names=None, class_names=None, label='all', filled=False, impurity=True, node_ids=False, proportion=False, rounded=False, precision=3, ax=None, fontsize=None)
```

8.2 save as pdf

- 추정한 `tree`에 대한 정보는 [export_graphviz \(\[https://scikit-learn.org/stable/modules/generated/sklearn.tree.export_graphviz.html#sklearn-tree-export-graphviz\]\(https://scikit-learn.org/stable/modules/generated/sklearn.tree.export_graphviz.html#sklearn-tree-export-graphviz\)\)](https://scikit-learn.org/stable/modules/generated/sklearn.tree.export_graphviz.html#sklearn-tree-export-graphviz)을 이용하여 Graphviz format으로 출력할 수 있다.

```
sklearn.tree.export_graphviz(decision_tree, out_file=None, *, max_depth=None, feature_names=None, class_names=None, label='all', filled=False, leaves_parallel=False, impurity=True, node_ids=False, proportion=False, rotate=False, rounded=False, special_characters=False, precision=3)
```

- Graphviz format은 graphviz를 설치해서 pdf로 저장할 수 있다.

```
$ conda install python-graphviz  
$ pip install graphviz # Anaconda에선 path를 별도로 설정해야 할 수 있다.
```

```
from sklearn.tree import export_graphviz  
import graphviz  
  
dot_data = export_graphviz(tree, out_file=None)  
graph = graphviz.Source(dot_data)  
graph.render("Titanic")
```

- Tree와 관련된 다른 visualization 방법들이 필요하면 `dtreeviz` 를 고려해 보자. [dtreeviz 설치 시 주의사항 \(<https://github.com/parrt/dtreeviz#install>\)](https://github.com/parrt/dtreeviz#install)

```
from dtreeviz.trees import dtreeviz  
viz = dtreeviz(clf, x, y)  
viz.save("Titanic.svg")
```

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - ▼ 2.1 Function approximation
 - 2.1.1 decision/tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic approach
 - ▼ 2.2 Training a tree model
 - 2.2.1 Decision region, decision boundary
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - ▼ 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1966)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

C:\Users\HK5\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please use get_feature_names_out instead.
warnings.warn(msg, category=FutureWarning)

```
[Text(0.5, 0.875, 'sex_female <= 0.5|gini = 0.478|namples = 783|nvalue = [474, 309]|nclass = died'),  
Text(0.25, 0.625, 'pclass_1 <= 0.5|gini = 0.329|namples = 496|nvalue = [393, 103]|nclass = died'),  
Text(0.125, 0.375, 'age <= 9.5|gini = 0.265|namples = 388|nvalue = [327, 61]|nclass = died'),  
Text(0.0625, 0.125, 'gini = 0.5|namples = 32|nvalue = [16, 16]|nclass = died'),  
Text(0.1875, 0.125, 'gini = 0.221|namples = 356|nvalue = [311, 45]|nclass = died'),  
Text(0.375, 0.375, 'age <= 14.0|gini = 0.475|namples = 108|nvalue = [66, 42]|nclass = died'),  
Text(0.3125, 0.125, 'gini = 0.0|namples = 4|nvalue = [0, 4]|nclass = survived'),  
Text(0.4375, 0.125, 'gini = 0.464|namples = 104|nvalue = [66, 38]|nclass = died'),  
Text(0.75, 0.625, 'fare <= 35.562|gini = 0.405|namples = 287|nvalue = [81, 206]|nclass = survived'),  
Text(0.625, 0.375, 'pclass_2 <= 0.5|gini = 0.477|namples = 194|nvalue = [76, 118]|nclass = survived'),  
Text(0.5625, 0.125, 'gini = 0.499|namples = 141|nvalue = [68, 73]|nclass = survived'),  
Text(0.6875, 0.125, 'gini = 0.256|namples = 8|nvalue = [8, 45]|nclass = survived'),  
Text(0.875, 0.375, 'parch <= 4.0|gini = 0.102|namples = 93|nvalue = [5, 88]|nclass = survived'),  
Text(0.8125, 0.125, 'gini = 0.064|namples = 91|nvalue = [3, 88]|nclass = survived'),  
Text(0.9375, 0.125, 'gini = 0.0|namples = 2|nvalue = [2, 0]|nclass = died')]
```



'Decision_Tree_of_Titanic_dataset.pdf'

- Logistic regression의 추정결과와 비교해보면 단순함에 비해 예측력이 상당히 뛰어나다.
- Decision tree는 sibsp 와 embarked에 대한 정보는 사용하지 않았다.

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - ▼ 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1983)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Alg
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
 - 5 Regression tree
 - ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
 - 7 Titanic dataset
 - ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
 - 9 Regression Tree
 - 10 분류모형의 비교

LogisticRegression - training dataset

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| died | 0.81 | 0.84 | 0.83 | 474 |
| survived | 0.74 | 0.70 | 0.72 | 309 |
| accuracy | | | 0.79 | 783 |
| macro avg | 0.78 | 0.77 | 0.77 | 783 |
| weighted avg | 0.78 | 0.79 | 0.78 | 783 |

LogisticRegression - test dataset

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| died | 0.84 | 0.88 | 0.86 | 333 |
| survived | 0.76 | 0.70 | 0.73 | 189 |
| accuracy | | | 0.81 | 522 |
| macro avg | 0.80 | 0.79 | 0.79 | 522 |
| weighted avg | 0.81 | 0.81 | 0.81 | 522 |

8.3 digit image dataset

- Decision tree 모형은 [multi-class classification \(<https://scikit-learn.org/stable/modules/tree.html#multi-output-problems>\)](https://scikit-learn.org/stable/modules/tree.html#multi-output-problems) 문제에서도 성능이 나쁘지 않다.
- 실제 decision tree를 정형화된 image 분석에 적용하기도 하며, MNIST dataset에서 test accuracy가 97%에 달하는 [tree decision 모형 \(<https://www.kaggle.com/carlolepealaars/97-on-mnist-with-a-single-decision-tree-t-sne>\)](https://www.kaggle.com/carlolepealaars/97-on-mnist-with-a-single-decision-tree-t-sne) 있다.

```
GridSearchCV(cv=12, estimator=DecisionTreeClassifier(random_state=42),
            param_grid={'max_depth': range(3, 25)}, scoring='f1_weighted')
```

Contents

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/data tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - ▼ 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1966)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
 - 5 Regression tree
 - ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
 - 7 Titanic dataset
 - ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
 - 9 Regression Tree
 - 10 분류모형의 비교

depth of the tree: 14
number of the leaves: 135

Best of Decision Trees – training dataset

Best of Decision Trees – training dataset

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 1.00 | 1.00 | 1.00 | 125 |
| 1 | 1.00 | 1.00 | 1.00 | 132 |
| 2 | 1.00 | 1.00 | 1.00 | 130 |
| 3 | 1.00 | 1.00 | 1.00 | 129 |
| 4 | 1.00 | 1.00 | 1.00 | 121 |
| 5 | 1.00 | 1.00 | 1.00 | 116 |
| 6 | 1.00 | 1.00 | 1.00 | 128 |
| 7 | 1.00 | 1.00 | 1.00 | 124 |
| 8 | 1.00 | 1.00 | 1.00 | 131 |
| 9 | 1.00 | 1.00 | 1.00 | 121 |
| accuracy | | | 1.00 | 1257 |
| macro avg | 1.00 | 1.00 | 1.00 | 1257 |
| weighted avg | 1.00 | 1.00 | 1.00 | 1257 |

Best of Decision Trees – test dataset

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.92 | 0.91 | 0.91 | 53 |
| 1 | 0.74 | 0.78 | 0.76 | 50 |
| 2 | 0.83 | 0.74 | 0.79 | 47 |
| 3 | 0.78 | 0.85 | 0.81 | 54 |
| 4 | 0.81 | 0.85 | 0.83 | 60 |
| 5 | 0.92 | 0.86 | 0.89 | 66 |
| 6 | 0.93 | 0.94 | 0.93 | 53 |
| 7 | 0.85 | 0.84 | 0.84 | 55 |
| 8 | 0.89 | 0.77 | 0.82 | 43 |
| 9 | 0.78 | 0.85 | 0.81 | 59 |
| accuracy | | | 0.84 | 540 |
| macro avg | 0.85 | 0.84 | 0.84 | 540 |
| weighted avg | 0.85 | 0.84 | 0.84 | 540 |

Contents ⚙

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - ▼ 2.1 Function approximation
 - 2.1.1 decision/tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - ▼ 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - ▼ 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1963)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Alg
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

L1 regularized Logistic Regression – training dataset

L1 regularized Logistic Regression – training dataset

| | precision | recall | f1-score | support |
|---|-----------|--------|----------|---------|
| 0 | 1.00 | 1.00 | 1.00 | 125 |
| 1 | 1.00 | 1.00 | 1.00 | 132 |
| 2 | 1.00 | 1.00 | 1.00 | 130 |
| 3 | 0.99 | 0.99 | 0.99 | 129 |
| 4 | 1.00 | 1.00 | 1.00 | 121 |
| 5 | 0.98 | 0.99 | 0.99 | 116 |
| 6 | 1.00 | 0.99 | 1.00 | 128 |
| 7 | 0.99 | 0.99 | 0.99 | 124 |
| 8 | 0.96 | 0.99 | 0.97 | 131 |
| 9 | 0.99 | 0.95 | 0.97 | 121 |

L1 regularized Logistic Regression – test dataset

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| accuracy | | | 0.99 | 1257 |
| macro avg | 0.99 | 0.99 | 0.99 | 1257 |
| weighted avg | 0.99 | 0.99 | 0.99 | 1257 |

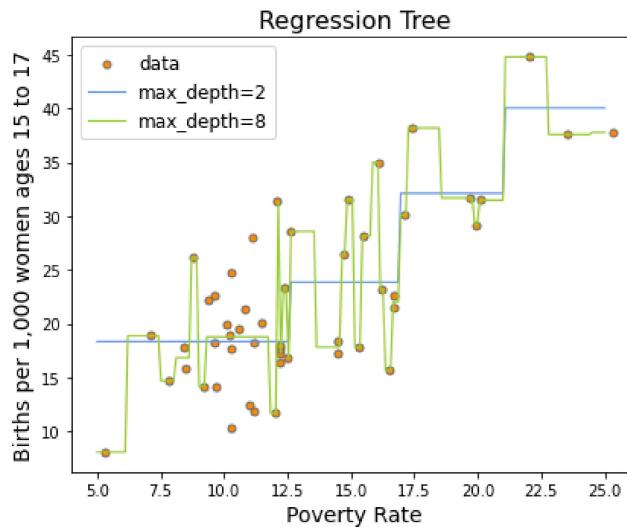
Contents ↗

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - 2.1 Function approximation
 - 2.1.1 decision/tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic approach
 - 2.2 Training a tree model
 - 2.2.1 Decision region, decision boundary
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross Quinlan, 1966)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Algorithm
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and regression trees)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeRegressor
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

9 Regression Tree

- 선형회귀모형은 하나의 관계식으로 전체표본공간 위에서 변수들 관계를 설명하므로 독립변수들 사이의 상관관계를 고려하기 어렵다.
- 비선형모형을 사용하면 독립변수가 많아지면서 모수의 수가 기하급수적으로 증가하므로 적합한 대안이 아니다.
- Regression tree는 feature space에 partition을 반복적으로 적용하여 각 leaf에 포함된 관찰값으로 예측한다.
- 훈련과 예측이 간단하다.
- Missing value가 포함된 자료를 이용할 수 있다.
- Regression surface의 형태에 구애받지 않는다.
- Feature의 수가 크지 않다면 각 feature의 영향력을 비교할 수 있다.
- Loss로 MSE나 MAE등을 사용하는 것을 제외하면 classification과 동일한 방법으로 tree를 훈련한다.

```
sklearn.tree.DecisionTreeRegressor(*, criterion='squared_error', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, ccp_alpha=0.0)
```



Contents ⚙

- 1 Function approximation
- ▼ 2 Decision Tree Model
 - ▼ 2.1 Function approximation
 - 2.1.1 decision/tree의 구조
 - 2.1.2 Feature and target spaces
 - 2.1.3 Top-down greedy heuristic appl
 - ▼ 2.2 Training a tree model
 - 2.2.1 Decision region, decision boun
 - 2.2.2 Tree 설계
 - 2.2.3 Overfitting
 - 2.2.4 Feature importances
- ▼ 3 분기의 결정
 - 3.1 예제: Decision Tree for PlayTennis
 - ▼ 3.2 Information gain
 - 3.2.1 Information gain
 - 3.3 Gini impurity (Gini Index)
 - 3.4 Classification Error
 - 3.5 Binary decision trees에서 분할기준
- ▼ 4 Algorithms
 - 4.1 ID3(Iterative Dichotomiser 3, Ross)
 - ▼ 4.2 C4.5 (Ross Quinlan, 1993)
 - 4.2.1 gain ratio
 - 4.2.2 Continuous attributes
 - 4.2.3 Missing value
 - 4.2.4 Attribute weights
 - ▼ 4.2.5 Error-Based (post) Pruning Alg
 - 4.2.5.1 Reduced-Error Pruning
 - 4.2.5.2 statistical pruning of C4.5
 - 4.2.5.3 Rule post pruning
 - 4.3 CART, C&RT (classification and reg)
- 5 Regression tree
- ▼ 6 Sklearn Implementation
 - 6.1 DecisionTreeClassifier, CART
 - ▼ 6.2 DecisionTreeClassifier
 - 6.2.1 Pre-pruning
 - 6.2.2 Post-pruning: Cost-Complexity
 - 6.2.3 Categorical variable
- 7 Titanic dataset
- ▼ 8 추정 모형에 대한 세부정보
 - 8.1 visualization
 - 8.2 save as pdf
 - 8.3 digit image dataset
- 9 Regression Tree
- 10 분류모형의 비교

10 분류모형의 비교

| | Linear Probability model | Logistic Regression model | Naïve Bayes | SVM | Decision Tree model |
|---------------------------------|---|----------------------------|--------------------------|--|--------------------------------------|
| Decision boundary의 형태 | linear | linear | linear | linear in non-linear space | non-linear |
| cost of computation, 모형의 유지와 보수 | low | high | extremely low | very high | high |
| class의 비율이 추정 결과에 미치는 영향 | sensitive | less | less | less | less |
| 추정 결과의 해석의 용이성 | easiest | easy | moderate | not possible unless linear | difficult |
| 추정에 필요한 표본의 수 | balanced | relatively smaller | relatively smaller | relatively smaller | large sample |
| 특징 | linear approximation of Logistic regression | log odds | conditional independence | highly nonlinear not efficient for overlaped sample | non-parametric robust to outliers |
| 유형 | probabilistic discriminant | probabilistic discriminant | generative | discriminant | discriminant |