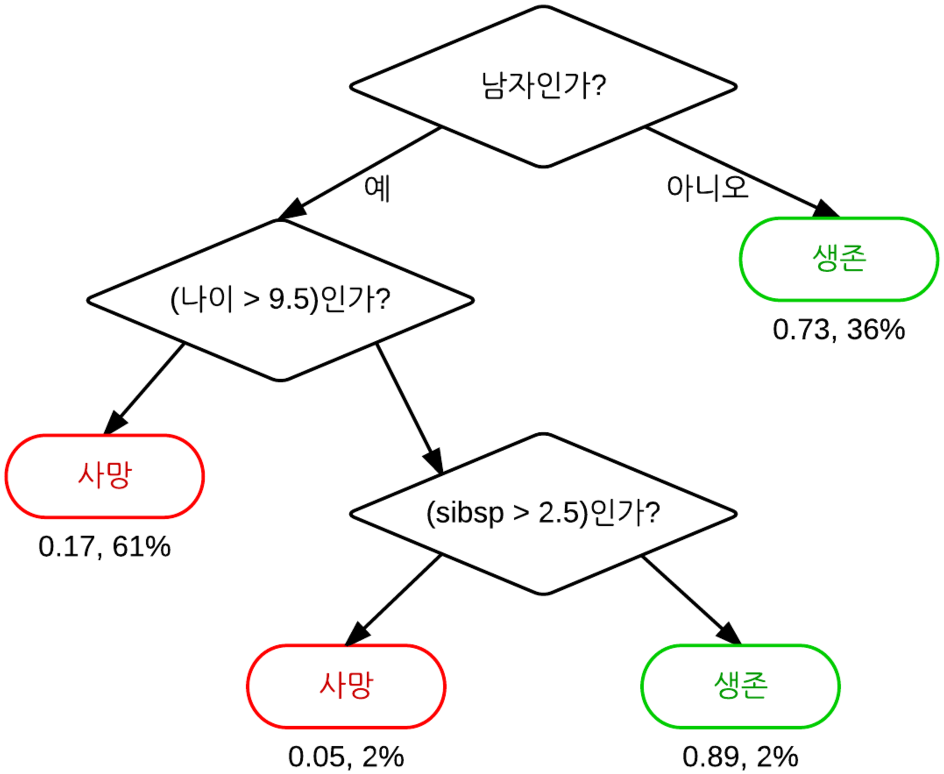
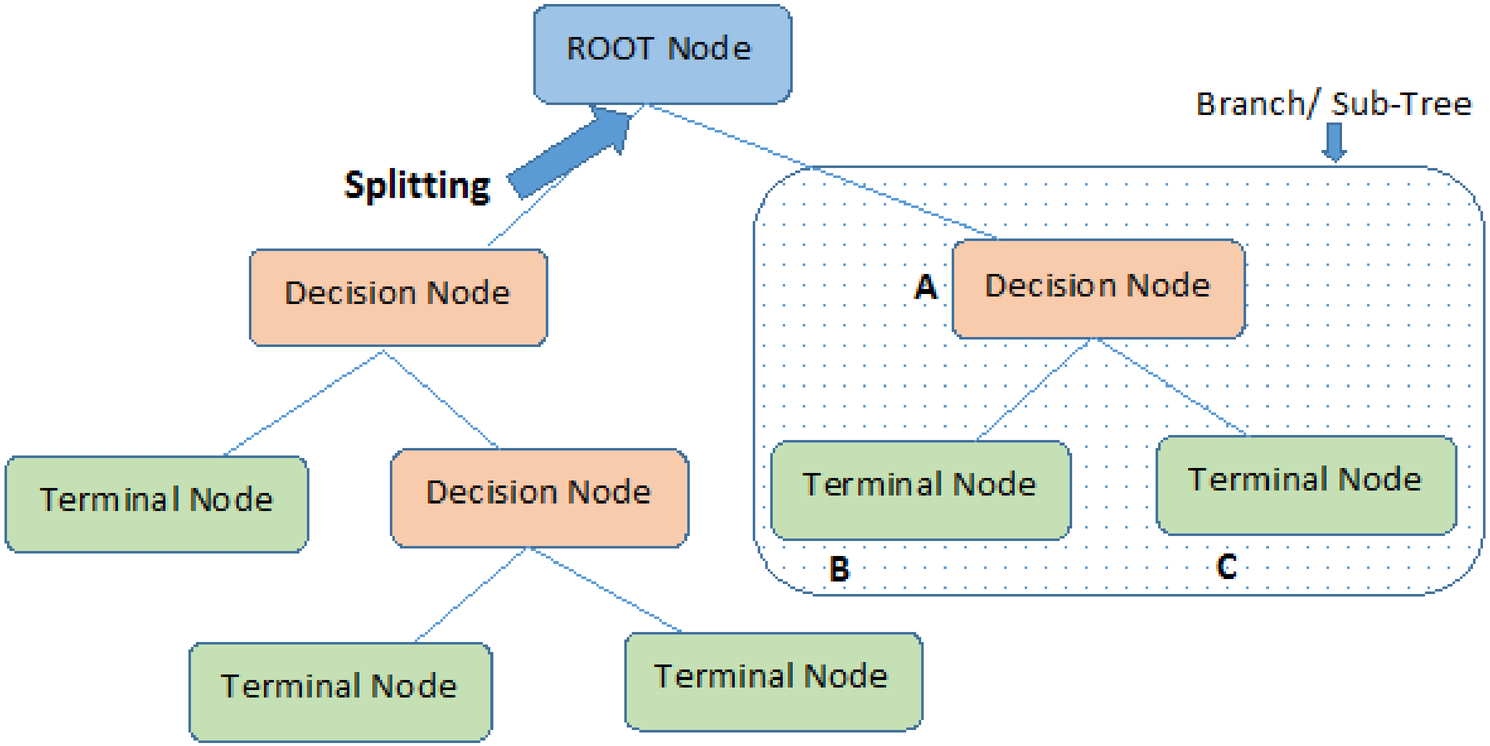
Decision Tree

Tree?



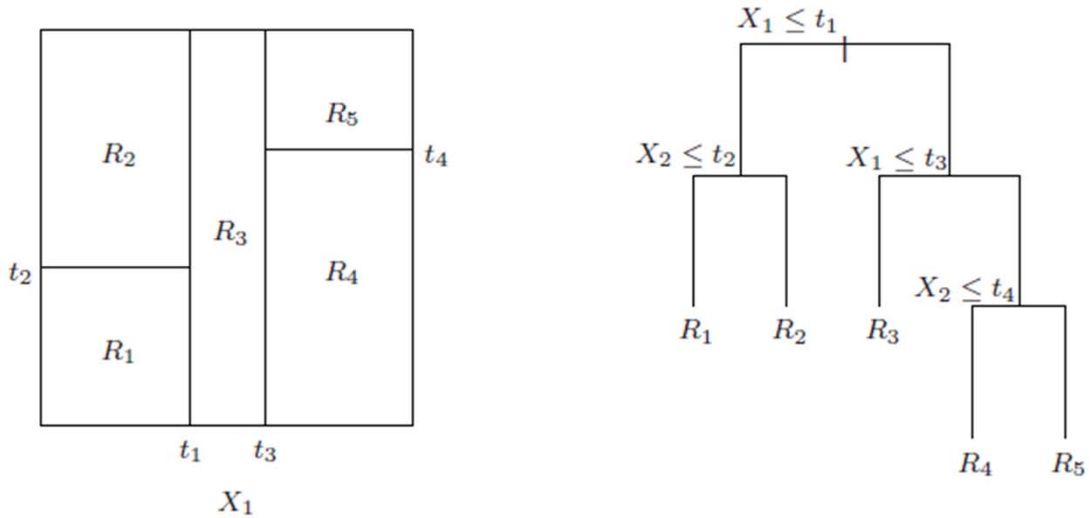
- 질문을 던지며 대상을 좁히는 스무고개와 유사
- Y/N 질문을 이어나가며 학습
- Decision Tree를 학습한다는 것은 정답에 가장 빨리 도달하는 Y/N 질문 목록을 학습한다는 것
- 데이터를 가장 잘 구분할 수 있는 Decision Tree를 생성

Terminology



Note:- A is parent node of B and C.

Background



$$\hat{f}(X) = \sum_{m=1}^{5} c_m \, I\{(X_1, X_2) \in R_m\}$$

Growing a Decision Tree

- CART (Classification and Regression Tree), C4.5,
CHAID(Chai-squared Automatic Interaction Detection)…

- 어떻게 해야 가장 잘 분류할 수 있는가? = 어떤 Attribute로 결정되어야 하는가?

- “어떤 Attribute가 더 확실한 정보를 제공하고 있는가?” 로 분기 Attribute를 선택

- 분기 기준 (Split Criterion)

-Categorical Response : Gini Index, Entropy , χ^2-$statistics$
-Numerical Response : RSS

CART Algorithm

- 한번에 하나의 변수를 사용
- Binary decision Rule
- Numerical attribute : test whether value is greater or less than constant.
- Nominal attribute : test whether value belongs to subset.
- 불순도(impurity) 를 사용한 Greedy search

Classification and Regression in CART

- 범주나 연속형 수치 모두 예측

- Classification
  - 새로운 데이터가 특정 terminal node에 속한다는 정보를 확인한 뒤 할당 된 terminal node의 major category의 확률을 반환

- Regression
  - terminal node의 종속변수(y)의 평균을 예측값으로 반환
  - 예측값의 종류는 terminal node 개수와 일치

Regression in CART~~(볼사람만 보세요...)~~

- Suppose first that we have a partition into M regions $R_1, R_2, \ldots, R_M$, and we model the response as a constant c_m in each region

$$f(X) = \sum_{m=1}^{M} c_m I\{ x \in R_m\}$$

- Adopt minimization SSR($= \sum (y_i - f(x_i))^2$) as criterion

- Best $\hat{c}_m$ is just the average of y_i in region R_m

$$\hat{c}_m = ave(y_i | x_i \in R_m)$$

- In general, computationally infeasible to find the best binary partition in terms of minimum sum of squares. So…greedy algoritm

Regression in CART~~(볼사람만 보세요...)~~

- Consider a splitting variable j, s and define the pair of half-planes

$$R_1\,(j,s) = \{X|X_j \le s\} \text{ and } R_2\,(j,s) = \{X|X_j > s\}$$

- Seek the splitting variable j and s that solve

$$\min_{j,s}\,[\min_{c1} \sum_{x_i \in R_1\,(j,s)} (y_i - c_1)^2 + \min_{c2} \sum_{x_i \in R_2\,(j,s)} (y_i - c_2)^2]$$

- Inner minimization is solved by

$$\hat{c}_1 = ave(y_i|x_i \in R_1\,(j,s)) \text{ and } \hat{c}_2 = ave(y_i|x_i \in R_2\,(j,s))$$

- Now, determination of the best pair (j,s) is feasible. Repeat(Resursive partitioning) this to the end...

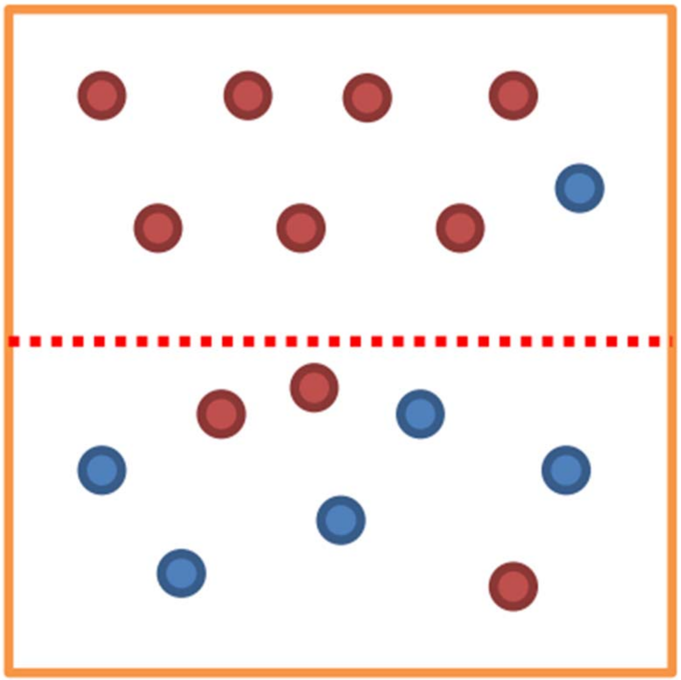
CART’s split idea

- Split Criterion(분기 기준)
  - Categorical Response : Gini Index, entropy
  - Numerical Response : RSS

- Parent Node의 criterion과 Child Node의 criterion 비교

- 가능한 split들 중에서 impurity의 개선 수준이 높은 attribute를 선택

Classification in CART



- Split Criterion으로 Gini Index, Entropy, Misclassification error
- The proportion of class k observations in node m, N_m obs, in region R_m

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

Misclassification Error

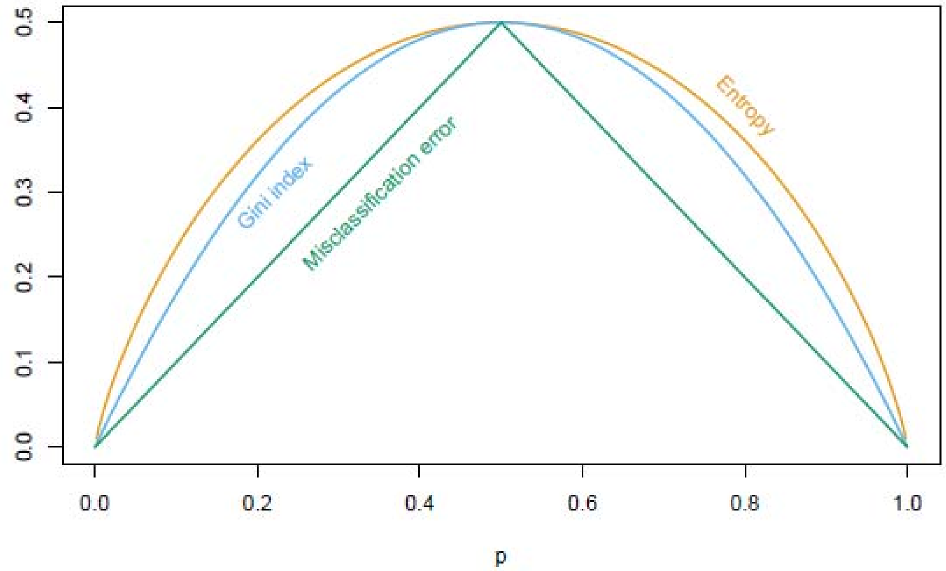
$$\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)}$$

Gini index

$$\sum_{k \neq k\prime} \hat{p}_{mk}\, \hat{p}_{mk\prime} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk\prime})$$

Cross-entropy or deviance

$$-\sum_{k=1}^{K} \hat{p}_{mk} log \hat{p}_{mk}$$



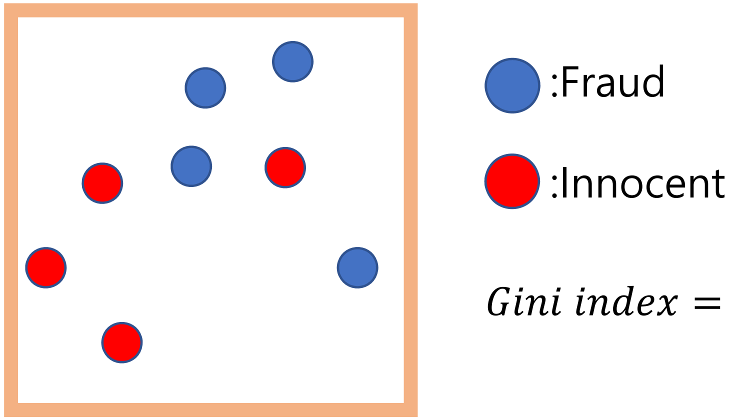
Node impurity measures for two-class classification, as a function of the proportion p in class 2. Cross-entropy has been scaled to pass through $(0.5, 0.5)$.
All three are similar, but cross-entropy and the Gini index are differentiable, and hence more amenable to numerical optimization.

Gini Index(Impurity)

- Gini 값을 가장 작게 해주는 attribute를 선택

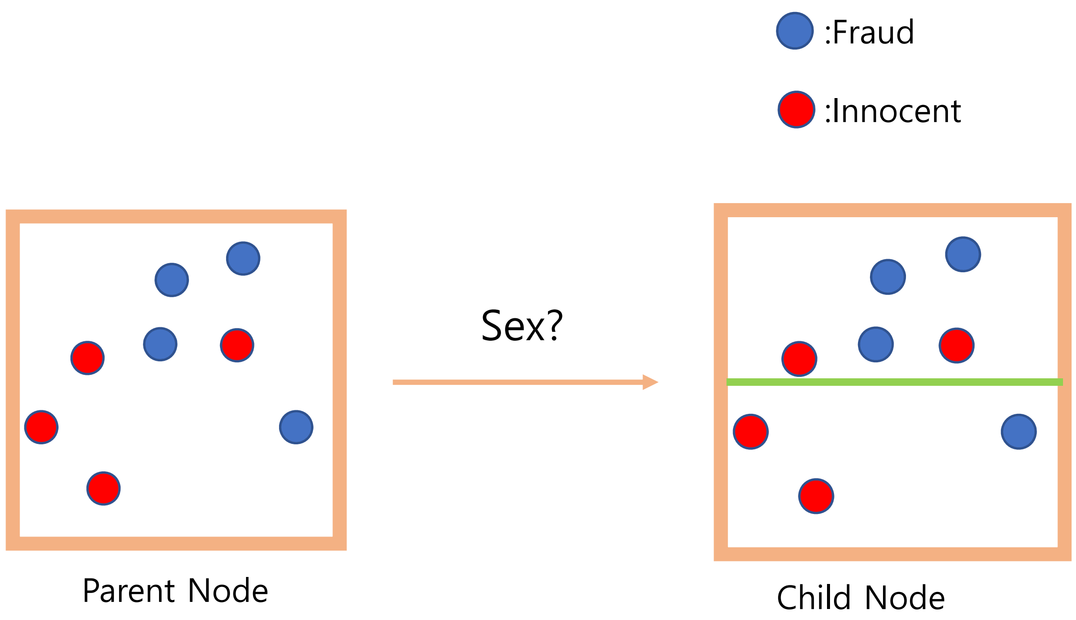
$$Gini\ index = \sum_{i=1}^{m} p_i\,(1 - p_i) = 1 - \sum_{i=1}^{m} {p_i}^2$$

Ex)



Parent Node

$$Gini\ index = 1 - \left(\frac{4}{8}\right)^2 - \left(\frac{4}{8}\right)^2 = 0.5$$

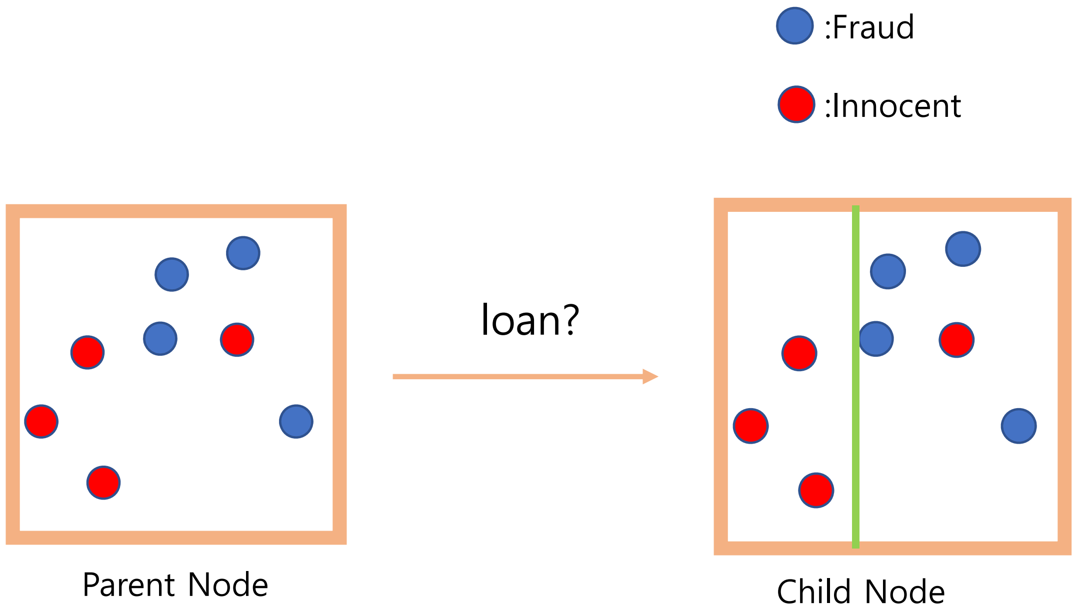


$$Gini\ index = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0.48$$

$$Gini\ index = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.44$$

$$weighted\ average = 0.48 \times \frac{5}{8} + 0.44 \times \frac{3}{8} = 0.465$$

$$Goodness\ of\ split = 0.5 - 0.465 = 0.035$$

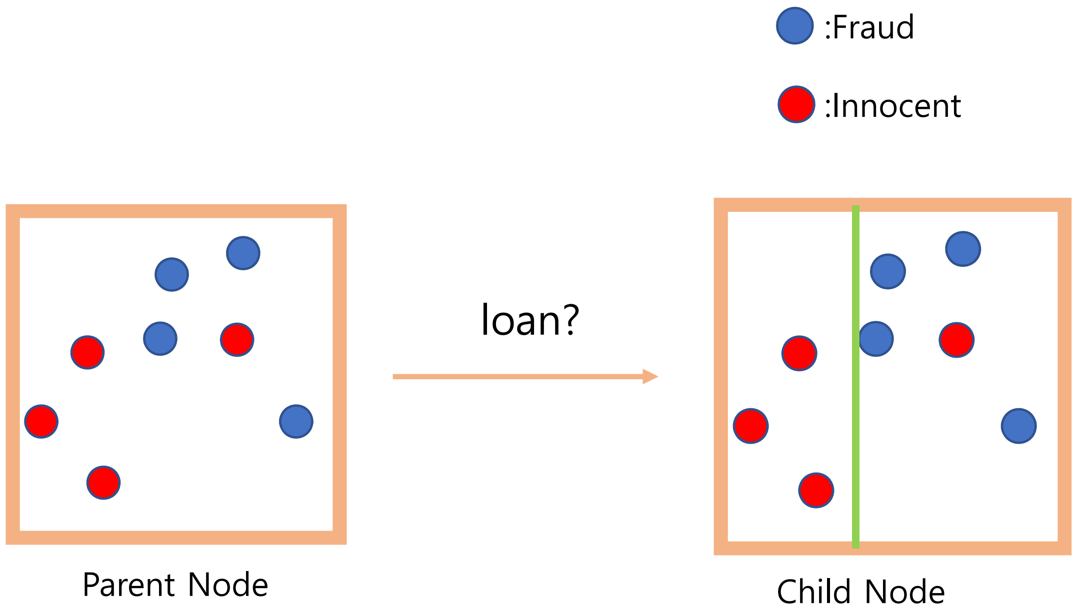


$$Gini\ index = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2 = 0.32$$

$$Gini\ index = 1 - \left(\frac{3}{3}\right)^2 = 0$$

$$weighted\ average = 0.32 \times \frac{5}{8} + 0 \times \frac{3}{8} = 0.2$$

$$Goodness\ of\ split = 0.5 - 0.2 = 0.3$$



$$Gini\ index = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2 = 0.32$$

$$Gini\ index = 1 - \left(\frac{3}{3}\right)^2 = 0$$

$$weighted\ average = 0.32 \times \frac{5}{8} + 0 \times \frac{3}{8} = 0.2$$

$$Goodness\ of\ split = 0.5 - 0.2 = 0.3$$

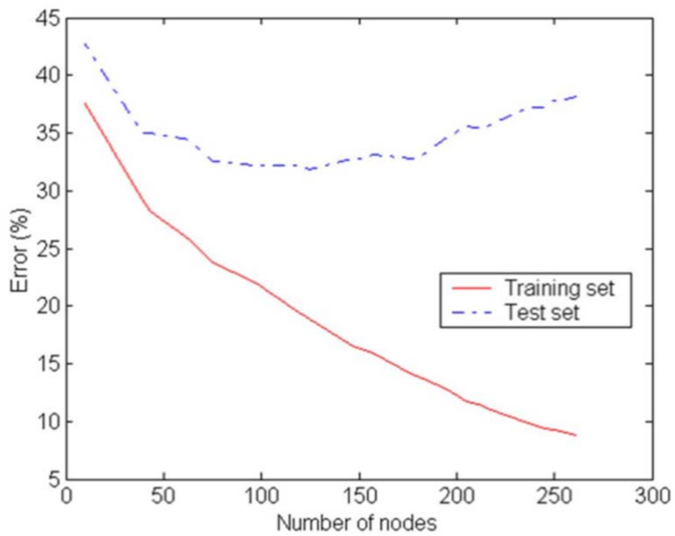
Greedy Search

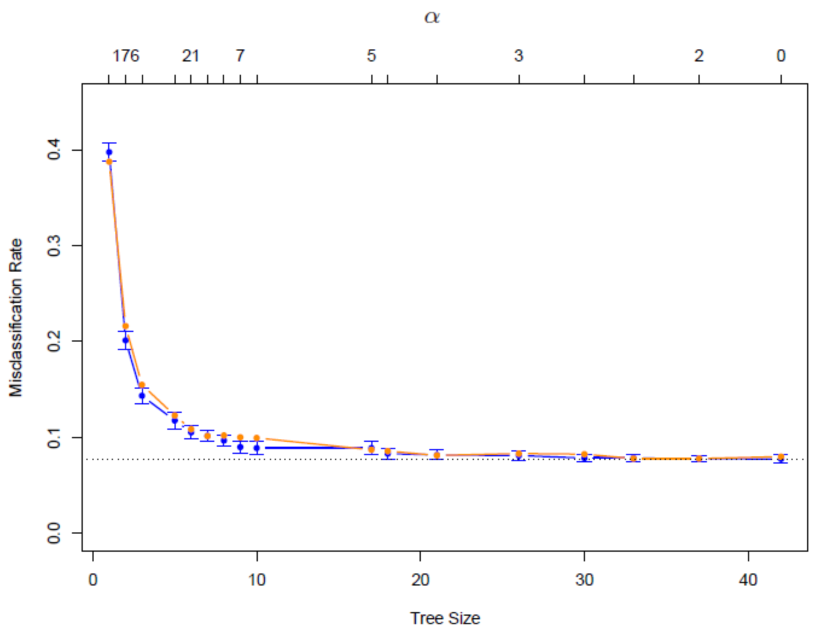
- 가능한 모든 경우의 수를 따져보아 최선의 방법을 찾아냄
- 연속형 변수 : unique한 관찰값 m개 → m-1회의 greedy search
- 범주형 변수 : q개의 범주 →$2^{q-1} - 1$
- 불순도 개선이 가장 높은 것을 선택

Why Binary decision rule

- 2개 이상의 그룹으로 분리할 수도 있지 않을까?
- 단순하고, 설명하기 쉬움
- Multiway split을 하면, 다음 단계에 insufficient data가 남을 수도 있음
- Multiway split은 a series of binary splits을 통해 얻을 수 있음

Overfitting

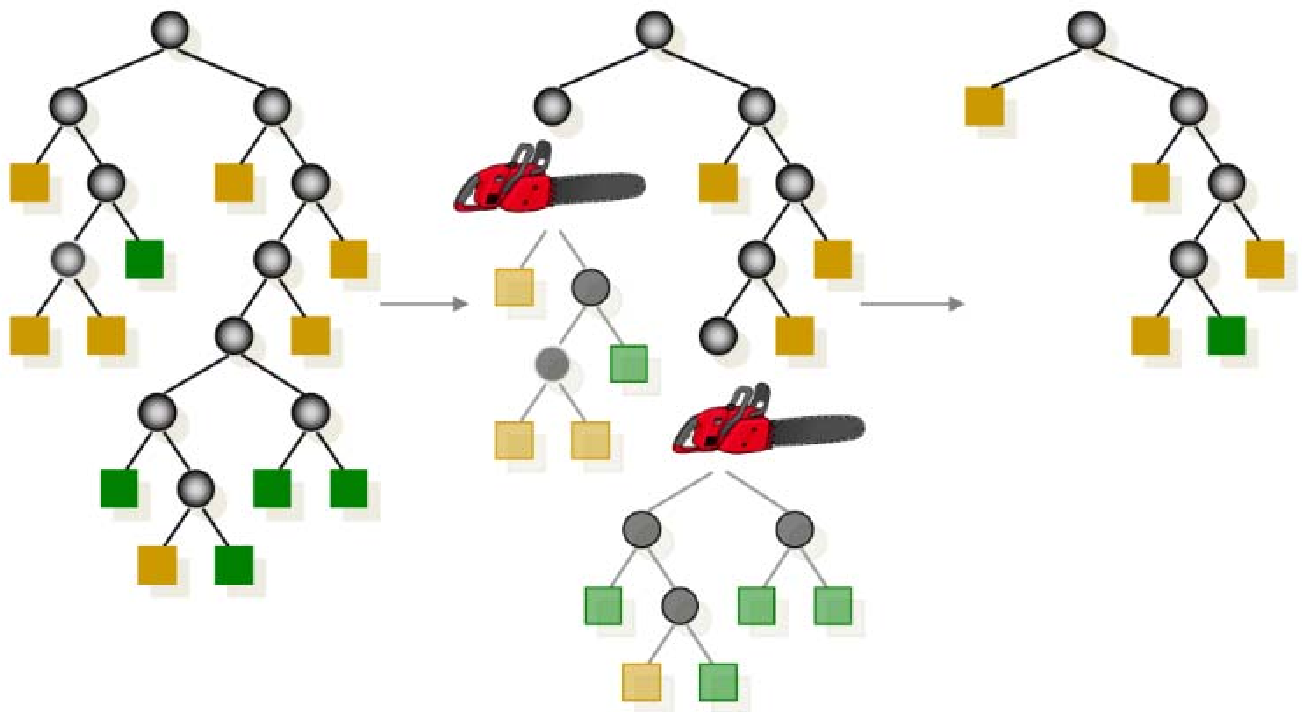




- Training data 오분류는 감소
- test data 오분류는 감소하다가 증가

- Tree size에 따른 오분류율

Pruning(가지치기)



Pruning(가지치기)

- 사전 가지치기(pre-pruning)
  - 트리 성장 작업 중 실시
  - 어느 정도 불순한 Leaf node가 있어도 성장 조기 종료
  - 트리의 최대 깊이(max_depth), Leaf node의 최대 개수를 제한
  - Node 분할을 위한 최소 포인트 개수를 지정

- 사후 가지치기(post-pruning)
  - 트리 성장 작업 완료 후 실시
  - 완성된 트리에서 일부 Leaf node를 제거

Cost Complexity Pruning

- 비용함수를 최소화 시키는 split을 찾아내어 pruning을 결정
- Cost complexity function

$$CC(T) = Err(T) + \ \alpha * L(T)$$

  - $CC(T)$: Decision tree의 cost complexity (오류가 적으면서 terminal node의 수가 작은 것이 작은 값을 가진다)
  - $Err(T)$: test data에 대한 오분류율
  - $L(T)$: Terminal node의 수
  - α : 가중치 (0.01~0.1)

- 새로운 split을 함으로써 생기는 error 감소 이득이 penalty 증가보다 크지 못하면 더 이상 split을 하지 않음

http://mlwiki.org/index.php/Cost-Complexity_Pruning#Choosing_.24.5Calpha.24

CHAID(Chai-squared Automatic Interaction Detection)

- 이산형 목표변수일 때, 카이제곱 검정
- 연속형 목표변수일 때, F-검정
- multiway split을 수행
- 카이제곱 통계량 : 관측도(frequency)로 이루어진 분할표(contingency table)로 계산

	범주1	범주2	…	범주c	계
범주1	f_{11}	f_{12}	…	f_{1c}	f_{1+}
범주2	f_{21}	f_{22}	…	f_{2c}	f_{2+}
…	…	…	…	…	…
범주r	f_{r1}	f_{r2}	…	f_{rc}	f_{r+}
계	f_{+1}	f_{+2}	…	f_{+c}	f_{++}

CHAID(Chai-squared Automatic Interaction Detection)

- Pearson 카이제곱 통계량

$$\chi^2 = \sum_{i,j} \frac{(f_{ij} - e_{ij})^2}{e_{ij}}$$

- 통계량 자유도는 (r-1)(c-1)

- e_{ij}분포의 동일성 또는 독립성 가설 하에서 계산된 기대도수 $e_{ij} = \frac{f_{i+} * f_{+j}}{f_{++}}$

CHAID(Chai-squared Automatic Interaction Detection)

- 카이제곱 통계량이 자유도에 비해 매우 작다 → 예측변수의 각 범주에 따른 목표변수. 분포가 서로 동일, 따라서 예측변수가 목표변수 분류에 영향을 주지 않음

- 자유도에 비해 카이제곱 통계량이 크고 작음은 p-value로 표현 가능, 카이제곱 통계량이 자유도에 비해 작으면 p-value ↑

- Split criterion을 카이제곱 통계량으로 하는 것 = p-value값이 가장 작은 예측변수와 그 때의 최적 split에 의해서 child node를 형성시키는 것

Decision Tree의 특징

- 간단하고 직관적인 결과 표현이 가능, 인간의 의사결정 과정과도 유사
- Numerical, Categorical variable 모두를 다룰 수 있음
- 질적 변수를 더미 변수를 생성하지 않고 쉽게 다룰 수 있음
- Attribute의 scaling이 필요 없음
- 관측치의 절대값이 아닌 순서가 중요 → Outlier에 Robust

Decision Tree의 특징

- Missing values 쉽게 다룰 수 있음

1) Categorical predictor : “missing " 이란 make a new category

2) Surrogate split : 대체 변수(surrogate variables) 사용하여 split 진행

- Greedy 알고리즘 → 부분 최적화, Global optimization 아닌 local optimization

Decision Tree의 특징

- Pruning을 해도 overfitting을 완벽하게 해결하지 못함
- 다른 방법들에 비해 예측 정확도가 떨어짐

Instability of Trees

- 데이터가 조금만 바뀌어도 split이 상당히 많이 달라질 수 있음 (instability)
- Tree는 high variance 이고 이는 tree 방법의 major problem
- Major reason for instability is the hierarchical nature of the process
- 상위 split에서의 effect of an error가 그 아래 모든 split으로 전파됨

Improvement

- **tree'들'을 모아 예측 정확도를 높일 수 있을까?**
  - Ensemble Method
    - 여러가지 Algorithm을 모아 성능을 향상
  - Bias: 실제 값과 예측 값 사이의 차이
  - Variance: 다른 데이터셋에 모델이 적용될 때의 차이
  - Deep하게 학습된 tree 모델은 Low Bias, High Variance
  - 여러 개의 모형을 만들고, 평균을 구함으로 Variance를 낮춤..
  - 자세한건 목요일에

Regression Model performance evaluation

- RMSE(Root Mean Squared Error)

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

◦ scale-dependent error

- MAPE(Mean Absolute Percentage Error)

$$\frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

◦ A_t: 실제값, F_t: 예측값
◦ A_t<1 일때 문제발생

Regression Model performance evaluation

- MASE (Mean Absolute Percentage Error)

$$\frac{1}{T}\sum_{t=1}^{T}\left(\frac{|e_t|}{\frac{1}{T-1}\sum_{t=2}^{T}|Y_t - T_{t-1}|}\right) = \frac{\sum_{t=1}^{T}|e_t|}{\frac{T}{T-1}\sum_{t=2}^{T}|Y_t - T_{t-1}|}$$

  - 변동폭에 비해 얼마나 오차가 나는지를 측정

Classification Model performance evaluation

		Predicted		
		Positive (Innocent)	Negative (Fraud)	
Observed	Positive (Innocent)	TP 700	FN 150	P 850
	Negative (Fraud)	FP 50	TN 100	N 150

- Precision

$$\frac{TP}{TP+FP} : \frac{700}{750}, \qquad \frac{TN}{TN+FN} : \frac{100}{250}$$

* 모델이 Positive로 분류한 것 중 실제 Positive인 비율.

- Recall(Sensitivity)

$$\frac{TP}{TP+FN} : \frac{700}{850}, \qquad \frac{TN}{TN+FP} : \frac{100}{150}$$

* 실제 Positive인 것 중 모델이 True라고 예측한 것의 비율

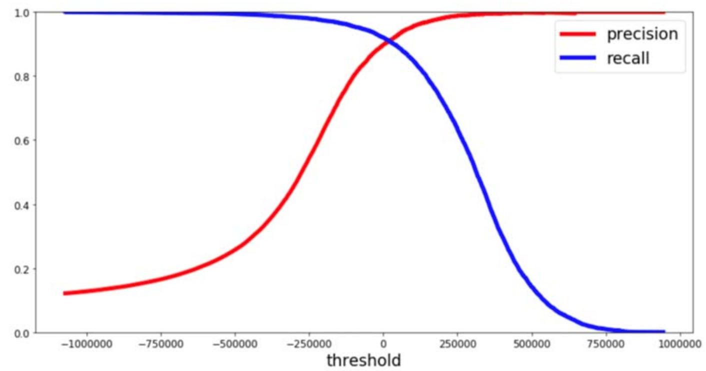
- Accuracy

$$\frac{TP+TN}{P+N\ (\text{전체 데이터})} : \frac{800}{1000}$$

- F-score :

$$2 * \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 * \frac{Precision\ * Recall}{Precision + Recall} : 0.5$$

* precision과 recall의 조화평균



Classification Model performance evaluation

- Recall(Sensitivity)

$$\frac{TP}{TP + FN}$$

* 실제 Positive인 것 중 모델이 True라고 예측한 것의 비율

- Fall-out(False Positive Rate, FPR)

$$\frac{FP}{TN + FP}$$

* 실제 False인 것 중 모델이 True라고 예측한 것의 비율

- ROC (Receiver Operating Characteristic Curve)

- AUC(Acrea Under Curve)