CSCI4622 Machine Learning

2020 Spring

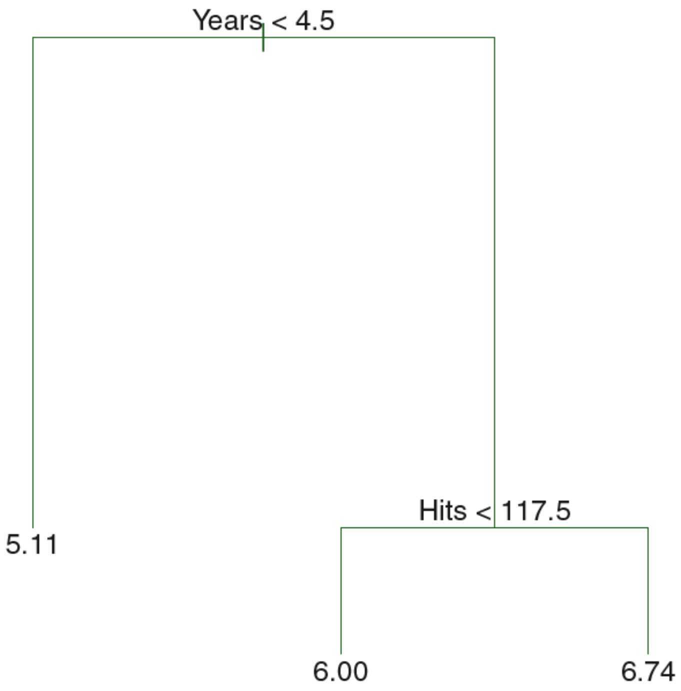
Tree Methods-continued

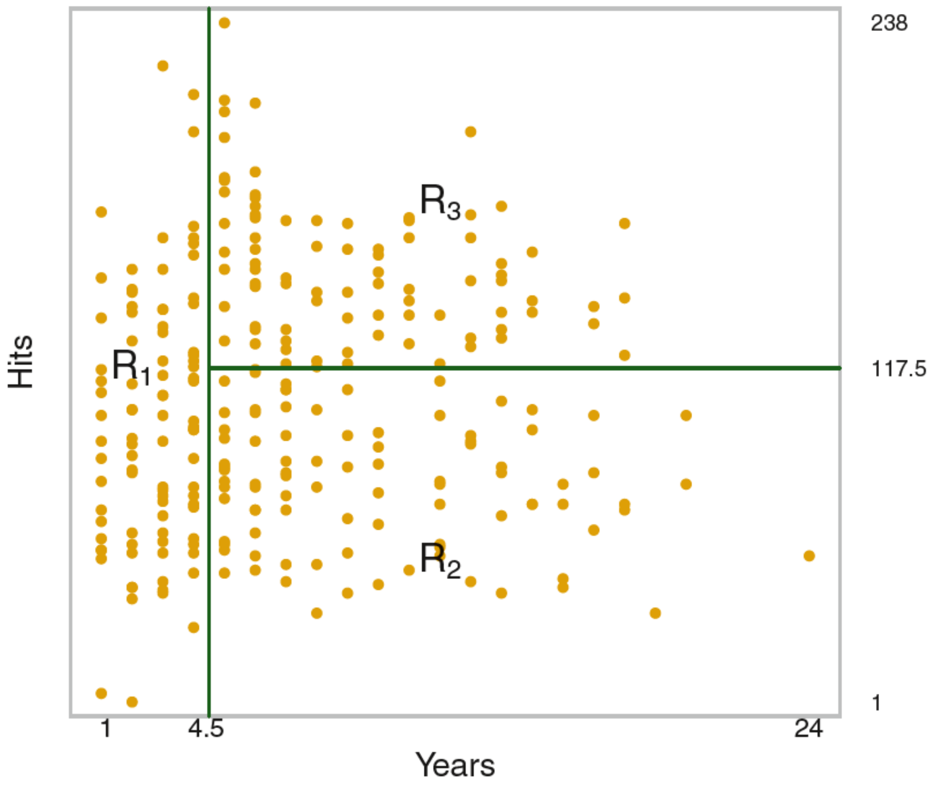
Geena Kim

*Slide contents adopted from ISLR material

Decision Tree

Split Rule: Minimize the metric (MSE, entropy, etc) of the boxes





Decision Tree Split Criteria

Regression Tree

MSE

$$H(X_m) = \frac{1}{N_m} \sum_{i \in N_m} (y_i - \bar{y}_m)^2$$

MAE

$$H(X_m) = \frac{1}{N_m} \sum_{i \in N_m} |y_i - \bar{y}_m|$$

Classification Tree

Gini

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk})$$

Entropy

$$H(X_m) = -\sum_k p_{mk} \log(p_{mk})$$

Information Gain

$$IG = E_{parent} - \frac{N_L}{N} E_L - \frac{N_R}{N} E_R$$

Hyperparameter search

Grid Search Tip

- Give a range of values for each hyperparameter
- Measure a training time for one, then estimate how long for the loop
- Adjust number of values, range, or hyperparameters to include

max_depth

min_samples_split

min_samples_leaf

max_features

min_impurity_decrease

Decision Tree Pros and Cons

Trees are easy to understand

Trees don't suffer collinearity

Trees are good for non-linear features

Trees handle categorical variables easily

Trees are weak-learner

Trees have high variance in general

Trees can overfit easily

Linear regression is a better choice if features are linear

Tree's performance can be greatly improved when ensembled

Decision Tree – Pruning

Issue: Sometimes a good split can happen later

Idea: Grow the tree fully then prune

How: Minimal Cost-Complexity Pruning

(New) Pruning option is available (new) in sklearn 0.22

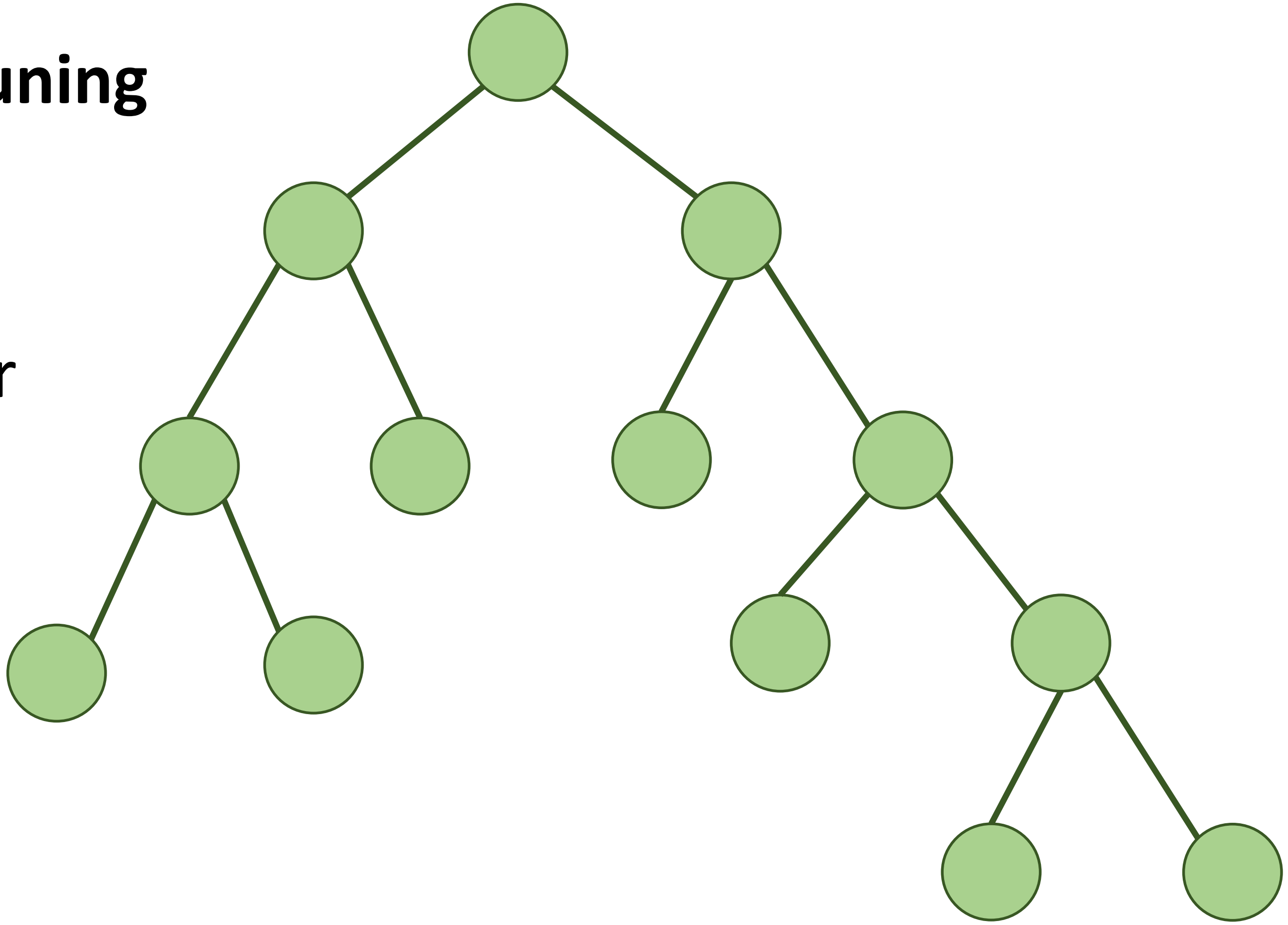
Decision Tree – Pruning

Minimal Cost-Complexity Pruning

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

α : complexity parameter

Full Tree T_0



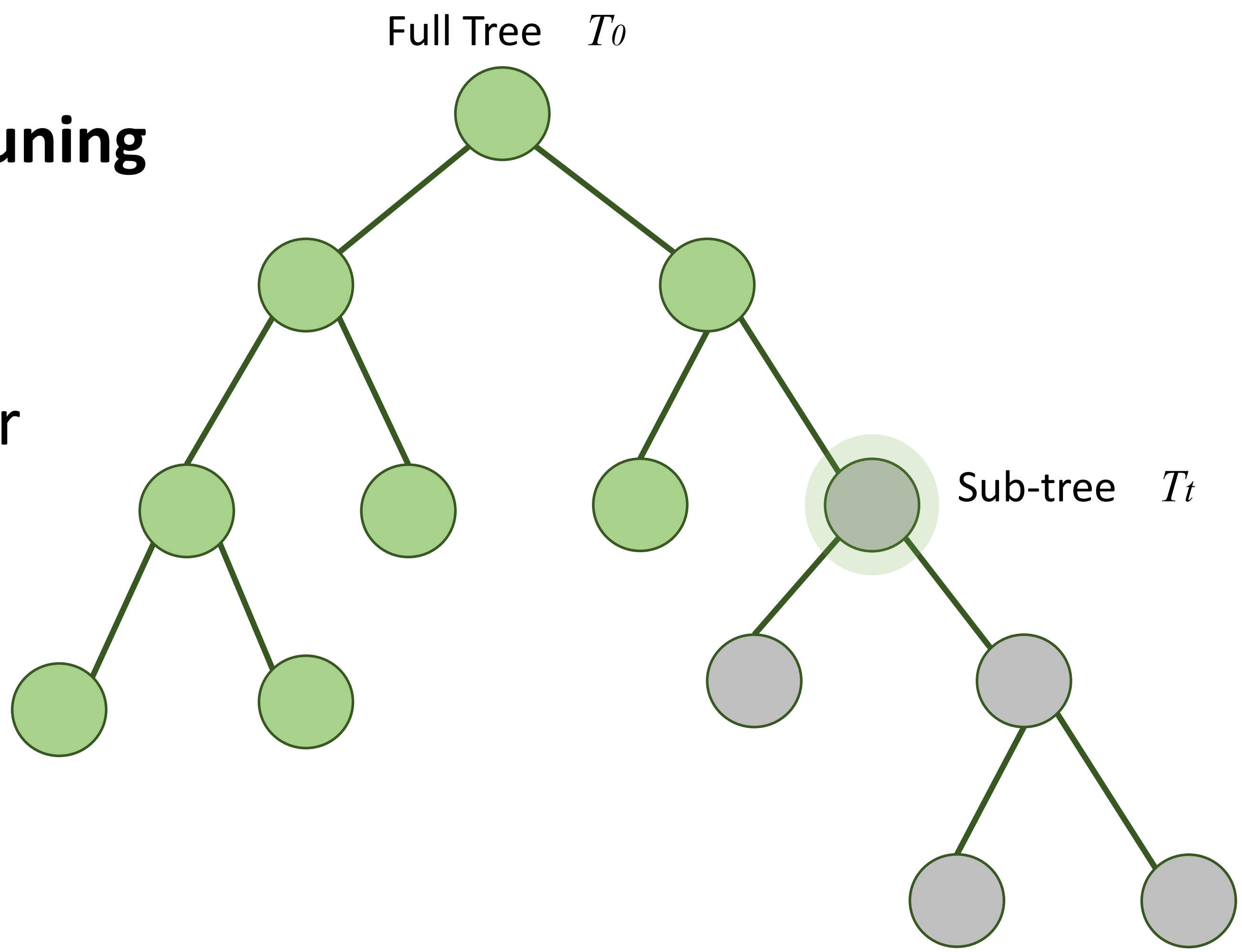
https://scikit-learn.org/stable/modules/tree.html#minimal-cost-complexity-pruning
https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html

Decision Tree – Pruning

Minimal Cost-Complexity Pruning

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

α : complexity parameter



Decision Tree – Pruning

Minimal Cost-Complexity Pruning

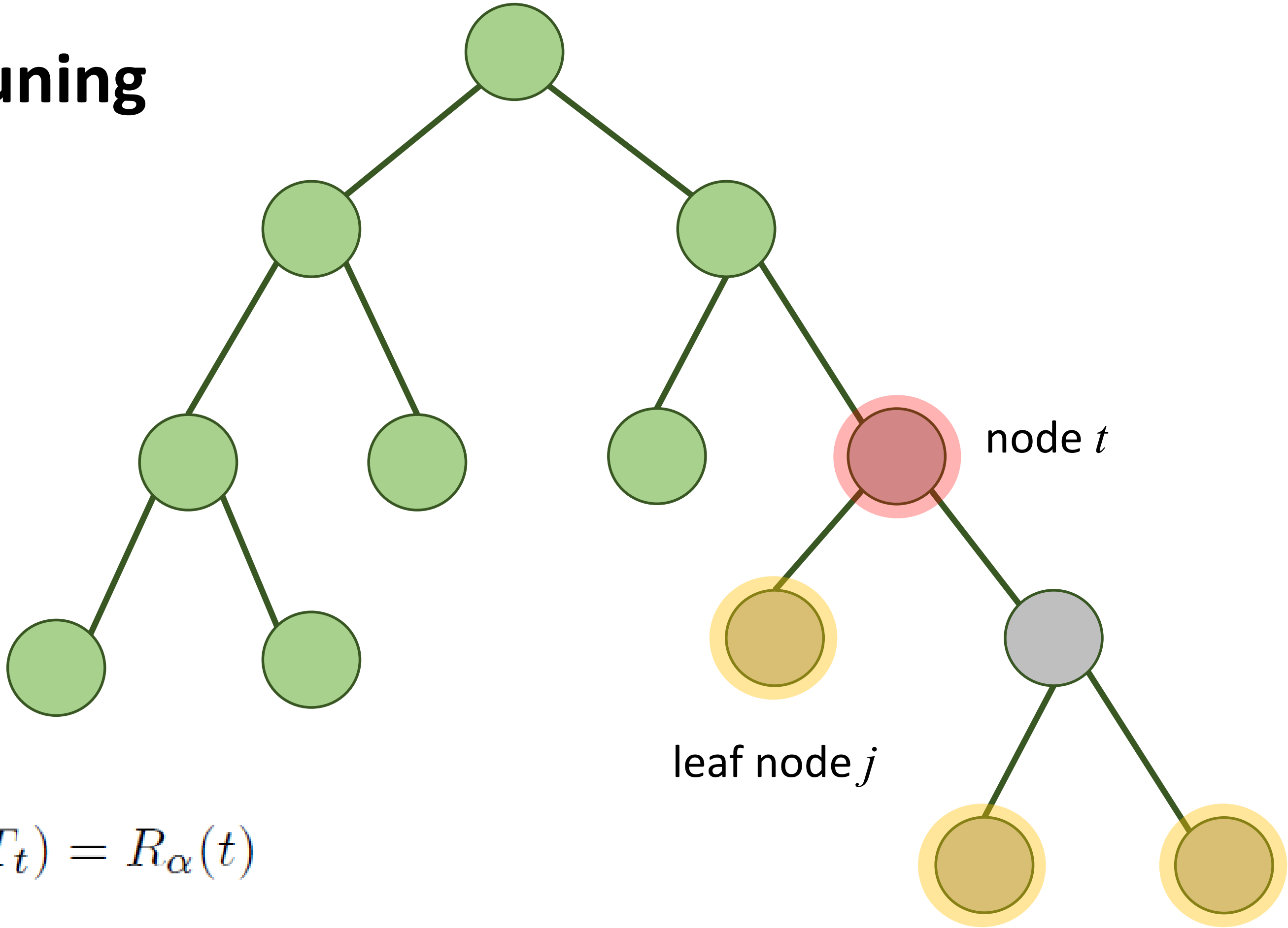
$$R_\alpha(T) = R(T) + \alpha|T|$$

α : complexity parameter
|T|: number of leaf nodes of the subtree

Impurity at the node t

$$R(T_t) < R(t)$$

Sum of the impurities
at the leaf nodes of the subtree T_t

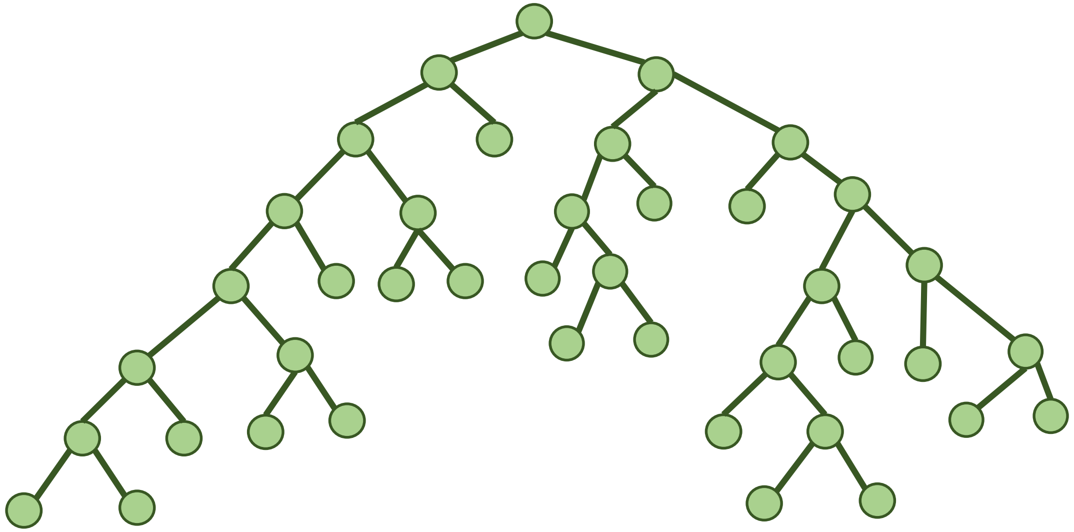


$$R_\alpha(T_t) = R_\alpha(t)$$

$$\alpha_{eff}(t) = \frac{R(t) - R(T_t)}{|T| - 1}$$

Decision Tree – Pruning

Minimal Cost-Complexity Pruning



Iteratively removes the weakest link

When does it stop pruning?

Stop when $\min(\alpha_{eff}) > \alpha_{ccp}$

α_{ccp} : cost complexity parameter, “`ccp_alpha`”

Decision Tree – Pruning

The cost complexity parameter(`ccp_alpha`) is a hyperparameter

How do we determine the right cost complexity parameter?

-> Use validation dataset (or cross-validation)