Decision Trees

He He

CDS, NYU

April 6, 2021

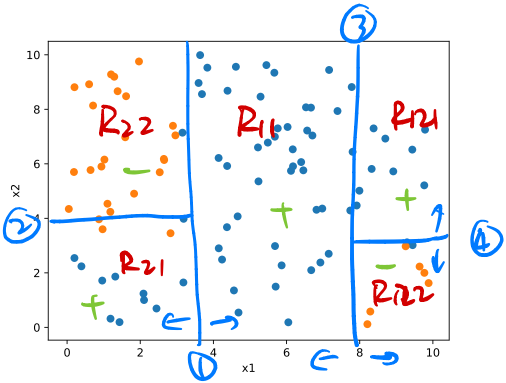
Today's lecture

- Our first inherently non-linear classifier: decision trees.
- Ensemble methods: bagging and boosting.

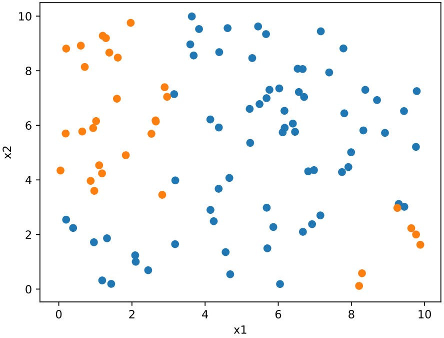
Decision Trees

Motivating example in 2d

- Partition data into different (axis-aligned) regions recursively

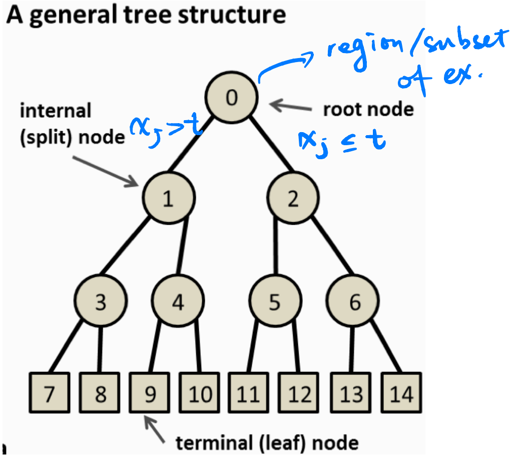


Classification flowchart



Is this a linear or non-linear classifier?

Decision trees setup



We'll only consider

- *binary* trees (vs multiway trees where nodes can have more than 2 children)
- each node contains a subset of data points
- decisions at each node involve only a *single* feature (i.e. input coordinate)
- for continuous variables, splits always of the form

$$x_i \leqslant t$$

- for discrete variables, partitions values into two groups

From Criminisi et al. MSR-TR-2011-114, 28 October 2011.

Regularization of decision trees

- What will happen if we keep splitting the data?
  - Every data point will be in its own region—overfitting.
- When to stop splitting? (control complexity of the hypothesis space)
  - Limit number of total nodes.
  - Limit number of terminal nodes.
  - Limit tree depth.
  - Require minimum number of data points in a terminal node.

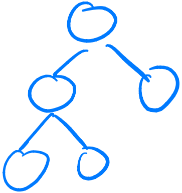
How to split

Goal Find a tree that minimize the task loss (e.g., squared loss) within a given complexity.

Problem Finding the optimal binary tree is computationally intractable.

Solution *Greedy* algorithm.

(x_j, t)



- Find the best split (according to some criteria) for a non-terminal node (initially the root)
- Add two children nodes
- Repeat until a stopping criterion is reached (e.g., max depth)

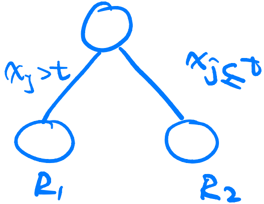
Evaluate splits

Let's think about what makes a good split.

Which one is better? $+$ $-$

Split 1 $R_1 : 8+/2-$ $R2 : 2+/8-$ 4

Split 2 $R_1 : 6+/4-$ $R2 : 1+/9-$ 5



Which one is better?

Split 1 $R_1 : 8+/2-$ $R2 : 2+/8-$ 4

Split 2 $R_1 : 6+/4-$ $R2 : 0+/10-$ 4

In general, we want to produce *pure* nodes, i.e. close to single-class node.

Misclassification error in a node

Let's formalize things a bit.

- Consider classification case: $\mathcal{Y} = \{1, 2, \ldots, K\}$.
- What's in a node?
  - Let node m represent region R_m, with N_m observations
  - Denote proportion of observations in R_m with class k by

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

- Predict the majority class in node m:

$$k(m) = \arg\max_k \hat{p}_{mk}.$$

- Misclassification rate in node m:

$$1 - \hat{p}_{mk(m)}.$$

Node Impurity Measures

How to quantify impurity?

- Three measures of **node impurity** for leaf node m:

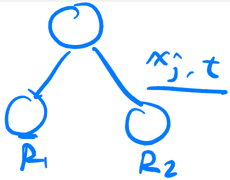
  Misclassification error
  $$1-\hat{p}_{mk(m)}.$$

  Gini index
  $$\sum_{k=1}^{K}\hat{p}_{mk}(1-\hat{p}_{mk})$$

  Entropy / Information gain
  $$-\sum_{k=1}^{K}\hat{p}_{mk}\log\hat{p}_{mk}.$$

- Gini index and entropy work well in practice.

Impurity of a split



A potential split produces two nodes, R_L and R_R. How do we score it?

- Suppose we have N_L points in R_L and N_R points in R_R.
- Let $Q(R_L)$ and $Q(R_R)$ be the node impurity measures for each node.
- Then find split that minimizes the *weighted average of node impurities*:

$$\frac{N_L Q(R_L) + N_R Q(R_R)}{N_L + N_R}$$

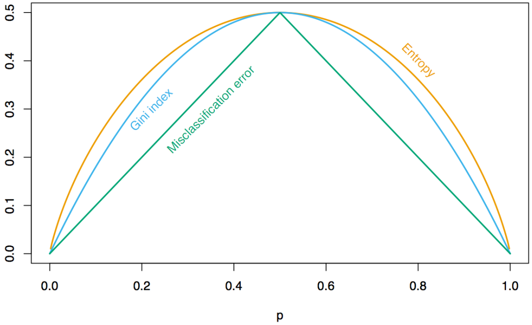
$\underline{\alpha} Q(R_L) + \underline{\beta} Q(R_R)$

Example:

$R_1 : 8+/2- \quad R2 : 1+/4-$
What's the weighted misclassification rate?

[discussion]Two-Class Node Impurity Measures

Consider binary classification. Let p be the relative frequency of class 1.



Misclassification error is not strictly concave thus may not guarantee improvement over the parent node.

HTF Figure 9.3

Finding the Split Point

How to find a split point that minimizes a given impurity measure?

- Enumerate d features and $n-1$ split points for each feature.

x_j 3 ex.

- Consider splitting on the j'th feature x_j.

.2 || .5 1.5

- If $x_{j(1)}, \ldots, x_{j(n)}$ are the sorted values of the j'th feature,
  - we only need to check split points between adjacent values
  - traditionally take split points halfway between adjacent values:

$$s_j \in \left\{ \frac{1}{2}\left(x_{j(r)} + x_{j(r+1)}\right) \mid r = 1, \ldots, n-1 \right\}. \qquad n-1 \text{ splits} \tag{1}$$

Regression trees

- Predict the mean value of a node

$$k(m) = \text{mean}(y_i \mid x_i \in R_m). \tag{2}$$

- Squared loss as the node impurity measure.
- Everything else remains the same as classification trees.

[discussion]Categorical features

- For a categorical feature, we split its values into two groups.
- Given a set of categories of size k, how many distinct splits? (its power set)
- Finding the optimal split is intractable in general.
- Approximations

  Numeric encoding: Randomly assign a number to each category
  - Binary classification: proportion of class 0
  - Regression: mean of targets of examples in the category, i.e. **mean encoding**

  One-hot encoding: May grow imbalanced trees, e.g., left-branching

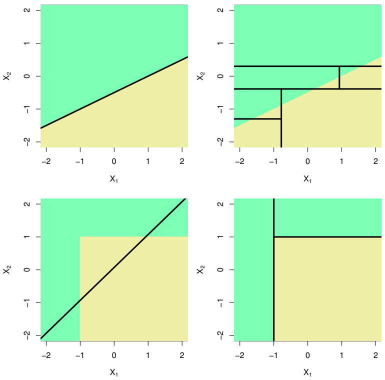
  Binary encoding: Robust to large cardinality
- Statistical issues with categorical features
  - If a category has a very large number of categories, we can overfit.

Interpretability

- Trees are certainly easier to explain than other classifiers.
- Can be used to discover non-linear features.
- Small trees seem interpretable. For large trees, maybe not so easy.
- Approximate neural network decision boundaries to gain interpretability
  - Wu M, Hughes M, Parbhoo S, Zazzi M, Roth V, Doshi-Velez F. Beyond Sparsity: Tree Regularization of Deep Models for Interpretability. Association for the Advancement of Artificial Intelligence (AAAI). 2018

Trees vs linear models

Trees have to work much harder to capture linear relations.



Review

Decision trees:

- *Non-linear* classifier that recursively partitions the input space.
- Non-metric: make no use of geometry, i.e. no inner-product or distances.
- Non-parametric: make no assumption of the data distribution.

Pros:

- Simple to understand.
- Interpretable, feature selection for free.

Cons:

- Poor linear modeling.
- Unstable / high variance, tend to overfit. $\rightarrow$ Next, how to fix this.