Decision trees

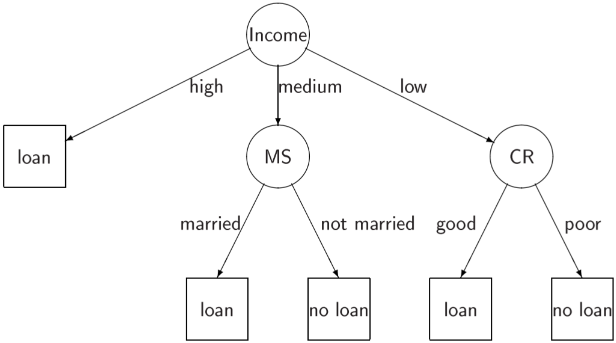
Victor Kitov

v.v.kitov@yandex.ru

Yandex School of Data Analysis

Table of Contents

Example of decision tree



Definition of decision tree

- Prediction is performed by tree T:
    - directed graph
    - without loops
    - with single root node

Definition of decision tree

?

- for each internal node t a check-function $Q_t(x)$ is associated
- for each of K_t edges a set of values of check-function $Q_t(x)$ is associated: $S_t(1), ...S_t(K_t)$ such that:
  - $\bigcup_k S_t(k) = range[Q_t]$
  - $S_t(i) \cap S_t(j) = \emptyset \; \forall i \neq j$

Prediction process

- a set of nodes is divided into:
  - internal nodes $int(T)$, each having ≥ 2 child nodes
  - terminal nodes $terminal(T)$, which do not have child nodes but have associated prediction values.

Prediction process

- a set of nodes is divided into:
    - internal nodes $int(T)$, each having ≥ 2 child nodes
    - terminal nodes $terminal(T)$, which do not have child nodes but have associated prediction values.
- Prediction process for tree T:
    - $t = root(T)$
    - while t is not a leaf node:
        - calculate $Q_t(x)$
        - determine j such that $Q_t(x) \in S_t(j)$
        - follow edge to j-th child node: $t = \tilde{t}_j$
    - return prediction, associated with leaf t.

Specification of decision tree

- To define a decision tree one needs to specify:
  - the check-function: $Q_t(x)$
  - the splitting criterion: K_t and $S_t(1), ...S_t(K_t)$
  - the termination criteria (when node is defined as a terminal node)
  - the predicted value for each leaf node.

Table of Contents

Possible definitions of splitting rules

1. $S_t(1)=\{x^{i(t)}\le h_t\}$, $S_t(2)=\{x^{i(t)}>h_t\}$

Possible definitions of splitting rules

1. $S_t(1)=\{x^{i(t)}\le h_t\}$, $S_t(2)=\{x^{i(t)}>h_t\}$
2. $Q_t(x)=x^{i(t)}$, where $S_t(j)=v_j$, where $v_1,...v_K$ are unique values of feature $x^{i(t)}$.

Possible definitions of splitting rules

1. $S_t(1) = \{x^{i(t)} \le h_t\}$, $S_t(2) = \{x^{i(t)} > h_t\}$
2. $Q_t(x) = x^{i(t)}$, where $S_t(j) = v_j$, where $v_1, ...v_K$ are unique values of feature $x^{i(t)}$.
3. $S_t(j) = \{h_j < x^{i(t)} \le h_{j+1}\}$ for set of partitioning thresholds $h_1, h_2, ...h_{K_t+1}$.

Possible definitions of splitting rules

1. $S_t(1) = \{x^{i(t)} \le h_t\}$, $S_t(2) = \{x^{i(t)} > h_t\}$
2. $Q_t(x) = x^{i(t)}$, where $S_t(j) = v_j$, where $v_1, ...v_K$ are unique values of feature $x^{i(t)}$.
3. $S_t(j) = \{h_j < x^{i(t)} \le h_{j+1}\}$ for set of partitioning thresholds $h_1, h_2, ...h_{K_t+1}$.
4. $S_t(1) = \{x : \langle x, v\rangle \le 0\}, \quad S_t(2) = \{x : \langle x, v\rangle > 0\}$

Possible definitions of splitting rules

1. $S_t(1) = \{x^{i(t)} \le h_t\}$, $S_t(2) = \{x^{i(t)} > h_t\}$
2. $Q_t(x) = x^{i(t)}$, where $S_t(j) = v_j$, where $v_1, ...v_K$ are unique values of feature $x^{i(t)}$.
3. $S_t(j) = \{h_j < x^{i(t)} \le h_{j+1}\}$ for set of partitioning thresholds $h_1, h_2, ...h_{K_t+1}$.
4. $S_t(1) = \{x : \langle x, v\rangle \le 0\}, \quad S_t(2) = \{x : \langle x, v\rangle > 0\}$
5. $S_t(1) = \{x : \|x\| \le h\}, \quad S_t(2) = \{x : \|x\| > h\}$

etc.

Most famous decision tree algorithms

- CART (classification and regression trees)
    - implemented in scikit-learn
- C4.5

CART version of splitting rule

- single feature value is considered:

$$Q_t(x) = x^{i(t)}$$

- binary splits:

$$K_t = 2$$

- split based on threshold h_t:

$$S_1 = \{x^{i(t)} \leq h_t\},\ S_2 = \{x^{i(t)} > h_t\}$$

- $h(t) \in \{x_1^{i(t)}, x_2^{i(t)}, ...x_N^{i(t)}\}$
    - applicable only for real, ordinal and binary features
    - discrete unordered features:

CART version of splitting rule

- single feature value is considered:

$$Q_t(x) = x^{i(t)}$$

- binary splits:

$$K_t = 2$$

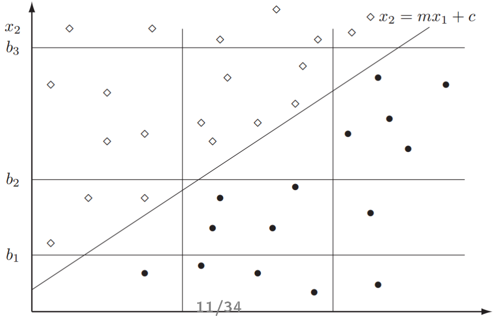
- split based on threshold h_t:

$$S_1 = \{x^{i(t)} \le h_t\},\ S_2 = \{x^{i(t)} > h_t\}$$

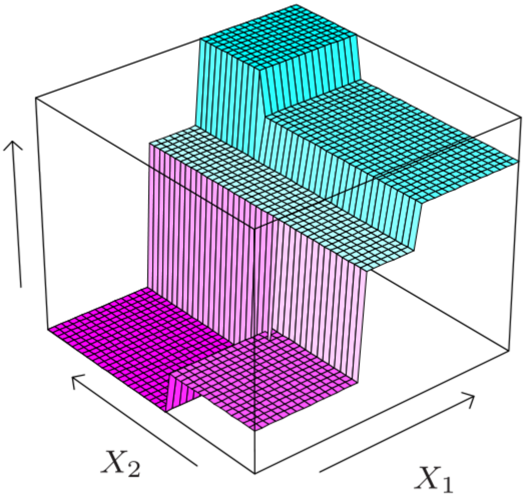
- $h(t) \in \{x_1^{i(t)}, x_2^{i(t)}, ...x_N^{i(t)}\}$
    - applicable only for real, ordinal and binary features
    - discrete unordered features:may use one-hot encoding.

Analysis of CART splitting rule

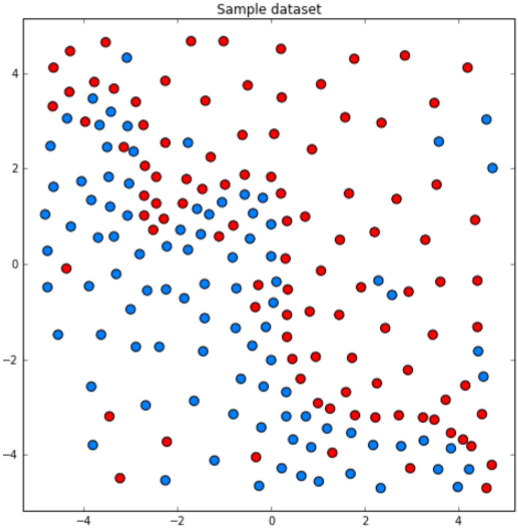
- Advantages:
  - simplicity
  - estimation efficiency
  - interpretability
- Drawbacks:
  - many nodes may be needed to describe boundaries not parallel to axes:



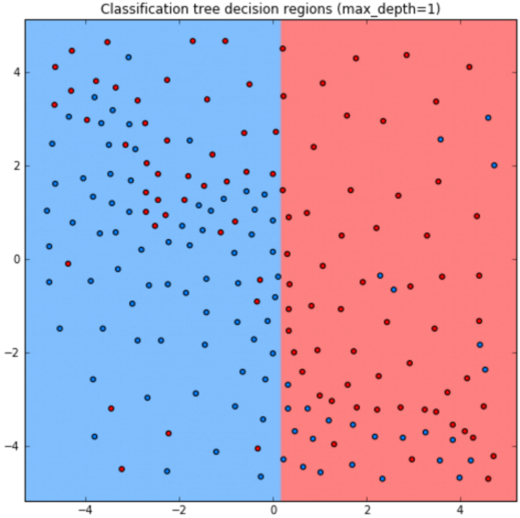
Piecewise constant predictions of decision trees



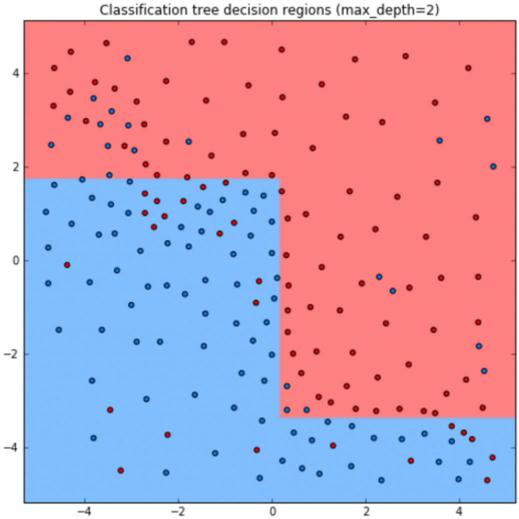
Sample dataset



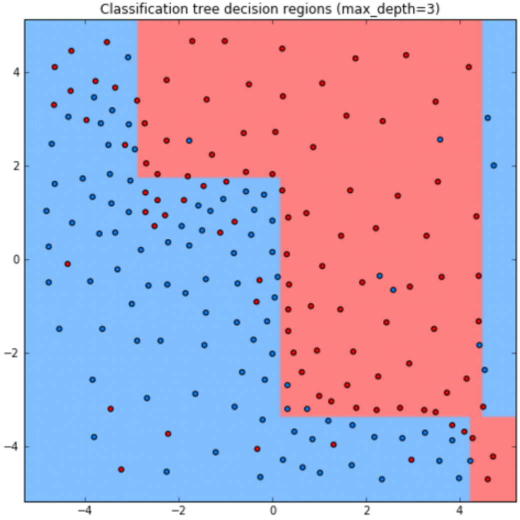
Example: Decision tree classification



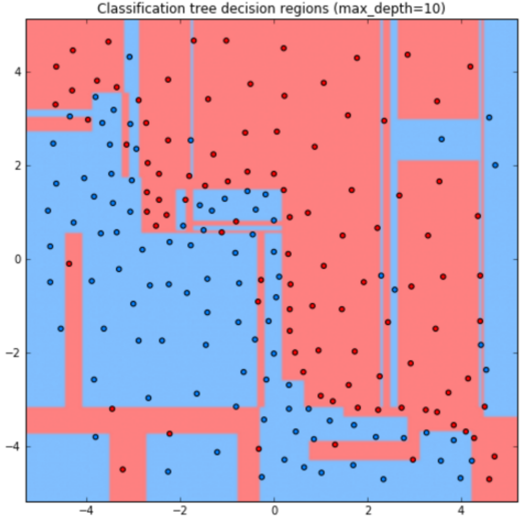
Example: Decision tree classification



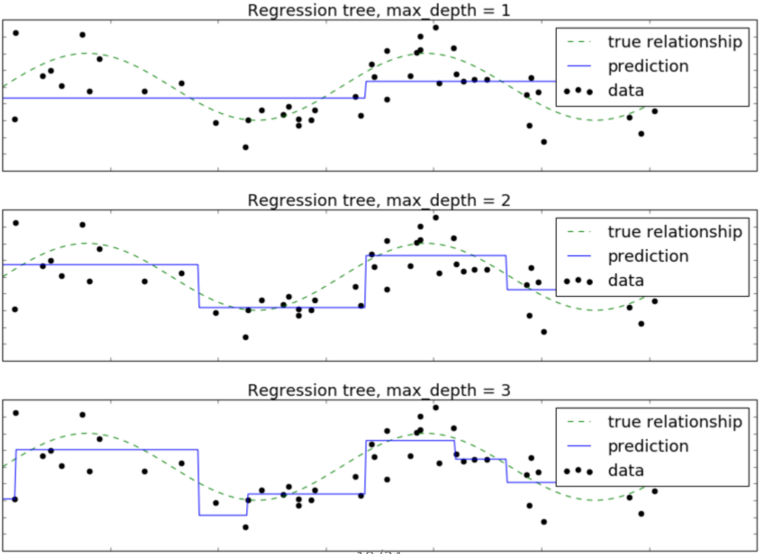
Example: Decision tree classification



Example: Decision tree classification



Example: Regression tree



Example: Regression tree

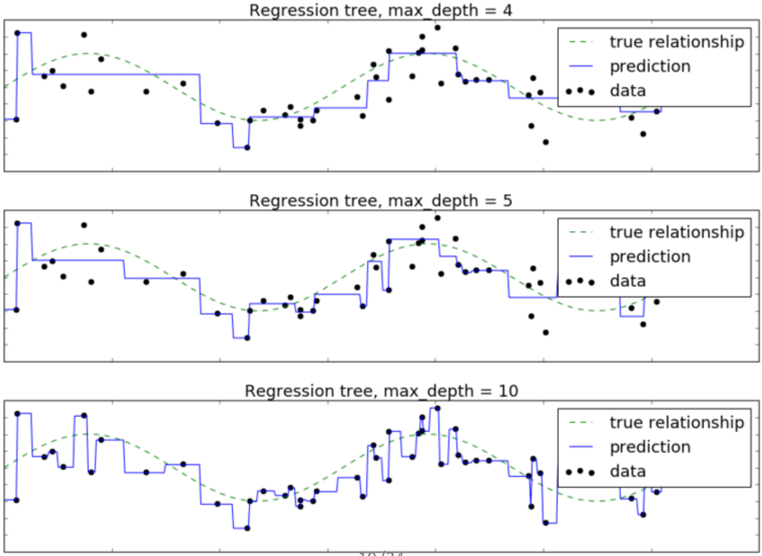


Table of Contents

Impurity function

- Impurity function $\phi(t) = \phi(p(y = 1|t), ...p(y = C|t))$ measures the mixture of classes using class probabilities inside node t.
- It can be any function $\phi(p_1, p_2, ...p_C)$ with the following properties:
  - ϕ is defined for $p_j \geq 0$ and $\sum_j p_j = 1$.
  - ϕ attains maximum for $p_j = 1/C$, $k = 1, 2, ...C$.
  - ϕ attains minimum when $\exists j : p_j = 1$, $p_i = 0\ \forall i \neq j$.
  - ϕ is symmetric function of $p_1, p_2, ...p_C$.
- Note: in regression $\phi(t)$ measures the spread of y inside node t.
  - may be MSE, MAE.

Typical impurity functions

- **Gini criterion**
  - interpretation: probability to make mistake when predicting class randomly with class probabilities $[p(\omega_1|t), ...p(\omega_C|t)]$:

$$I(t) = \sum_i p_i(1 - p_i) = 1 - \sum_i [p_i]^2$$

http://scikit-learn.org/stable/modules/tree.html

- **Entropy**
  - interpretation: measure of uncertainty of random variable

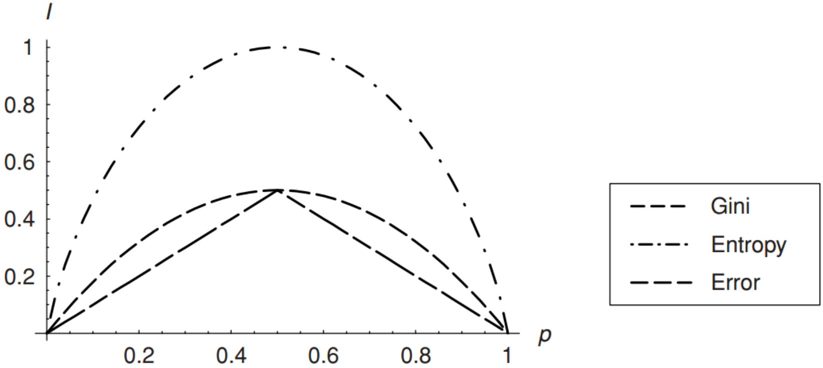
$$I(t) = -\sum_i p_i \ln p_i$$

- **Classification error**
  - interpretation: frequency of errors when classifying with the most common class

$$I(t) = 1 - \max_i p_i$$

Typical impurity functions

Impurity functions for binary classification with class probabilities $p = p_1$ and $1 - p = p_2$.



Splitting criterion selection

$$\Delta I(t)=I(t)-\sum_{i=1}^{R}I(t_i)\frac{N(t_i)}{N(t)}$$

- $\Delta I(t)$ is the quality of the split[1] of node t into child nodes $t_1,...t_R$.

[1] If $I(t)$ is entropy, then $\Delta I(t)$ is called *information gain*.

Splitting criterion selection

$$\Delta I(t) = I(t) - \sum_{i=1}^{R} I(t_i)\frac{N(t_i)}{N(t)}$$

- $\Delta I(t)$ is the quality of the split[1] of node t into child nodes $t_1, ...t_R$.
- CART selection: select feature i_t and threshold h_t, which maximize $\Delta I(t)$:

$$i_t,\ h_t = \arg\max_{k,h} \Delta I(t)$$

- CART decision making: from node t follow:

$$\begin{cases} \text{left child } t_1, & \text{if } x^{i_t} \leq h_t \\ \text{right child } t_2, & \text{if } x^{i_t} > h_t \end{cases}$$

[1] If $I(t)$ is entropy, then $\Delta I(t)$ is called *information gain*.

Table of Contents

Regression: prediction assignment for leaf nodes[2]

- Define $I_t = \{i : x_i \in \text{node } t\}$
- For mean squared error loss (MSE):

$$\widehat{y} = \arg\min_{\mu} \sum_{i \in I_t} (y_i - \mu)^2 = \frac{1}{|I_t|} \sum_{i \in I_t} y_i,$$

- For mean absolute error loss (MAE):

$$\widehat{y} = \arg\min_{\mu} \sum_{i \in I_t} |y - \mu| = median\{y_i : i \in I_t\}.$$

[2] Prove optimality of estimators for MSE and MAE loss.

Classification: prediction assignment for leaf nodes

- Define $\lambda(\omega_i, \omega_j)$ - the cost of predicting object of class ω_i as belonging to class ω_j.
    - For $\lambda(\omega_i, \omega_j) = \mathbb{I}[\omega_i \neq \omega_j]$:

Classification: prediction assignment for leaf nodes

- Define $\lambda(\omega_i, \omega_j)$ - the cost of predicting object of class ω_i as belonging to class ω_j.
    - For $\lambda(\omega_i, \omega_j) = \mathbb{I}[\omega_i \neq \omega_j]$:most common class will be associated with the leaf node:

$$c = \arg\max_{\omega} |\{i : \ i \in I_t, \ y_i = \omega\}|$$

    - Minimum loss class assignment:

Classification: prediction assignment for leaf nodes

- Define $\lambda(\omega_i, \omega_j)$ - the cost of predicting object of class ω_i as belonging to class ω_j.
  - For $\lambda(\omega_i, \omega_j) = \mathbb{I}[\omega_i \neq \omega_j]$:most common class will be associated with the leaf node:

$$c = \arg\max_{\omega} \left|\{i : \ i \in I_t, \ y_i = \omega\}\right|$$

  - Minimum loss class assignment:

$$c = \arg\min_{\omega} \sum_{i \in I_t} \lambda(c_i, \omega)$$

Table of Contents

Termination criterion

- Bias-variance tradeoff:
    - very large complex trees -> overfitting
    - very short simple trees -> underfitting
- Approaches to stopping:
    - rule-based
    - based on pruning

5 Termination criterion

- Rule based termination

Rule-base termination criteria

- Rule-based: a criterion is compared with a threshold.
- Variants of criterion:
    - depth of tree
    - number of objects in a node
    - minimal number of objects in one of the child nodes
    - impurity of classes
    - change of impurity of classes after the split

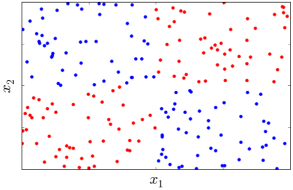
Analysis of rule-based termination

Advantages:

- simplicity
- interpretability

Disadvantages:

- specification of threshold is needed
- impurity change is suboptimal: further splits may become better than current one, like here:



- that's why pruning gives better results-build the tree up to the end and then remove redundant nodes.

Handling missing values

If checked feature is missing:

- we may always (for any ML method) fill missing values:
    - with feature mean
    - with new categorical value "missing" (for categorical values)
    - predict them using other known features
- CART uses prediction of unknown feature using another feature that best predicts the missing one: "surrogate split" - technique
- ID3 and C4.5 decision trees use averaging of predictions made by each child node with weights $N(t_1)/N(t)$, $N(t_2)/N(t), \dots N(t_S)/N(t)$.

Analysis of decision trees

- Advantages:
  - simplicity
  - interpretability
  - implicit feature selection
  - naturally handles both discrete and real features
  - prediction is invariant to monotone transformations of features for $Q_t(x) = x^{i(t)}$
    - work well for features of different nature
- Disadvantages:
  - non-parallel to axes class separating boundary may lead to many nodes in the tree for $Q_t(x) = x^{i(t)}$
  - one step ahead lookup strategy for split selection may be insufficient (XOR example)
  - not online - slight modification of the training set will require full tree reconstruction.