Supervised Learning

Chapter VI: Decision Tree Learning

Johannes Jurgovsky

Outline

Decision Tree Learning

1. Introduction
2. Impurity Function
3. Algorithms
4. Pruning
5. Ensemble Learning
6. Random Forest

1. Introduction

Introduction: Decision Tree Basics

Specification of Classification Problems [ML Introduction]

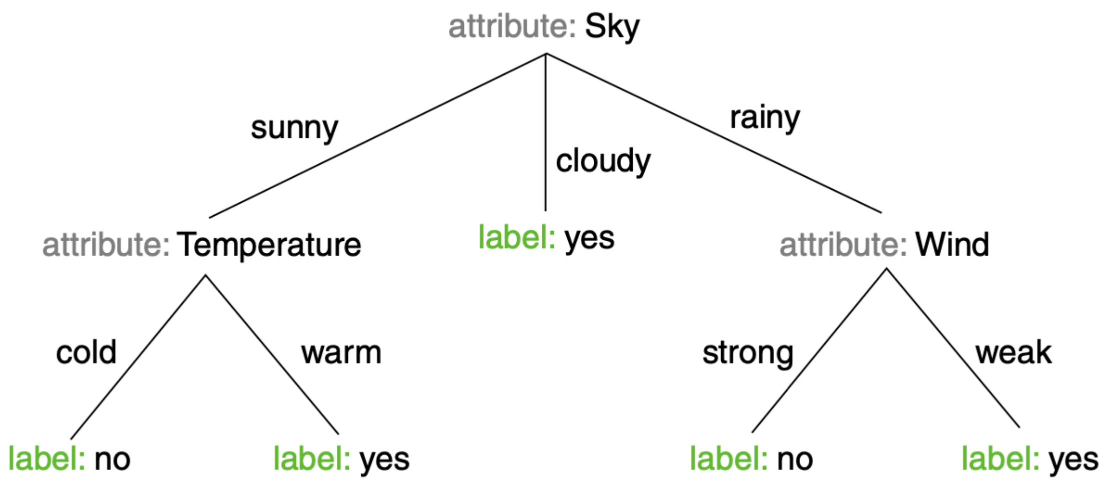
Characterization of the model (model world):

- X is a set of feature vectors, also called feature space.
- Y is a set of classes.
- $y : X \to Y$ is the ideal classifier for X.
- $D = \{(\mathbf{x}_1, y(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, y(\mathbf{x}_n))\} \subseteq X \times Y$ is a set of examples.

Introduction: Decision Tree Basics

Decision Tree for the Concept “EnjoySport”

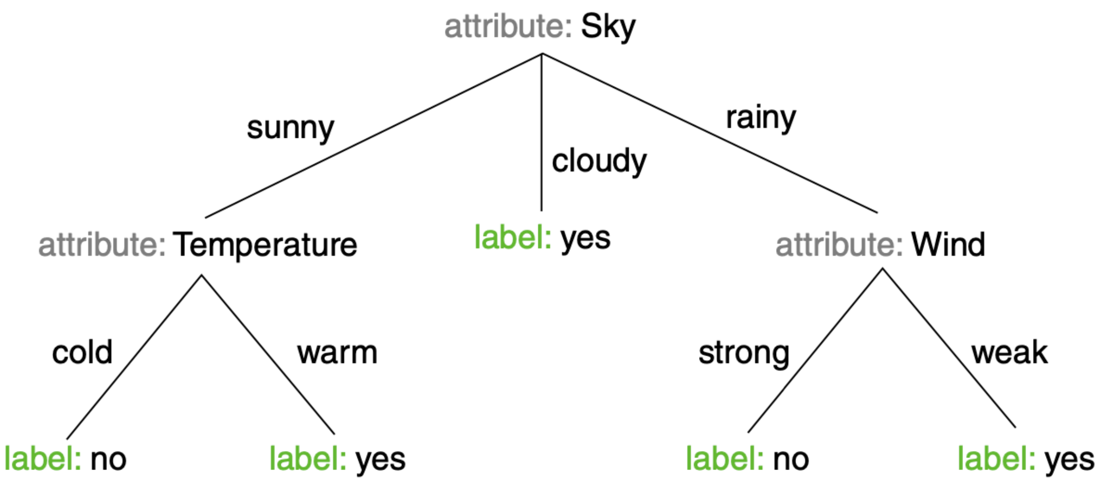
Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
1	sunny	warm	normal	strong	warm	same	yes
2	sunny	warm	high	strong	warm	same	yes
3	rainy	cold	high	strong	warm	change	no
4	sunny	warm	high	strong	cool	change	yes



Introduction: Decision Tree Basics

Decision Tree for the Concept "EnjoySport"

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
1	sunny	warm	normal	strong	warm	same	yes
2	sunny	warm	high	strong	warm	same	yes
3	rainy	cold	high	strong	warm	change	no
4	sunny	warm	high	strong	cool	change	yes



Partitioning of X at the root node:

$$X = \{\mathbf{x} \in X : \mathbf{x}|_{\text{Sky}} = \text{sunny}\} \cup \{\mathbf{x} \in X : \mathbf{x}|_{\text{Sky}} = \text{cloudy}\} \cup \{\mathbf{x} \in X : \mathbf{x}|_{\text{Sky}} = \text{rainy}\}$$

Introduction: Decision Tree Basics

Definition 1 (Splitting)

Let X be feature space and let D be a set of examples. A splitting of X is a partitioning of X into mutually exclusive subsets $X_1, \ldots, X_s$. I.e., $X = X_1 \cup \ldots \cup X_s$ with $X_j \neq \emptyset$ and $X_j \cap X_{j'} = \emptyset$, where $j, j' \in \{1, \ldots, s\}, j \neq j'$.

A splitting $X_1, \ldots, X_s$ of X induces a splitting $D_1, \ldots, D_s$ of D, where D_j, $j = 1, \ldots, s$, is defined as $\{(\mathbf{x}, y(\mathbf{x})) \in D \mid \mathbf{x} \in X_j\}$.

Introduction: Decision Tree Basics

Definition 1 (Splitting)

Let X be feature space and let D be a set of examples. A splitting of X is a partitioning of X into mutually exclusive subsets $X_1, \ldots, X_s$. I.e., $X = X_1 \cup \ldots \cup X_s$ with $X_j \neq \emptyset$ and $X_j \cap X_{j'} = \emptyset$, where $j, j' \in \{1, \ldots, s\}, j \neq j'$.

A splitting $X_1, \ldots, X_s$ of X induces a splitting $D_1, \ldots, D_s$ of D, where D_j, $j = 1, \ldots, s$, is defined as $\{(\mathbf{x}, y(\mathbf{x})) \in D \mid \mathbf{x} \in X_j\}$.

A splitting depends on the measurement scale of a feature:

1. m-ary splitting induced by a (nominal) feature A with finite domain:

   $A = \{a_1, \ldots, a_m\}: \; X = \{\mathbf{x} \in X : \mathbf{x}|_A = a_1\} \cup \ldots \cup \{\mathbf{x} \in X : \mathbf{x}|_A = a_m\}$

Introduction: Decision Tree Basics

Definition 1 (Splitting)

Let X be feature space and let D be a set of examples. A splitting of X is a partitioning of X into mutually exclusive subsets $X_1, \ldots, X_s$. I.e., $X = X_1 \cup \ldots \cup X_s$ with $X_j \neq \emptyset$ and $X_j \cap X_{j'} = \emptyset$, where $j, j' \in \{1, \ldots, s\}, j \neq j'$.

A splitting $X_1, \ldots, X_s$ of X induces a splitting $D_1, \ldots, D_s$ of D, where D_j, $j = 1, \ldots, s$, is defined as $\{(\mathbf{x}, y(\mathbf{x})) \in D \mid \mathbf{x} \in X_j\}$.

A splitting depends on the measurement scale of a feature:

1. m-ary splitting induced by a (nominal) feature A with finite domain:

   $A = \{a_1, \ldots, a_m\}: \quad X = \{\mathbf{x} \in X : \mathbf{x}|_A = a_1\} \cup \ldots \cup \{\mathbf{x} \in X : \mathbf{x}|_A = a_m\}$

2. Binary splitting induced by a (nominal) feature A:

   $A' \subset A: \quad X = \{\mathbf{x} \in X : \mathbf{x}|_A \in A'\} \cup \{\mathbf{x} \in X : \mathbf{x}|_A \notin A'\}$

3. Binary splitting induced by an ordinal feature A:

   $v \in \mathit{dom}(A): \quad X = \{\mathbf{x} \in X : \mathbf{x}|_A \succeq v\} \cup \{\mathbf{x} \in X : \mathbf{x}|_A \prec v\}$

Remarks:

- The syntax $\mathbf{x}|_A$ denotes the projection operator, which returns that vector component (dimension) of $\mathbf{x} = (x_1, \ldots, x_p)$ that is associated with A. Without loss of generality this projection can be presumed being unique.
- A splitting of X into two disjoint, non-empty subsets is called a binary splitting.
- We consider only splittings of X that are induced by a splitting of a single feature A of X.
  Keyword: monothetic splitting

Decision Trees Basics

Definition 2 (Decision Tree)

Let X be feature space and let Y be a set of classes. A decision tree T for X and Y is a finite tree with a distinguished root node. A non-leaf node t of T has assigned (1) a set $X(t) \subseteq X$, (2) a splitting of $X(t)$, and (3) a one-to-one mapping of the subsets of the splitting to its successors.

$X(t) = X$ iff t is root node. A leaf node of T has assigned a class from Y.

Introduction: Decision Tree Basics

Definition 2 (Decision Tree)

Let X be feature space and let Y be a set of classes. A decision tree T for X and Y is a finite tree with a distinguished root node. A non-leaf node t of T has assigned (1) a set $X(t) \subseteq X$, (2) a splitting of $X(t)$, and (3) a one-to-one mapping of the subsets of the splitting to its successors.

$X(t) = X$ iff t is root node. A leaf node of T has assigned a class from Y.

Classification of some $\mathbf{x} \in X$ given a decision tree T:

1. Find the root node of T.
2. If t is a non-leaf node, find among its successors that node whose subset of the splitting of $X(t)$ contains $\mathbf{x}$. Repeat this step.
3. If t is a leaf node, label $\mathbf{x}$ with the respective class.

→ The set of possible decision trees forms the hypothesis space H.

Remarks:

- The classification of an **x** $\in X$ determines a unique path from the root node of T to some leaf node of T.
- At each non-leaf node a particular feature of **x** is evaluated in order to find the next node along with a possible next feature to be analyzed.
- Each path from the root node to some leaf node corresponds to a conjunction of feature values, which are successively tested. This test can be formulated as a decision rule. Example:

  IF Sky=rainy AND Wind=weak THEN EnjoySport=yes

  If all tests in T are of the kind shown in the example, namely, a comparison with a single feature value, all feature domains must be finite.
- If in all non-leaf nodes of T only one feature is evaluated at a time, T is called a *monothetic* decision tree. Examples for *polythetic* decision trees are the so-called oblique decision trees.
- Decision trees became popular in 1986, with the introduction of the ID3 Algorithm by J. R. Quinlan.

Introduction: Decision Tree Basics

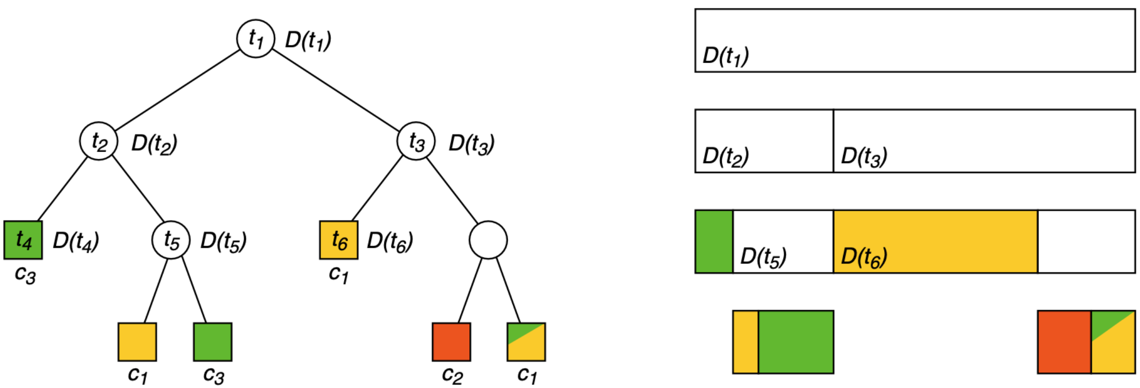
Notation

Let T be decision tree for X and Y, let D be a set of examples, and let t be a node of T. Then we agree on the following notation:

- $X(t)$ denotes the subset of the feature space X that is represented by t.
  (as used in the decision tree definition)

- $D(t)$ denotes the subset of the example set D that is represented by t, where $D(t) = \{(\mathbf{x}, y(\mathbf{x})) \in D \mid \mathbf{x} \in X(t)\}$. (see the splitting definition)

Illustration:



Remarks:

- The set $X(t)$ is comprised of those members **x** of X that are filtered by a path from the root node of T to the node t.
- *leaves*(T) denotes the set of all leaf nodes of T.
- A single node t of a decision tree T, and hence T itself, encode a piecewise constant function. This way, t as well as T can form complex non-linear classifiers. The functions encoded by t and T differ in the number of evaluated features of **x**, which is one for t and the tree height for T.
- In the following we will use the symbols "t" and "T" to denote also the classifiers that are encoded by a node t and a tree T respectively:

$$t, T : X \to Y \quad \text{(instead of } y_t, y_T : X \to Y\text{)}$$

Introduction: Decision Tree Basics

Algorithm Template: Construction

Algorithm:	*DT-construct*	Decision Tree Construction
Input:	D	(Sub)set of examples.
Output:	t	Root node of a decision (sub)tree.

DT-construct(D)

1. $t =$ *newNode*$()$
   label$(t) =$ *representativeClass*(D)
2. **IF** *impure*(D)
   THEN *criterion* $=$ *splitCriterion*(D)
   ELSE *return*(t)
3. $\{D_1, \ldots, D_s\} =$ *decompose*$(D,$ *criterion*$)$
4. **FOREACH** D' **IN** $\{D_1, \ldots, D_s\}$ **DO**

   addSuccessor$(t,$ *DT-construct*$(D'))$

   ENDDO
5. *return*(t)

[Illustration]

Introduction: Decision Tree Basics

Algorithm Template: Classification

Algorithm:	*DT-classify*	Decision Tree Classification
Input:	$\mathbf{x}$	Feature vector.
	t	Root node of a decision (sub)tree.
Output:	$y(\mathbf{x})$	Class of feature vector $\mathbf{x}$ in the decision (sub)tree below t.

DT-classify$(\mathbf{x}, t)$

1. **IF** *isLeafNode*(t)
   THEN *return*(*label*(t))
   ELSE *return*(*DT-classify*$(\mathbf{x}$, *splitSuccessor*$(t, \mathbf{x})$))

Remarks:

- Since *DT-construct* assigns to each node of a decision tree T a class, each subtree of T (as well as each pruned version of a subtree of T) represents a valid decision tree on its own.

- Functions of *DT-construct*:
  - *representativeClass*(D)
    Returns a representative class for the example set D. Note that, due to pruning, each node may become a leaf node.
  - *impure*(D)
    Evaluates the (im)purity of a set D of examples.
  - *splitCriterion*(D)
    Returns a split criterion for $X(t)$ based on the examples in $D(t)$.
  - *decompose*$(D,$ *criterion*$)$
    Returns a splitting of D according to *criterion*.
  - *addSuccessor*(t, t')
    Inserts the successor t' for node t.

- Functions of *DT-classify*:
  - *isLeafNode*(t)
    Tests whether t is a leaf node.
  - *splitSuccessor*$(t, \mathbf{x})$
    Returns the (unique) successor t' of t for which $\mathbf{x} \in X(t')$ holds.

Introduction: Decision Tree Basics

When to Use Decision Trees

Problem characteristics that may suggest a decision tree classifier:

- the objects can be described by feature-value combinations.
- the domain and range of the target function are discrete
- hypotheses take the form of disjunctions
- the training set contains noise

Selected application areas:

- medical diagnosis
- fault detection in technical systems
- risk analysis for credit approval
- basic scheduling tasks such as calendar management
- classification of design flaws in software engineering

Introduction: Decision Tree Basics

On the Construction of Decision Trees

- How to exploit an example set both efficiently and effectively?
- According to what rationale should a node become a leaf node?
- How to assign a class for nodes of impure example sets?
- How to evaluate decision tree performance?

2. Impurity Function

Impurity Function:
Splitting

Let t be a leaf node of an incomplete decision tree, and let $D(t)$ be the subset of the example set D that is represented by t. [Illustration]

Possible criteria for a splitting of $X(t)$:

1. Size of $D(t)$.

2. Purity of $D(t)$.

3. Ockham's Razor.

Impurity Function:
Splitting

Let t be a leaf node of an incomplete decision tree, and let $D(t)$ be the subset of the example set D that is represented by t. [Illustration]

Possible criteria for a splitting of $X(t)$:

1. Size of $D(t)$.
   $D(t)$ will not be partitioned further if the number of examples, $|D(t)|$, is below a certain threshold.

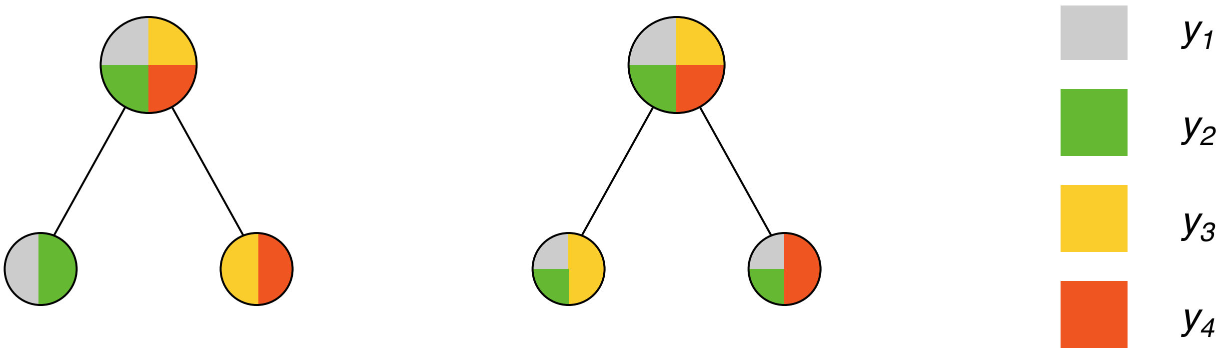
2. Purity of $D(t)$.
   $D(t)$ will not be partitioned further if all examples in D are members of the same class.

3. Ockham's Razor.
   $D(t)$ will not be partitioned further if the resulting decision tree is not improved significantly by the splitting.

Impurity Function:

Splitting (continued)

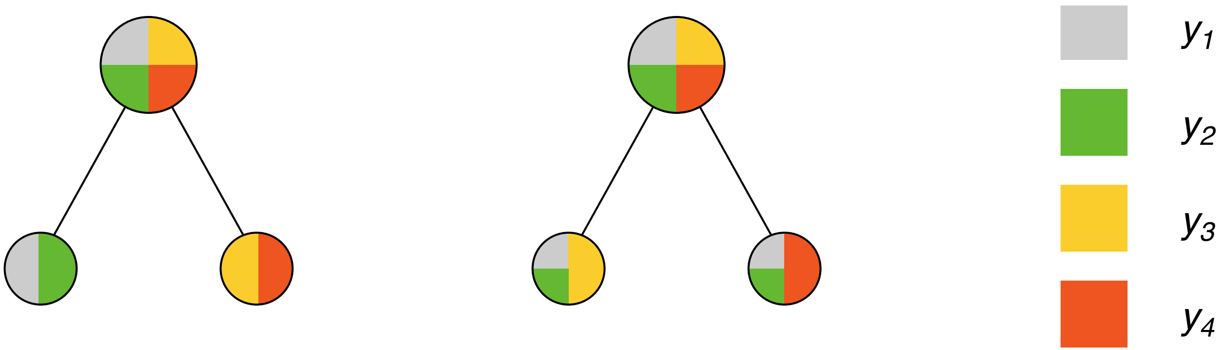
Let D be a set of examples over a feature space X and a set of classes $Y = \{y_1, y_2, y_3, y_4\}$. Distribution of D for two possible splittings of X :



Impurity Function:

Splitting (continued)

Let D be a set of examples over a feature space X and a set of classes $Y = \{y_1, y_2, y_3, y_4\}$. Distribution of D for two possible splittings of X :



- The left splitting should be preferred, since it minimizes the *impurity* of the subsets of D in the leaf nodes. The argumentation presumes that the misclassification costs are independent of the classes in Y.

- The impurity is a function defined on $\mathcal{P}(D)$, the set of all subsets of an example set D.

Impurity Function:

Definition 3 (Impurity Function ι)

Let $k \in \mathbf{N}$. An impurity function $\iota : [0;1]^k \to \mathbf{R}$ is a partial function defined on the standard $k{-}1$-simplex Δ^{k-1} for which the following properties hold:

(a) ι becomes minimum at points $(1,0,\ldots,0), (0,1,\ldots,0), \ldots, (0,\ldots,0,1)$.

(b) ι is symmetric with regard to its arguments, $p_1, \ldots, p_k$.

(c) ι becomes maximum at point $(1/k, \ldots, 1/k)$.

Impurity Function:

Definition 4 (Impurity of an Example Set $\iota(D)$)

Let D be a set of examples, let $Y = \{y_1, \ldots, y_k\}$ be set of classes, and let $y : X \to Y$ be the ideal classifier for X. Moreover, let $\iota : [0;1]^k \to \mathbb{R}$ an impurity function. Then, the impurity of D, denoted as $\iota(D)$, is defined as follows:

$$\iota(D) = \iota\left(\frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_1\}|}{|D|}, \ldots, \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_k\}|}{|D|}\right)$$

Impurity Function:

Definition 4 (Impurity of an Example Set $\iota(D)$)

Let D be a set of examples, let $Y = \{y_1, \ldots, y_k\}$ be set of classes, and let $y : X \to Y$ be the ideal classifier for X. Moreover, let $\iota : [0;1]^k \to \mathbb{R}$ an impurity function. Then, the impurity of D, denoted as $\iota(D)$, is defined as follows:

$$\iota(D) = \iota\left(\frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_1\}|}{|D|}, \ldots, \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_k\}|}{|D|}\right)$$

Definition 5 (Impurity Reduction $\Delta\iota$)

Let $D_1, \ldots, D_s$ be a partitioning of an example set D, which is induced by a splitting of a feature space X. Then, the resulting impurity reduction, denoted as $\Delta\iota(D, \{D_1, \ldots, D_s\})$, is defined as follows:

$$\Delta\iota(D, \{D_1, \ldots, D_s\}) = \iota(D) - \sum_{j=1}^{s} \frac{|D_j|}{|D|} \cdot \iota(D_j)$$

Remarks:

- The standard $k-1$-simplex comprises all k-tuples with non-negative elements that sum to 1:
  $\Delta^{k-1} = \left\{(p_1, \ldots, p_k) \in \mathbb{R}^k : \sum_{i=1}^{k} p_i = 1 \text{ and } p_i \geq 0 \text{ for all } i\right\}$
- Observe the different domains of the impurity function ι in the Definitions 3 and 4, namely, $[0;1]^k$ and D. The domains correspond to each other: the set of examples, D, defines via its class portions an element from $[0;1]^k$ and vice versa.
- The properties in the definition of ι suggest to minimize the external path length of T with respect to D in order to minimize the overall impurity characteristics of T.
- Within the *DT-construct* algorithm usually a greedy strategy (local optimization) is employed to minimize the overall impurity characteristics of a decision tree T.

Impurity Function:

Impurity Functions Based on the Misclassification Rate

Definition for two classes:

$$\iota_{misclass}(p_1, p_2) = 1 - \max\{p_1, p_2\} = \begin{cases} p_1 & \text{if } 0 \leq p_1 \leq 0.5 \\ 1 - p_1 & \text{otherwise} \end{cases}$$

$$\iota_{misclass}(D) = 1 - \max\left\{\frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_1\}|}{|D|}, \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_2\}|}{|D|}\right\}$$

Impurity Function:

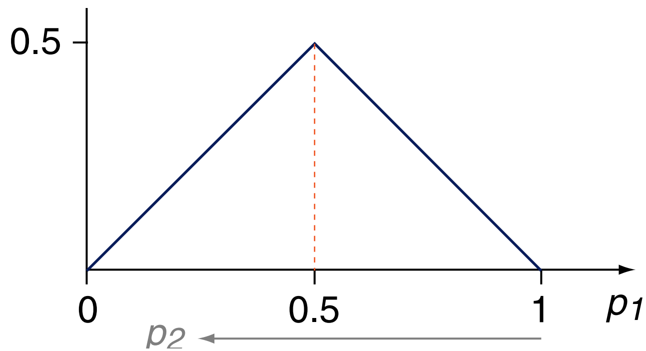
Impurity Functions Based on the Misclassification Rate

Definition for two classes:

$$\iota_{misclass}(p_1, p_2) = 1 - \max\{p_1, p_2\} = \begin{cases} p_1 & \text{if } 0 \leq p_1 \leq 0.5 \\ 1 - p_1 & \text{otherwise} \end{cases}$$

$$\iota_{misclass}(D) = 1 - \max\left\{\frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_1\}|}{|D|}, \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_2\}|}{|D|}\right\}$$

Graph of the function $\iota_{misclass}(p_1, 1 - p_1)$:



[Graph: Entropy, Gini]

Impurity Function:

Impurity Functions Based on the Misclassification Rate (continued)

Definition for k classes:

$$\iota_{misclass}(p_1, \ldots, p_k) = 1 - \max_{i=1,\ldots,k} \; p_i$$

$$\iota_{misclass}(D) = 1 - \max_{y \in Y} \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y\}|}{|D|}$$

Impurity Function:

Impurity Functions Based on Entropy (continued)

Definition for two classes:

$$\iota_{entropy}(p_1, p_2) = -(p_1 \log_2(p_1) + p_2 \log_2(p_2))$$

$$\iota_{entropy}(D) = -\left(\frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_1\}|}{|D|} \cdot \log_2 \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_1\}|}{|D|} + \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_2\}|}{|D|} \cdot \log_2 \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_2\}|}{|D|}\right)$$

Impurity Function:

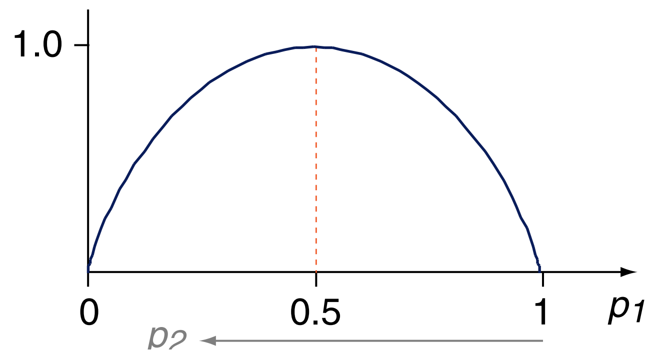
Impurity Functions Based on Entropy (continued)

Definition for two classes:

$$\iota_{entropy}(p_1, p_2) = -(p_1 \log_2(p_1) + p_2 \log_2(p_2))$$

$$\iota_{entropy}(D) = -\left(\frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_1\}|}{|D|} \cdot \log_2 \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_1\}|}{|D|} + \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_2\}|}{|D|} \cdot \log_2 \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_2\}|}{|D|} \right)$$

Graph of the function $\iota_{entropy}(p_1, 1 - p_1)$:

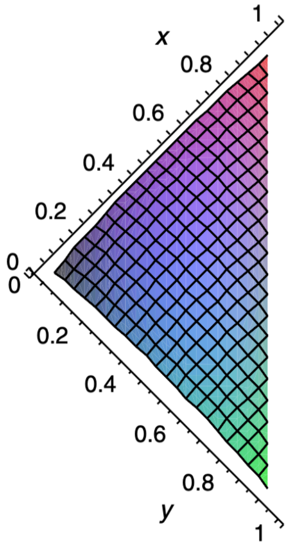


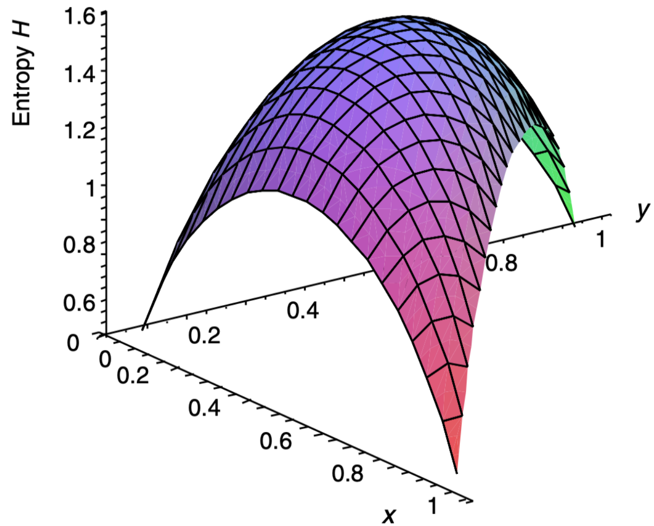
[Graph: Misclassification, Gini]

Impurity Function:

Impurity Functions Based on Entropy (continued)

Graph of the function $\iota_{entropy}(p_1, p_2, 1 - p_1 - p_2)$:





Impurity Function:

Impurity Functions Based on Entropy (continued)

Definition for k classes:

$$\iota_{entropy}(p_1, \ldots, p_k) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

$$\iota_{entropy}(D) = -\sum_{i=1}^{k} \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_i\}|}{|D|} \cdot \log_2 \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_i\}|}{|D|}$$

Impurity Function:

Impurity Functions Based on Entropy (continued)

$\Delta\iota_{entropy}$ corresponds to the information gain $H(\mathcal{A}) - H(\mathcal{A} \mid \mathcal{B})$:

$$\Delta\iota_{entropy} = \underbrace{\iota_{entropy}(D)}_{H(\mathcal{A})} \quad - \quad \underbrace{\sum_{j=1}^{s} \frac{|D_j|}{|D|} \cdot \iota_{entropy}(D_j)}_{H(\mathcal{A}|\mathcal{B})}$$

Impurity Function:
Impurity Functions Based on Entropy (continued)

$\Delta\iota_{entropy}$ corresponds to the information gain $H(\mathcal{A}) - H(\mathcal{A} \mid \mathcal{B})$:

$$\Delta\iota_{entropy} = \underbrace{\iota_{entropy}(D)}_{H(\mathcal{A})} - \underbrace{\sum_{j=1}^{s} \frac{|D_j|}{|D|} \cdot \iota_{entropy}(D_j)}_{H(\mathcal{A}|\mathcal{B})}$$

Legend:

- $\iota_{entropy}(D) = \iota_{entropy} P[A_1], \ldots, P[A_k])$
- $\iota_{entropy}(D_j) = \iota_{entropy}(P[A_1 \mid B_j], \ldots, P[A_k \mid B_j]),\ j = 1, \ldots, s$
- $\iota_{entropy}(p_1, \ldots, p_k) = -\sum_{i=1}^{k} p_i \cdot \log_2(p_i)$
- $\frac{|D_j|}{|D|} = P[B_j],\ j = 1, \ldots, s$
- $A_i,\ i = 1, \ldots, k$, denotes the event that $\mathbf{x} \in X(t)$ belongs to class y_i. The experiment $\mathcal{A}$ corresponds to the classification $y : X(t) \to Y$.
- $B_j,\ j = 1, \ldots, s$, denotes the event that $\mathbf{x} \in X(t)$ has value b_j for feature B. The experiment $\mathcal{B}$ corresponds to evaluating feature B and entails the following splitting:
  $X(t) = X(t_1) \cup \ldots \cup X(t_s) = \{\mathbf{x} \in X(t) : \mathbf{x}|_B = b_1\} \cup \ldots \cup \{\mathbf{x} \in X(t) : \mathbf{x}|_B = b_s\}$
- $P[A_i], P[B_j], P[A_i \mid B_j]$ are estimated as relative frequencies based on D.

Impurity Function:

Impurity Functions Based on the Gini Index

Definition for two classes:

$$\iota_{Gini}(p_1, p_2) = 1 - ({p_1}^2 + {p_2}^2) = 2 \cdot p_1 \cdot p_2$$

$$\iota_{Gini}(D) = 2 \cdot \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_1\}|}{|D|} \cdot \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_2\}|}{|D|}$$

Impurity Function:

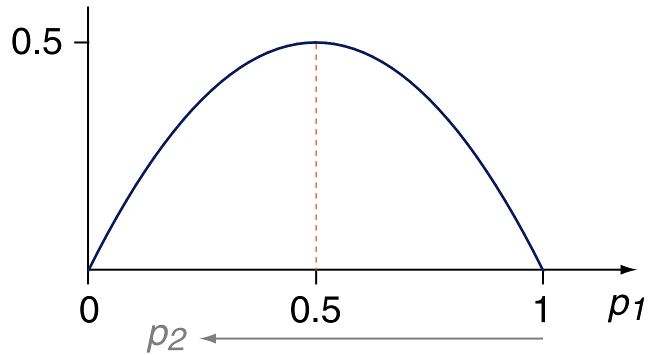
Impurity Functions Based on the Gini Index

Definition for two classes:

$$\iota_{Gini}(p_1, p_2) = 1 - ({p_1}^2 + {p_2}^2) = 2 \cdot p_1 \cdot p_2$$

$$\iota_{Gini}(D) = 2 \cdot \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_1\}|}{|D|} \cdot \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_2\}|}{|D|}$$

Graph of the function $\iota_{Gini}(p_1, 1 - p_1)$:



[Graph: Misclassification, Entropy]

Impurity Function:

Impurity Functions Based on the Gini Index (continued)

Definition for k classes:

$$\iota_{Gini}(p_1, \ldots, p_k) = 1 - \sum_{i=1}^{k} (p_i)^2$$

$$\iota_{Gini}(D) = \left(\sum_{i=1}^{k} \frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_i\}|}{|D|} \right)^2 - \sum_{i=1}^{k} \left(\frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_i\}|}{|D|} \right)^2$$

$$= 1 - \sum_{i=1}^{k} \left(\frac{|\{(\mathbf{x}, y(\mathbf{x})) \in D : y(\mathbf{x}) = y_i\}|}{|D|} \right)^2$$