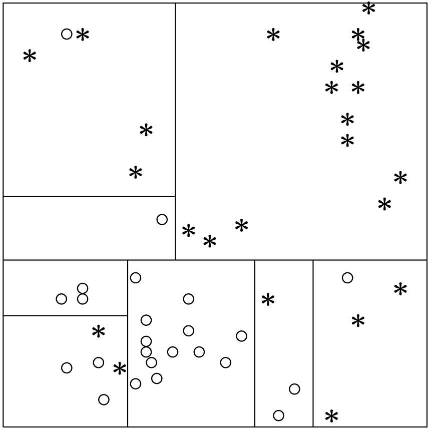
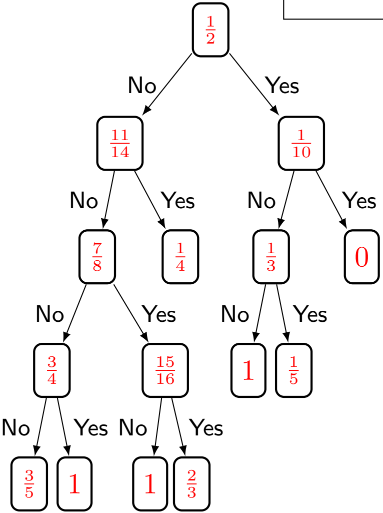
Decision and Regression trees

MATH-412 - Statistical Machine Learning

Decision tree





Decision/regression trees principle

- Is a local averaging method of the type histogram except that the partition $\Pi = \{R_1, \ldots, R_d\}$ is build from the data.
- Tree predictors are of the form :

$$f_{\boldsymbol{w}}(\mathbf{x}) = \sum_{j=1}^{d} w_j \, 1_{\{\mathbf{x} \in R_j\}}$$

where the (hyper)-rectangular regions R_j are obtained by recursive partitioning of the space based on splits that place a threshold on a single variable at a time.

Entropy associated with a loss function

If ℓ is a loss function, we define *the associated entropy for constant predictors* as $H_\ell(Y) = \inf_{a \in \mathcal{A}} \mathbb{E}[\ell(a, Y)]$.

Examples :

Regression. For $\ell(a, y) = (a - y)^2$, $\quad H_\ell(Y) = \inf_{a \in \mathbb{R}} \mathbb{E}[(a - Y)^2] = \mathrm{Var}(Y)$

Binary Classification. For $Y \sim \mathrm{Ber}(p)$,

- if $\ell(a, y) = (a - y)^2$, then $H_\ell(Y) = \mathrm{Var}(Y) = p(1 - p)$ is the *Gini entropy*
- if $\ell(a, y) = -\big[y \log a + (1 - y) \log(1 - a)\big]$, then we get the *Shannon entropy*

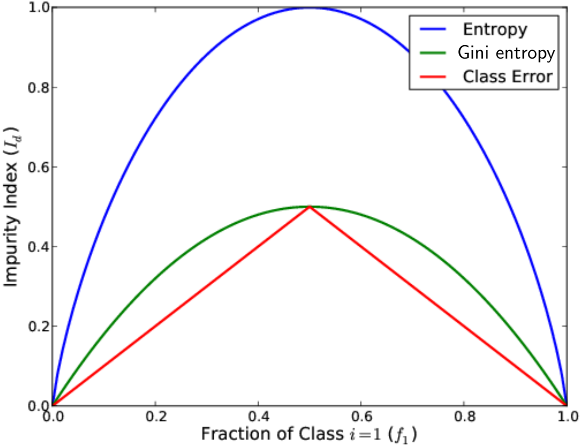
$$H_\ell(Y) = \min_{a \in [0,1]} -\mathbb{E}[Y \log a + (1 - Y) \log(1 - a)] = -p \log p - (1 - p) \log(1 - p)$$

- if $\ell(a, y) = 1_{\{a \neq y\}}$ then $H_\ell(Y) = \min_{a \in \{0,1\}} \mathbb{P}(Y \neq a) = \min_{a \in \{0,1\}} a(1 - p) + (1 - a)p$

  so that $H_\ell(Y) = \min(p, (1 - p))$ is the *(oracle) misclassification error*

These entropies are called impurity measures because $H_\ell(Y) \to 0$ when $p \to 0$ or $p \to 1$.

Impurity measures for binary classification



$$h_G(p) = p(1-p)$$

$$h_S(p) = -p\log p - (1-p)\log(1-p)$$

$$h_{\text{0-1}}(p) = \min\big(p, (1-p)\big)$$

Empirical impurity measures

The same impurity measures can be defined in an empirical setting

For least square *regression*, we have $\hat{\sigma}^2 = \min_a \frac{1}{n} \sum_i (y_i - a)^2$

For *binary classification*, $\hat{p} = \frac{1}{n} \sum_i y_i$.

We can define the Gini, Shannon and 0-1 entropies as :

$$h_G(\hat{p}) = \hat{p}(1-\hat{p}) = \min_a \frac{1}{n} \sum_i (y_i - a)^2$$

$$h_S(\hat{p}) = -\hat{p}\log\hat{p} - (1-\hat{p})\log(1-\hat{p}) = \min_a \frac{1}{n} \sum_i y_i \log a + (1-y_i)\log(1-a)$$

$$h_{\text{0-1}}(\hat{p}) = \min\big(\hat{p}, (1-\hat{p})\big) = \min_a \frac{1}{n} \sum_i 1_{\{y_i \neq a\}}$$

We denote generically

$$h_\ell(\hat{p}) = \min_a \frac{1}{n} \sum_i \ell(y_i, a) \qquad \text{for} \quad h_\ell \in \big\{h_G, h_S, h_{\text{0-1}}\big\}.$$

ERM on histograms in terms of the impurity measures

Let

- $\Pi = \{R_1, \dots, R_d\}$ and
- $\mathcal{F}_\Pi = \left\{ f_{\boldsymbol{w}} \mid f_{\boldsymbol{w}}(x) = \sum_{j=1}^d w_j \, 1_{\{x \in R_j\}} \right\}$ the histogram functions on Π

$$\forall f_{\boldsymbol{w}} \in \mathcal{F}_\Pi, \qquad \widehat{\mathcal{R}}_n(f_{\boldsymbol{w}}) = \frac{1}{n} \sum_{j=1}^d \sum_{i: x_i \in R_j} \ell(w_j, y_i)$$

$$\min_{f \in \mathcal{F}_\Pi} \widehat{\mathcal{R}}_n(f) = \frac{1}{n} \sum_{j=1}^d \min_{w_j} \sum_{i: x_i \in R_j} \ell(w_j, y_i) = \frac{1}{n} \sum_{j=1}^d n_j \, h_\ell(\hat{p}_j)$$

with $n_j = \sum_i 1_{\{x_i \in R_j\}}$ and $\hat{p}_j = \frac{1}{n_j} \sum_i y_i 1_{\{x_i \in R_j\}}$.

Impurity decrease via a split

- let $\Pi \;= \{R_1, \dots, R_{d-2}, R_{d-1}, R_d\}$
- and $\Pi_- = \{R_1, \dots, R_{d-2}, R_\cup\}$ with $R_\cup = R_{d-1} \cup R_d$

→ so that Π is obtained from Π_- by splitting $R_\cup$ into R_{d-1} and R_d

- let $\mathcal{F}_\Pi = \left\{ f_{\boldsymbol{w}} \mid f_{\boldsymbol{w}}(x) = \sum_{j=1}^d w_j \, 1_{\{x \in R_j\}} \right\}$ as before, and $\mathcal{F}_{\Pi_-}$ similarly.
- let $n_j = \sum_i 1_{\{x_i \in R_j\}}$ and $\hat{p}_j = \frac{1}{n_j} \sum_i y_i 1_{\{x_i \in R_j\}}$.

We have shown that

$$\min_{f \in \mathcal{F}_\Pi} \widehat{\mathcal{R}}_n(f) = \frac{1}{n} \sum_{j=1}^d n_j \, h_\ell(\hat{p}_j)$$

Let $\hat{f}_\Pi$ be the minimizer of $\widehat{\mathcal{R}}_n(f)$ in $\mathcal{F}_\Pi$, and likewise for $\hat{f}_{\Pi_-}$. Then the "decrease of impurity" due to the split is

$$\boxed{\widehat{\mathcal{R}}_n(\hat{f}_{\Pi_-}) - \widehat{\mathcal{R}}_n(\hat{f}_\Pi) = \frac{n_\cup}{n} h_\ell(\hat{p}_\cup) - \left[\frac{n_{d-1}}{n} h_\ell(\hat{p}_{d-1}) + \frac{n_d}{n} h_\ell(\hat{p}_d) \right]}$$

with

$$n_\cup = n_{d-1} + n_d \quad \text{and} \quad \hat{p}_\cup = \frac{n_{d-1}\, \hat{p}_{d-1} + n_d \, \hat{p}_d}{n_\cup}$$

Greedy decision tree learning algorithm

Given a training set $\big\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\big\}$ with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \{0,1\}$,

Algorithm 1 Decision tree building

```
Initialize $R_1$ a hyper-rectangle containing the data, $d \leftarrow 1$
while stopping criterion not met do
    for $j = 1$ to $d$, and $k = 1$ to $p$, do
        Let $x_{i_1,k} \leq \dots \leq x_{i_{n_j},k}$ be the sorted $(x_{i,k})_{i:\mathbf{x}_i \in R_j}$.
        for $s = 1$ to $n_j - 1$ do
            $\theta \leftarrow \frac{1}{2}\big(x_{i_s,k} + x_{i_{s+1},k}\big)$
            let $R_{j,k,\theta,-} = R_j \cap \{\mathbf{x} | x_k \leq \theta\}$, $R_{j,k,\theta,+} = R_j \cap \{\mathbf{x} | x_k > \theta\}$
            $\Delta H_{j,k,\theta} = n_j h_\ell(\hat{p}_j) - \big[n^-_{j,k,\theta}\, h_\ell(\hat{p}^-_{j,k,\theta}) + n^+_{j,k,\theta}\, h_\ell(\hat{p}^+_{j,k,\theta})\big]$
        end for
    end for
    $(j,k,\theta) = \mathrm{argmax}_{(j',k',\theta')} \Delta H_{j',k',\theta'}$
    $R_j \leftarrow R_{j,k,\theta,-}$,   $R_{d+1} \leftarrow R_{j,k,\theta,+}$, and $d \leftarrow d+1$
end while
```

with

$z_{i,j} = 1_{\{\mathbf{x}_i \in R_j\}}$

$n_j = \sum_i z_{i,j}$

$\hat{p}_j = \frac{1}{n_j} \sum_i y_i\, z_{i,j}$

$z_{i,j,k,\theta,-} = 1_{\{\mathbf{x}_i \in R_{j,k,\theta,-}\}}$

$n^-_{j,k,\theta} = \sum_i z_{i,j,k,\theta,-}$

$\hat{p}^-_{j,k,\theta} = \frac{\sum_i y_i\, z_{i,j,k,\theta,-}}{n^-_{j,k,\theta}}$

$z_{i,j,k,\theta,+} = 1_{\{\mathbf{x}_i \in R_{j,k,\theta,+}\}}$

$n^+_{j,k,\theta} = \sum_i z_{i,j,k,\theta,+}$

$\hat{p}^+_{j,k,\theta} = \frac{\sum_i y_i\, z_{i,j,k,\theta,+}}{n^+_{j,k,\theta}}$

Greedy *regression* tree learning for the *square loss*

Given a training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$,

Algorithm 2 Regression tree building

```
Initialize $R_1$ a hyper-rectangle containing the data, $d \leftarrow 1$
while stopping criterion not met do
    for $j = 1$ to $d$, and $k = 1$ to $p$, do
        Let $x_{i_1,k} \leq \ldots \leq x_{i_{n_j},k}$ be the sorted $(x_{i,k})_{i:\mathbf{x}_i \in R_j}$.
        for $s = 1$ to $n_j - 1$ do
            $\theta \leftarrow \frac{1}{2}\big(x_{i_s,k} + x_{i_{s+1},k}\big)$
            let $R_{j,k,\theta,-} = R_j \cap \{\mathbf{x} | x_k \leq \theta\}$, $R_{j,k,\theta,+} = R_j \cap \{\mathbf{x} | x_k > \theta\}$
            $\Delta H_{j,k,\theta} = n_j \hat{\sigma}_j^2 - \big[n_{jk\theta}^- \, \hat{\sigma}_{jk\theta-}^2 + n_{jk\theta}^+ \, \hat{\sigma}_{jk\theta+}^2\big]$
        end for
    end for
    $(j, k, \theta) = \text{argmax}_{(j',k',\theta')} \Delta H_{j',k',\theta'}$
    $R_j \leftarrow R_{j,k,\theta,-}$,   $R_{d+1} \leftarrow R_{j,k,\theta,+}$, and $d \leftarrow d+1$
end while
```

with

$$z_{i,j} = 1_{\{\mathbf{x}_i \in R_j\}}$$

$$n_j = \sum_i z_{i,j}$$

$$\hat{\mu}_j = \frac{1}{n_j} \sum_i y_i \, z_{i,j}$$

$$\hat{\sigma}_j^2 = \frac{1}{n_j} \sum_i z_{i,j} \, (y_i - \hat{\mu}_j)^2$$

$$z_{ijk\theta}^- = 1_{\{\mathbf{x}_i \in R_{j,k,\theta,-}\}}$$

$$n_{jk\theta}^- = \sum_i z_{ijk\theta}^-$$

$$\hat{\mu}_{jk\theta-} = \frac{\sum_i y_i \, z_{ijk\theta}^-}{n_{jk\theta}^-}$$

$$\hat{\sigma}_{jk\theta-}^2 = \frac{\sum_i z_{ijk\theta}^- (y_i - \hat{\mu}_{jk\theta-})^2}{n_{jk\theta}^-}$$

And similarly for

$n_{jk\theta}^+$, $\hat{\mu}_{jk\theta+}$, and $\hat{\sigma}_{jk\theta+}^2$

Impurity measures for multi-class classification

We consider a (one hot encoding) multinomial variable

$$Y \sim \text{Multi}\big((p_1, \ldots, p_K), 1\big)$$

$\mathcal{A}$	$\ell(\boldsymbol{a}, \boldsymbol{y})$	Impurity	Binary	Multiclass
$\mathbb{R}^K$	$\lVert \boldsymbol{a} - \boldsymbol{y} \rVert^2$	Gini entropy	$p(1-p)$	$\sum_{k=1}^K p_k(1-p_k)$
$\triangle$	$-\sum_{k=1}^K y_k \log a_k$	Shannon entropy	$-\log\left(p^p(1-p)^{1-p}\right)$	$-\sum_{k=1}^K p_k \log p_k$
$\dot{\triangle}$	$1_{\{\boldsymbol{a} \neq \boldsymbol{y}\}}$	Misclassification err.	$\min\left(p, (1-p)\right)$	$1 - \max_k p_k$

with

- the simplex : $\triangle = \{\boldsymbol{a} \in [0,1]^K \mid a_1 + \ldots + a_K = 1\}$
- the discrete simplex : $\dot{\triangle} = \triangle \cap \{0,1\}^K$

Tree Pruning

- One can stop splitting nodes when a minimal number of points per region is reached
- In addition, the tree is then pruned to minimize

$$\min_{\boldsymbol{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(f_{\boldsymbol{w}}(x_i), y_i) + \lambda d$$

- Pruning does not simply merge leaves in reverse order of appearance, because a poor split can be followed by a better split.

Weakest link pruning :

- Merge sibling leaf nodes that lead to the smallest possible increase of the empirical risk.
- Repeat this procedure iteratively
- Choose the best model by cross-validation.

Implementations and Criticisms

There are multiple variants of decision and regression trees. The algorithms presented correspond essentially to CART (Breiman et al., 1984). Other well known implementations include C4.5 (Quinlan, 1993).

If a region R_0 is split into $\{R_1, R_2\}$ with R_j having n_j points and class 1 proportion $\hat{p}_j$, then for the Shannon entropy, the decrease in impurity

$$\Delta H = n_0 \, h_\ell(\hat{p}_0) - \left[n_1 h_\ell(\hat{p}_1) + n_2 h_\ell(\hat{p}_2) \right]$$

can be overfitted...

- But it does not take into account significance/estimation uncertainty which is large for small nodes. This leads to the selection of irrelevant variables, which partially addressed by pruning but not completely.
- Since there are more possible splits for continuous variables and variables which have large number of levels, there is a bias in favor of these variables.

Other decisions and regression tree learning algorithm have tried to address these issues : QUEST (Loh and Shih, 1997), CRUISE (Kim and Loh, 2001), GUIDE (Loh, 2002), and *Conditional Inference trees* (Hothorn et al., 2006).

Conditional Inference Trees (Hothorn et al., 2006)

A tree is constructed by recursive splits as before except the choice of the splits are based on proper conditional independence tests.

1. At each leaf R_j a test of **independence** is performed between each variable X_k and Y (on the data in R_j) to test $H_0: X_k \perp\!\!\!\perp Y \mid X \in R_j$.

   A split is done on the variable X_k with the most significant test score, provided independence is rejected by the test.

2. Once the variable X_k chosen, the splitting threshold θ is chosen by performing again another independence test of $1_{\{X_k \le \theta\}}$ and Y inside R_j to reject the hull hypothesis

   $$H_0: 1_{\{X_k \le \theta\}} \perp\!\!\!\perp Y \mid X \in R_j,$$

   and the value of θ with the most significant rejection is selected.

The CI trees are implemented in the R-packages `party` (Hothorn et al., 2010) and `partykit` (Hothorn and Zeileis, 2015). These are included in the `caret` package (Kuhn et al., 2008).

References I

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Hothorn, T., Hornik, K., Strobl, C., and Zeileis, A. (2010). Party : A laboratory for recursive partytioning.

Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning : A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3) :651–674.

Hothorn, T. and Zeileis, A. (2015). partykit : A modular toolkit for recursive partytioning in R. *The Journal of Machine Learning Research*, 16(1) :3905–3909.

Kim, H. and Loh, W.-Y. (2001). Classification trees with unbiased multiway splits. *Journal of the American Statistical Association*, 96(454) :589–604.

Kuhn, M. et al. (2008). Building predictive models in R using the `caret` package. *Journal of statistical software*, 28(5) :1–26.

Loh, W.-Y. (2002). Regression tress with unbiased variable selection and interaction detection. *Statistica sinica*, pages 361–386.

Loh, W.-Y. and Shih, Y.-S. (1997). Split selection methods for classification trees. *Statistica sinica*, pages 815–840.

Quinlan, J. R. (1993). C4. 5 : Programs for machine learning.