Decision Trees

Decision Trees. Information Criteria.

Fourth Machine Learning in High Energy Physics Summer School,
MLHEP 2018, August 6—12

Alexey Artemov[1,2]

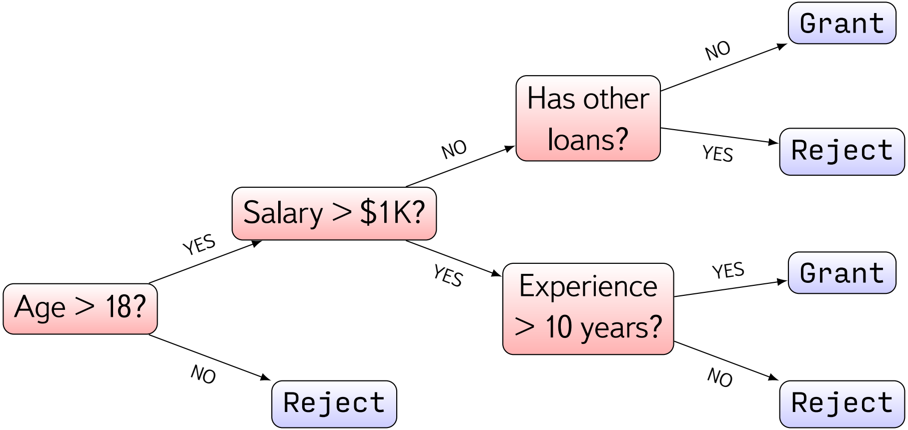
[1] Skoltech [2] National Research University Higher School of Economics

Lecture overview

- Decision Trees

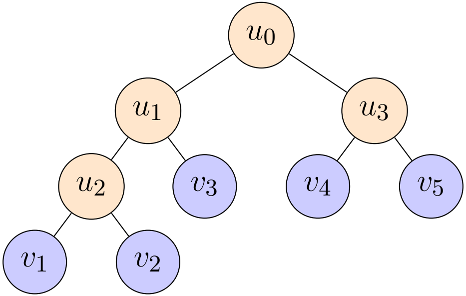
Decision trees

Decision making at a bank



Decision tree formalism

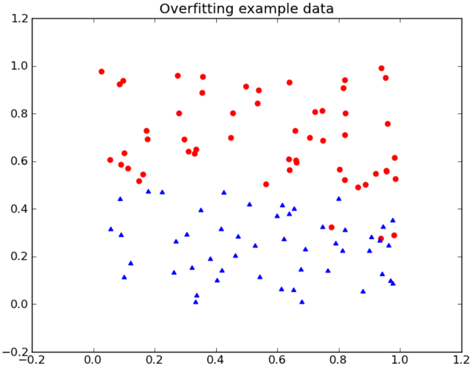
- Decision tree is a binary tree V
- Internal nodes $u \in V$: predicates $\beta_u : \mathbb{X} \to \{0, 1\}$
- Leafs $v \in V$: predictions x
- Algorithm $h(\mathbf{x})$ starts at $u = u_0$
  - Compute $b = \beta_u(\mathbf{x})$
  - If $b = 0$, $u \leftarrow \mathrm{LeftChild}(u)$
  - If $b = 1$, $u \leftarrow \mathrm{RightChild}(u)$
  - If u is a leaf, return b
- In practice: $\beta_u(\mathbf{x}; j, t) = [\mathbf{x}_j < t]$

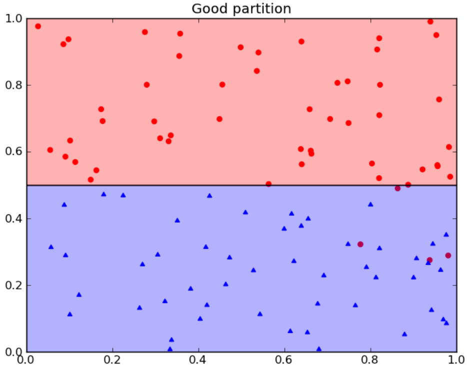


Greedy tree learning for binary classification

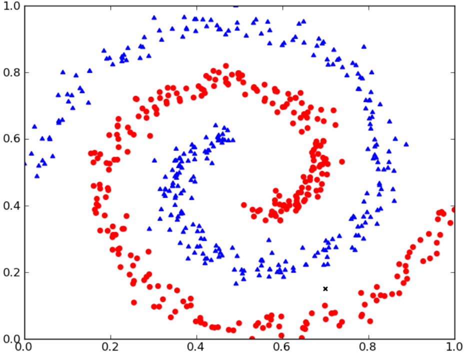
- Input: training set $X^\ell = \left\{(\mathbf{x}_i, y_i)\right\}_{i=1}^{\ell}$
  1. Greedily split X^ℓ into R_1 and R_2:
     $$R_1(j,t) = \{\mathbf{x} \in X^\ell | \mathbf{x}_j < t\}, \qquad R_2(j,t) = \{\mathbf{x} \in X^\ell | \mathbf{x}_j > t\}$$
     optimizing a given loss: $Q(X^\ell, j, t) \to \min\limits_{(j,t)}$
  2. Create internal node u corresponding to the predicate $[\mathbf{x}_j < t]$
  3. If a stopping criterion is satisfied for u,
     declare it a leaf, setting some $c_u \in \mathbb{Y}$ as leaf prediction
  4. If not, repeat 1–2 for $R_1(j,t)$ and $R_2(j,t)$
- Output: a decision tree V

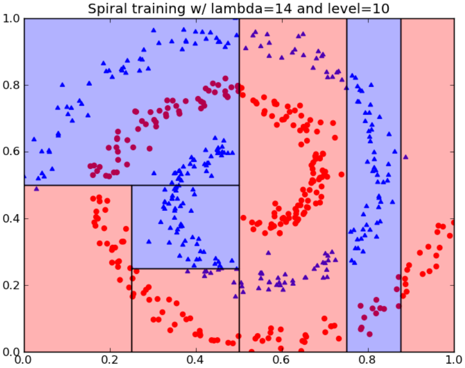
Greedy tree learning for binary classification





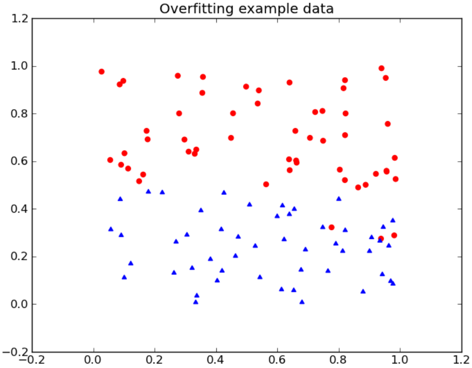
Greedy tree learning for binary classification

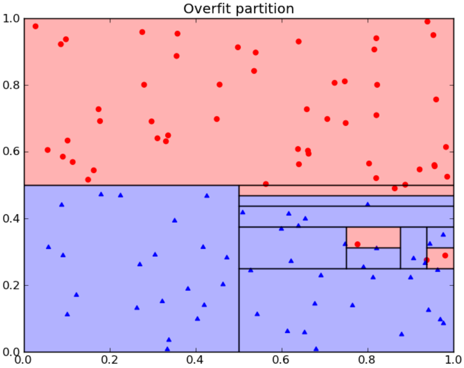




Underfitting

With decision trees, overfitting is extra-easy!





Design choices for learning a decision tree classifier

- Type of predicate in internal nodes
- The loss function $Q(X^\ell, j, t)$
- The stopping criterion
- Hacks: missing values, pruning, etc.

- CART, C4.5, ID3

The loss function $Q(X^\ell, j, t)$

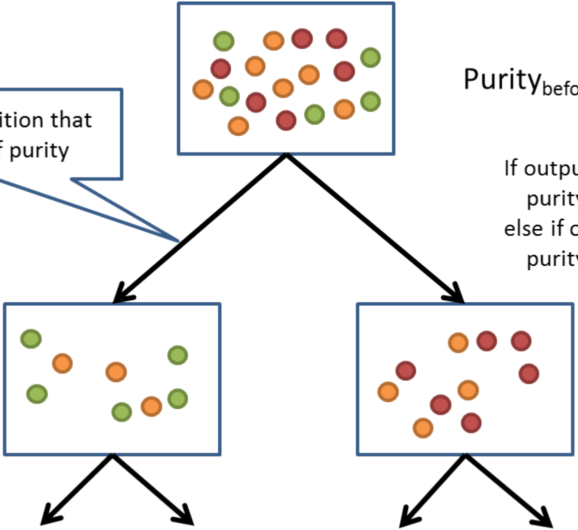
- R_m: the subset of X^ℓ at step m
- With the current split, let $R_l \subseteq R_m$ go left and $R_l \subseteq R_m$ go right
- Choose predicate to optimize

$$Q(R_m, j, t) = H(R_m) - \frac{|R_l|}{|R_m|} H(R_l) - \frac{|R_r|}{|R_m|} H(R_r) \to \max$$

- $H(R)$: impurity criterion
- Generally $H(R) = \min_{c \in \mathbb{Y}} \frac{1}{|R|} \sum_{(\mathbf{x}_i, y_i) \in R} L(y_i, c)$

The idea: maximize purity

Select split condition that maximize gain of purity



$\text{Purity}_{\text{before-split}}$

If output is numeric
purity = variance
else if output is categorical
purity = entropy or gini

$\text{Purity}_{\text{after-split}}$

Picture credit: https://dzone.com/refcardz/machine-learning-predictive

Examples of information criteria

Regression Wiki: In statistical modeling, regression analysis is a

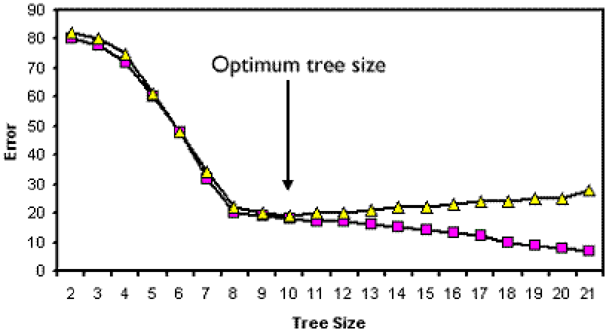
- Regression:
  - $H(R) = \min\limits_{c \in \mathbb{Y}} |R|^{-1} \sum_{(\mathbf{x}_i, y_i) \in R} (y_i - c)^2$
  - Sum of squared residuals minimized by $c = |R|^{-1} \sum_{(\mathbf{x}_j, y_j) \in R} y_j$
  - Impurity $\equiv$ variance of the target
- Classification:
  - Let $p_k = |R|^{-1} \sum_{(\mathbf{x}_i, y_i) \in R} [y_i = k]$ (share of y_i's equal to k)
  - Miss rate: $H(R) = \min\limits_{c \in \mathbb{Y}} |R|^{-1} \sum_{(\mathbf{x}_i, y_i) \in R} [y_i \neq c]$

Minimizing miss rate	$1 - p_{k_*}$,
Gini index	$\sum_{k=1}^{K} p_k (1 - p_k)$,
Cross-entropy	$-\sum_{k=1}^{K} p_k \log p_k$

Stopping rules for decision tree learning

- Significantly impacts learning performance
- Multiple choices available:
  - Maximum tree depth
  - Minimum number of objects in leaf
  - Maximum number of leafs in tree
  - Stop if all objects fall into same leaf
  - Constrain quality improvement
    (stop when improvement gains drop below s%)
- Typically selected via exhaustive search and cross-validation

Decision tree pruning



- Learn a large tree (effectively overfit the training set)
- Detect overfitting via K-fold cross-validation
- Optimize structure by removing least important nodes

Conclusion

- Decision trees: intuitive and interpretable, yet prone to overfitting