I2ML :: CHEAT SHEET

The **I2ML**: Introduction to Machine Learning course offers an introductory and applied overview of "supervised" Machine Learning. It is organized as a digital lecture.

Introduction to CART

CART - Classification And Regression Trees

- Divide feature space into sub-regions.
- Learn best constant prediction from training data for each region.

Advantages: Automatic feature selection; Fast and scales well larger data; Less preprocessing; Can model discontinuities and non-linearities.
Disadvantages: Linear dependencies; Predictions are step functions, never smooth; Empirically not the best; High instability (variance).

Binary Trees

- Represent a top-down hierarchy with binary split that contains: root node, internal nodes, and terminal nodes (leaves).
- Nodes have relative relationships: parent nodes and child nodes.
- Root nodes don't have parents – leaves don't have children.

Classification Trees - use the structure of a binary tree

- Binary splits are constructed top-down in a *data optimal* way.
- Each split is a threshold decision for a single feature.
- Each node contains the training points which follow its path.
- Each leaf contains a constant prediction.

Trees as an additive model:

Divide the feature space $\mathcal{X}$ with M leaf nodes into **rectangular regions**:

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m \mathbb{I}(x \in Q_m).$$

c_m: predicted numerical response, class label or class distribution in respective leaf node.

Growing a Tree

Recursively applying *greedy* optimization to each node $\mathcal{N}$. Greedy means **exhaustive search**: all possible splits of $\mathcal{N}$ on all possible points t for all features x_j are compared in terms of empirical risk $\mathcal{R}(\mathcal{N}, j, t)$. Training data is then distributed to child nodes according to the optimal split.

- Start with an empty tree, a root node contains all data.
- Search for feature and split point that minimizes the empirical risk in child nodes – makes label distribution more homogenous.
- Proceed recursively for each child node: select best split and divide data from parent node into left and right child nodes.
- Repeat until a stop criterion, e.g., until each leaf cannot be split.

Splitting Criteria

Use **empirical risk minimization**.
$\mathcal{N} \subseteq \mathcal{D}$: data contained in this node. $c_{\mathcal{N}}$: predicted constant for $\mathcal{N}$.
The risk $\mathcal{R}(\mathcal{N})$ for a node is: $\mathcal{R}(\mathcal{N}) = \sum_{(\mathbf{x},y)\in\mathcal{N}} L(y, c_{\mathcal{N}})$.
The optimal constant is: $c_{\mathcal{N}} = \arg\min_c \sum_{(\mathbf{x},y)\in\mathcal{N}} L(y, c)$
A split w.r.t. feature x_j at split point t divides a parent node $\mathcal{N}$ into

$$\mathcal{N}_1 = \{(\mathbf{x}, y) \in \mathcal{N} : x_j < t\} \text{ and } \mathcal{N}_2 = \{(\mathbf{x}, y) \in \mathcal{N} : x_j \geq t\}.$$

Finding the best way to split $\mathcal{N}$ into $\mathcal{N}_1, \mathcal{N}_2$ means solving

$$\arg\min_{j,t} \mathcal{R}(\mathcal{N}, j, t) = \arg\min_{j,t} \mathcal{R}(\mathcal{N}_1) + \mathcal{R}(\mathcal{N}_2)$$

If use averages $\frac{1}{|\mathcal{N}|}$, we have to reweight the terms to obtain a global average w.r.t. $\mathcal{N}$ as the children have different sizes

$$\bar{\mathcal{R}}(\mathcal{N}, j, t) = \frac{|\mathcal{N}_1|}{|\mathcal{N}|}\bar{\mathcal{R}}(\mathcal{N}_1) + \frac{|\mathcal{N}_2|}{|\mathcal{N}|}\bar{\mathcal{R}}(\mathcal{N}_2)$$

Splitting criteria

- Regression trees — L_2 loss.
- Classification trees — Brier score.

Minimize Brier score $\iff$ Minimize **Gini impurity**:

$$I(\mathcal{N}) = \sum_{k=1}^{g} \hat{\pi}_k^{(\mathcal{N})}(1 - \hat{\pi}_k^{(\mathcal{N})})$$

- Classification trees — Bernoulli loss.

Minimize Bernoulli loss $\iff$ Minimize **entropy impurity**:

$$I(\mathcal{N}) = -\sum_{k=1}^{g} \hat{\pi}_k^{(\mathcal{N})} \log \hat{\pi}_k^{(\mathcal{N})}$$

For classification, predicted probabilities in node $\mathcal{N}$ are the class proportions in the node:

$$\hat{\pi}_k^{(\mathcal{N})} = \frac{1}{|\mathcal{N}|} \sum_{(x,y)\in\mathcal{N}} \mathbb{I}(y = k)$$

Brier score and Bernoulli loss are more sensitive to changes in the node probabilities, and therefore often preferred than misclassification loss.

Split computation

Monotone feature transformations of feautures:

Will only change the numerical value of the split point.

Categorical Features

- A split on a categorical feature partitions the feature levels:

$$x_j \in \{a, c, e\} \leftarrow \mathcal{N} \rightarrow x_j \in \{b, d\}$$

- A feature with m levels results in about 2^m different possible binary partitions ($2^{m-1} - 1$ because of symmetry and empty groups).

$0-1$ responses, in each node:

- Calculate the proportion of 1-outcomes for each category.
- Sort the categories according to these proportions.
- Can then treat feature as ordinal, check at most $m - 1$ splits.

Continuous responses, in each node:

- Calculate the mean of the outcome in each category.
- Sort the categories by increasing mean of the outcome.

Missing Feature values:

Use **surrogate splits** to define replacement splitting rules, with a different feature that result in almost the same child nodes as original split.

Overfitting

Reduce overfitting:

- Use a less deep tree.
- Define different **stopping criteria**.
- **Pruning**: pre-pruing or post-pruning.

Cost-complexity pruning (CCP):

A post-prunig method to grow a large tree and remove the least informative leaves. CCP is steered with a regularization parameter α that penalizes the number of leaves in a sub tree

$$\mathcal{R}_{\text{reg}}(T) = \sum_{m=1}^{|T|} \sum_{i:x^{(i)}\in Q_m} L(y^{(i)}, c_m) + \alpha|T|,$$

$|T|$: number of leaves of sub tree T. Q_m: subset of the feature space, m-th terminal node. c_m: m-th node prediction. T_0: the complete tree.
CCP performs a greedy backward search:

- Computes $\mathcal{R}_{\text{reg}}(T)$ with a fixed α for all possible sub trees that can be created by replacing one internal node with a leaf.
- By replacing a node we also eliminate all subsequent nodes.
- Select the sub tree with lowest risk and repeat the procedure.
- Stop if pruning does not further reduce the risk.

Hyperparameter α is typically selected via cross-validation.