Decision Tree Models

Start w/ training data, labels

$x_i \in \mathbb{R}^n$, $i = 1 \ldots N$ data

$y \in \mathbb{R}^N$ labels.

No index, just concat. all labels into one big vector.

We want to partition the space of features S.T. the x_i's w/ the same y are grouped together.

I.e. each split ~~elt~~ element of $\mathbb{R}^n$

Let's represent the data @ some node m w/ Q_m, w/ n_m samples.

made up of a set of x_i, y labels

Then, represent each split as $\theta = (j, t_m)$

Feature (I.e. index of each x_i element)

Threshold value.

⇒ Partition data into left & right subsets:

$$Q_m^{left}(\theta) = \{(x, y) \mid x_j \leq t_m\}$$ (Below or equal to threshold)

$$Q_m^{right}(\theta) = Q_m \setminus Q_m^{left}(\theta)$$ (everything else)

Then, define a quality score based on a loss function:

$$G(Q_m, \theta) = \frac{n_m^{left}}{n_m} H(Q_m^{left}(\theta)) + \frac{n_m^{right}}{n_m} H(Q_m^{right}(\theta))$$

Then, ~~sel~~ select θ^* that min. G: $\theta^* = \operatorname{argmin}_\theta G(Q_m, \theta)$

To define H, define a measure for the proportion of class k observations in node m:

$$p_{mk} = \frac{1}{n_m} \sum_{y \in Q_m} I(y=k)$$

$\Rightarrow$ Gini loss : $H(Q_m) = \sum_k p_{mk}(1-p_{mk})$

Log loss : $H(Q_m) = -\sum_k p_{mk} \log(p_{mk})$

Random Forest: & XGBoost.

General idea: combine influence of several different decision tree models.

RF :

Build a forest where each tree uses 1) a random subset of features -> low correlation b/w trees.

Sklearn implementation - use all features or random subset of features. "Random subspace method"

2) Each tree is built from ~~x~~ a sample drawn w/ replacement from the training set.

Net effect: decrease variance of ensemble, reduce overfitting. Combine diverse trees, at cost of increasing model bias.

-> If training data is $[x_1, x_2, x_3, x_4, x_5, x_6]$,
one tree gets $[x_1, x_1, x_3, x_4, x_4, x_4, \cancel{x_6}]$. same size

Sampled with replacement

"Bagging" / "Bootstrapping"

Form a prediction w/ a consensus of trees.

XGi Boost

Additive trees: classifier & regressor trees
XG CART:

Group Members
Louis
Rob.
Sabari
Yeonjoon
Shree.

Interests

Likes GPUs
Yes → Sabari, Yeonjoon, Shree — Pred. score +2 — ML.
No. → Louis, Rob. — Pred score 0.5 — Organic

Runs XTB regularly
Yes → Louis, Rob, Yeonjoon — Pred Score + 1.5 — Organic
No. → Sabari, Shree — Pred score 0.3 — ML

f(Sabari) = 2 + 0.3 = 2.3.

Mathematically, $\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$, $f_k \in \mathcal{F}$ ⇒ $\mathcal{F}$ is set of all possible CARTS.
K ⇒ total number of trees.

$$\Rightarrow \min \cancel{obj(\theta)}\ obj = \underbrace{\sum_{i=1}^{N} l(y_i, \hat{y}_i)}_{\text{Loss fn. over data.}} + \underbrace{\sum_{i=1}^{K} \omega(f_k)}_{\text{Regularization over trees}}$$

↝ Controls tree complexity

③.

What are parameters of trees ⇒ what's "f"?

↳ Structure of tree + leaf scores!

This isn't a gradient optimization problem:

Learning on space of all possible trees is totally intractable!

Instead, additive predictions: Fix what we have learned, add one tree at a time.

$\cancel{\hat{y}_i(}$

$$\hat{y}_i^0 = 0$$

$$\hat{y}_i^1 = f_1(x_i) := \hat{y}_i^0 + f_1(x_i)$$

First CART tree.

$$\hat{y}_i^2 = \overset{f_1(x_i)+f_2(x_i)}{\cancel{f_2(x_i)}} := \hat{y}_i^{(1)} + f_2(x_i)$$

$$\vdots$$

$$y_i^{(t)} = \cancel{f} \sum_{k=1}^{t} f_k(x_i) := \hat{y}_i^{(t-1)} + f_t(x_i)$$

Into objective fn:

$$obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \omega(f_i)$$

$$= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \omega(f_t) + c$$

We can only optimize structure of t^{th} tree! Roll other ω's into const.

If we apply MSE as the loss fn:

$$obj^{(t)} = \sum_{i=1}^{n} (y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \omega(f_t) + c$$

$$\cancel{= y_i^2}$$

$$= \sum_{i=1}^{n} (\underbrace{\cancel{y_i^2}}_{\text{const}} - 2y_i(\underbrace{\cancel{\hat{y}_i^{t-1}}}_{\text{const}} + f_t(x_i)) + \underbrace{\cancel{(\hat{y}_i^{t-1})^2}}_{\text{const}} + 2\hat{y}_i^{(t-1)} f_t(x_i) + (f_t(x_i))^2)$$

$$+ \omega(f_t) + c$$

$$= \sum^{n} [2(\hat{y}_i^{(t-1)} - y) f_t(x_i) + (f_t(x_i))^2] + \omega(f_t) + c$$

$$obj^{(t)} = \sum_{t=1}^{n} \Big[2(\hat{y}^{(t-1)} - y_i) f_t(x_i) + (f_t(x_i))^2 \Big] + \omega(f_t) + c.$$

$2(\hat{y}^{(t-1)} - y_i) f_t(x_i)$: 1st order term (residual)

$(f_t(x_i))^2$: 2nd order term

We got a nice form in the case of MSE; but for other loss fns this isn't the case! So, in general, use Taylor expansion:

$$obj^{(t)} = \sum_{t=1}^{n} \Big[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \Big] + \omega(f_t) + c$$

where $g_i = \dfrac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$; $h_i = \dfrac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial y_i^{(t-1)}}$ } "gradient boosting"

Constants don't affect minimization procedure:

$$\Rightarrow obj^{(t)} = \sum_{t=1}^{n} \Big[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \Big] + \omega(f_t)$$

XGBoost defines $\omega(f)$ as:

$$\omega(f) = \underbrace{\gamma}_{\text{Gamma fn}} \underbrace{T}_{\text{Total num of trees}} + \frac{1}{2} \underbrace{\lambda}_{\text{hyperparam}} \sum_{j=1}^{T} \underbrace{\omega_j^2}_{\text{Vector of scores on each leaf}}$$

Rewrite & compress: $obj^{(t)} = \sum_{j=1}^{T} \Big[G_j \omega_j + \frac{1}{2}(H_j + \lambda)\omega_j^2 \Big] + \gamma T.$

SHAP Analysis

SHAP ⇒ "Shapley Additive Explanations"

Comes from cooperative game theory:

A number of players cooperate to achieve an objective, that leads to some overall gain.

But, some players may contribute more

One might have more bargaining power

Another might threaten to destroy everything

⇒ Question: what amount of the overall gain should be assigned to each player?

· Ie. how important is each player to the overall gain, and how much payoff should each player expect

John: Imagine the group being formed one person /feature at a time. Each person demands their contribution as "fair compensation".

Then, for each person, average contribution over the different ways we can form the group.

For ML: Assume only some features are present, while others aren't

>

Shapley values are only attribution method that satisfies:

1) Efficiency: Feature contributions sum to the diff of a prediction x and the average:

$$\sum_{j=1}^{p} \phi_j = \hat{f}(x) - \underbrace{E_X}_{\text{Expectation value.}}(\hat{f}(x))$$

2) Symmetry: Contributions of features j & k are the same if they contribute equally to all possible groups

If $val(S \cup \{j\}) = val(S \cup \{k\})$.

$$\forall\ S \subseteq \underbrace{\{1, \dots p\} \setminus \{j, k\}}_{\text{Group w/o } j,k.}$$

Then $\phi_j = \phi_k$.

3) Dummy: A feature j that doesn't change predicted value, regardless of group, gets a value of 0.

If $val(S \cup \{j\}) = val(S) \quad \forall\ S \subseteq \{1, \dots p\}$.

$$\phi_j = 0$$

Additivity: For a objective w/ combined payouts, ~~th~~ SHAP values are additive:

$$\phi_j + \phi_j^{+} \Rightarrow \text{~~th~~ combined value.}$$

This is important, since for RF/ensemble models, additivity guarantees that if we calc. SHAP val. for each indiv. tree & average $\Rightarrow$ SHAP for forest