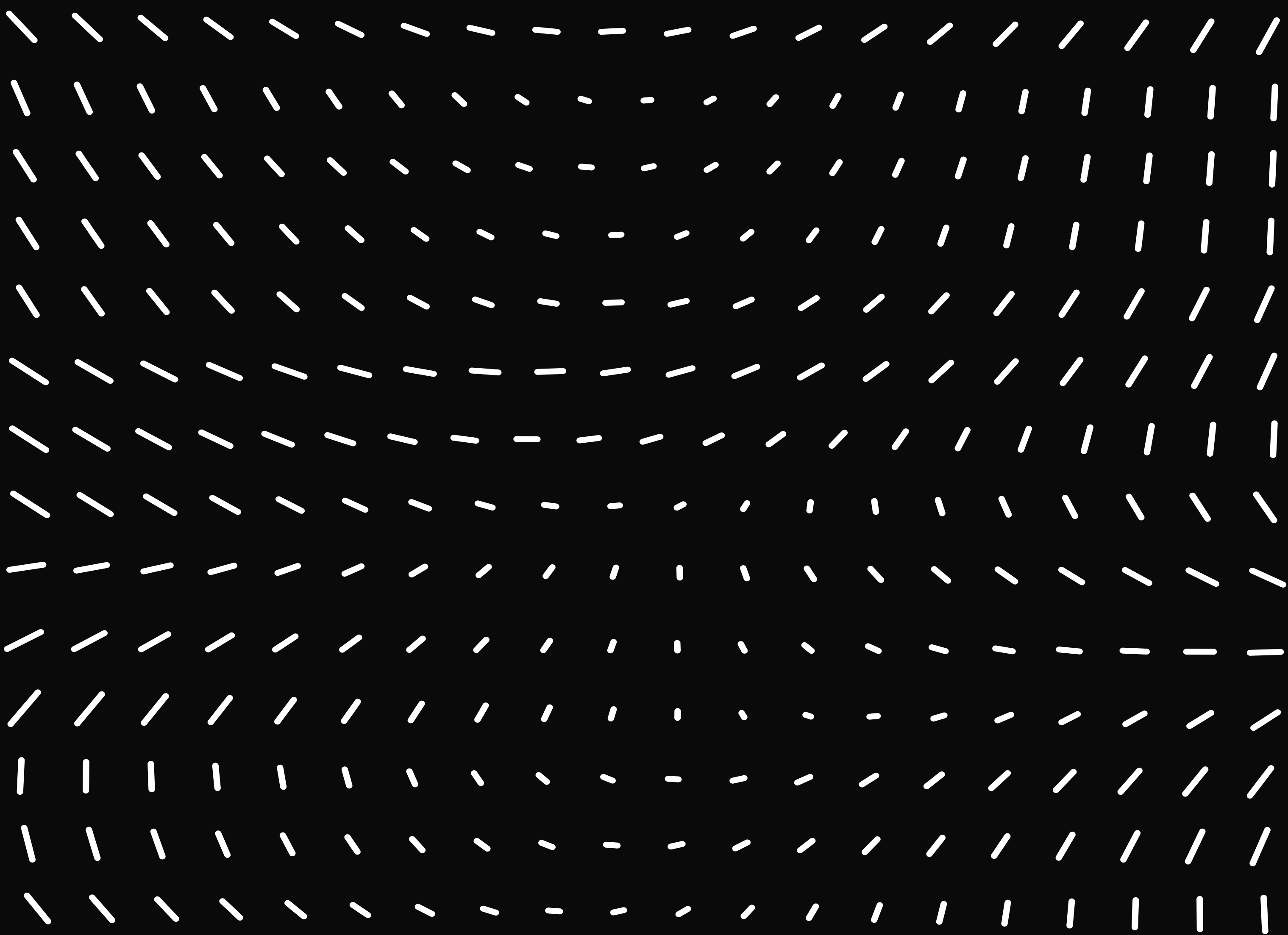
Additive Model & Decision Tree

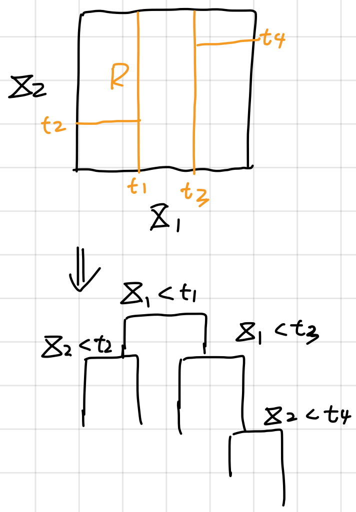


1. Generalized Additive Model

$$E(Y \mid X_1, X_2 \cdots X_p) = \alpha + f_1(X_1) + \cdots + f_p(X_p)$$

2. Decision Tree

$$f(X) = \sum_{m=1}^{M} C_m I\{(X_1, X_2) \in R_m\}$$



binary partition

2.1 given data (x_i, y_i)

$$x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})$$

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

$$\Downarrow$$

$$\hat{c}_m = avg(y_i \mid x_i \in R_m)$$

greedy algorithm

proof: given the data (x_i, y_i) and $\begin{cases} \text{splitting var. } j \\ \text{splitting point } s \end{cases}$

$$R_1(j,s) = \{X \mid X_j \le s\} \qquad R_2(j,s) = \{X \mid X_j > s\}$$

object: $\min_{j,s}\left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2\right]$

$$\Downarrow$$

$$c_1 = avg(y_i \mid x_i \in R_1(j,s)) \qquad c_2 = avg(y_i \mid x_i \in R_2(j,s))$$

repeat on R_1 and R_2.

2.2 Tree size.

strategy: grow large tree T_0. stop when minimum node size reached.

cost-complexity pruning: collapse any number of its internal (not terminal) nodes.

$|T|$: # of terminal nodes ($R_1 R_2 \cdots R_m$, index by m)

$T \subseteq T_0$ means T can be obtained by pruning T_0

N_m : # $\{x_i \in R_m\}$

$$\hat{c}_m = \frac{1}{N_m} \cdot \sum_{x_i \in R_m} y_i$$

$$Q_m(T) = \frac{1}{N_m} \cdot \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2$$

complexity criterion: $C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$

objective: for each α, find $T_\alpha \subseteq T_0$ that minimize $C_\alpha(T)$

2.2.1 * for each α, there is a unique smallest T_α that minimize $C_\alpha(T)$

weakest link pruning: □ successively collapse internal nodes that produces the smallest per-node increase in $\Sigma N_m Q_m(T)$

until produce a single node tree

- collect a sequence of subtrees, must contain T_α
- use CV to find the best $\hat{\alpha}$ minimize MSE

$\Downarrow$

$T_{\hat{\alpha}}$

2.3. CART : leaf only contain decision values

Tree for classification : (1, 2, ..., K categories)

$$\begin{cases} \text{in regression : } Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2 \\ \text{in classification : } \hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k) \end{cases}$$

← called node impurity

$$k(m) = \arg\max_k \hat{p}_{mk}$$

(the majority in node m)

$$Q_m(T) = \begin{cases} \text{misclassification error : (non differentiable)} \\ \quad \frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)} \\ \text{Gini index :} \\ \quad \sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk} (1 - \hat{p}_{mk}) \\ \text{Cross-entropy or deviance :} \\ \quad - \sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk} \end{cases}$$

- Gini index / cross-entropy used for T_0 tree growing
- misclassification rate use to guide cost-complexity pruning

3. Random Forest. Modification of bagging

trees { can capture complex interaction structure
if grown sufficiently deep, can have low bias
noisy, so will benefit great from average
each tree in RF is identical, so E(average) = E(itself)

3.1 Bagging for variance reduction

B i.d. trees with correlation ρ and variance σ^2: $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$

when $B \to \infty$, $\frac{1-\rho}{B}\sigma^2 \to 0$, only $\rho\sigma^2$ left

ρ will affect averaging B trees (bagging)

RF: reduce ρ while control σ^2

3.2. from bagging to RF in classification & regression

input at each split has p variables,
choose $m \leq p$ at random as candidates for splitting

also called feature bagging
reduce correlation between those strong predicting features

classification: obtain a class vote from each tree, then use majority
($m = \sqrt{p}$ and minimum node size is 1)

regression: average
($m = p/3$ and minimum node size is 5)

3.3 Out of bag samples

for each observation $z_i = (x_i, y_i)$, construct its RF predictor by averaging only those trees corresponding to bootstrap samples in which $z_i = (x_i, y_i)$ did not included.

(kind like CV, and it's built in so we only have to run it in one sequence)

3.4 variable importance:

Fit RF to data set, calculate out-of-bag error. Then for each feature j, permute it, calculate new out-of-bag error.

importance score = average dif of out-of-bag error → normalize

4. XGBoost : decision tree ensembles

consists of a set of CART : $f_k(x)$

Model formula:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F$$

K : number of trees
F : set of all possible CARTs

same as RF, dif is how we train them.

Objective function :

$$obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

4.1 Additive Training

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

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

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

$$\vdots$$

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

4.1.1 objective at step t :

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

$$= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + \underbrace{\sum_{k=1}^{t-1} \Omega(f_k)}_{\text{constant}}$$

if l is MSE ;

$$obj^{(t)} = \sum_{i=1}^{n} [(y_i - \hat{y}_i^{(t-1)}) - f_t(x_i)]^2 + \Omega(f_t) + \text{constant}$$

$$= \sum_{i=1}^{n} [\underbrace{(y_i - \hat{y}_i^{(t-1)})^2}_{\text{constant}} + f_t(x_i)^2 - 2(y_i - \hat{y}_i^{(t-1)}) f_t(x_i)] + \Omega(f_t) + \cdots$$

$$= \sum_{i=1}^{n} [f_t(x_i)^2 - 2(y_i - \hat{y}_i^{(t-1)}) f_t(x_i)] + \Omega(f_t) + \text{constant}$$

In general :

$$obj^{(t)} = \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \tfrac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + \text{constant}$$

$$\begin{cases} g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) & \text{first derivative} \\ h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) & \text{second derivative} \end{cases}$$

$$\vdots$$

$$= \underbrace{\sum_{i=1}^{n} [g_i f_t(x_i) + \tfrac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)}_{\text{this made XGBoost possible to support custom loss function}}$$

4.1.2. regularization term $\Omega(f_t)$

$$f_t(x) = w_{q(x)}, \quad w \in R^T, \quad q \in R^d \to \{1, 2, 3, \cdots, T\}$$

$\begin{cases} w: \text{ vector of scores on leaves} \\ T: \text{ number of leaves} \\ q: \text{ function that assign each data point } (R^d) \to \text{ a leaf} \end{cases}$

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

4.1.3. structure score

$$obj^{(t)} = \sum_{i=1}^{n} [g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

$$= \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T$$

$I_j = I\{ i \mid q(x_i) = j \}$ set of indices of data point assign to leaf j

$$= \sum_{j=1}^{T} [G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2] + \gamma T$$

$\begin{cases} G_j = \sum_{i \in I_j} g_i \\ H_j = \sum_{i \in I_j} h_i \end{cases}$

w_j are independent to each other, by solve the $\min\{obj^{(t)}\}$

we get $\begin{cases} w_j^* = -\dfrac{G_j}{H_j + \lambda} \\ obj^{(t)*} = -\dfrac{1}{2} \sum_{j=1}^{T} \dfrac{G_j^2}{H_j + \lambda} + \gamma T \end{cases}$ ← measures how good a tree is

when trying to creat a split of a leaf into L and R

$$\Delta obj = \gamma - \underbrace{\frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right]}$$

if it's smaller than γ, then obj ↑, no need to add this split (aka pruning)