Q→ Guess the word ?

Agenda :

1. Bagging Cont.
2. Grid Search.
3. Boosting Intuition.
4. Boosting Mathematics
5. Gradient Boosting

Feature Importance.

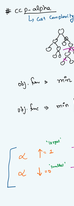
CC & α-alpha.

↳ Cost Complexity Pruning.

a) for → min [Loss + α · weight] — Low Cost Complexity

b) for → min [Loss (weight) + α · (# leaves nodes)]

α = 0 → Tree



Cons:

RF: DT + RS + CC Pros.

↳ large dim., not linear or separable, Outliers, Imbalanced

1. Feature Importance.
2. Out of Bag - Validation → K-Fold → Best learner

Extremely Randomized Trees

↳ Extra-Trees Classifier
↳ Extra-Trees Regressor

Extra-Trees = DT - RS + CS + randomization on "K."
+ Fast

Grid Search → gridsearcv

↳ for automated hyper-parameter tuning.

Est = [svc, knn, DT]
par_grid = [K, P, M]

⇒ cv × 4 × 16 Combinations
"cross" to 80 models

→ "Exhaustive"
→ "Expensive"

"Best Estimator"

} 3 Est.

⇒ 4 × 3

Random Search CV

"Pill Combination"

Use → Plan → hyper-Tuner

cv
↳ 4 × 4 × 5 × 2 × 6 ⇒

Meet the Tuning

Max-depth = [10, 20, None]
Max-depth = [Low, Med, High]
[10, 20, ∞]
[Low, Med, High]
→ [Low, Med, ∞]

Max-depth
10 20 None
→ [Low, Med, None]

Boosting

Bagging : models → Parallelization (cpu)
[Bagging + Boosting] → Reduce var.

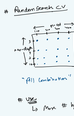
Boosting : Base Learner [high Bias + low var] (weak DT) → Reduce the bias, training slow for.

Sequential → Cost Complexity

Cost Idea : "Reduce the Bias"

Strategy : "Trees → Regression"

Trees → { $x^{(1)}, y^{(1)}$ ... $x^{(m)}, y^{(m)}$ }



Step 1 ⑥ D-set → $h_1(x)$ high Bias, low variance } e.g. DT → depth DT → depth

h_1 → $\{y - h_1(x)\}$ → new Model

"leaky back cover"

⑥ errors / residuals

$x^{(1)}, \overset{h_1(x^{(1)})}{y^{(1)}}$ [Target diff error]

for all data point. $\{x^{(1)}, y^{(1)} - h_1(x^{(1)})\}_{D^1}$

Step 2 $\{x^{(1)}, \overset{y^{(1)} - h_1(x^{(1)})}{y^{(1)}}\}_{D^1}$ → $h_2(x)$ D-set for data residuals

$M_1 = h_1(x)$

$F_1(x)$ ⇒ Model at the end of step 1

$F_1(x)$ ⇒ $h_1(x) + \alpha_1 h_2(x)$
"weight"

Step 3 $\{x^{(1)}, \overset{y^{(1)} - h_1(x^{(1)})}{y^{(1)}}\}_{D^1}$ → $h_3(x)$

$y^{(1)} - h_1(x^{(1)})$ → $h_3(x)$ D-set target data

$F_2(x)$: $h_1(x) + \alpha_1 h_2(x) + \alpha_2 h_3(x)$

Step n :

$F_m(x) = h_1(x) + \alpha_1 h_2(x) + \alpha_2 h_3(x) \ldots + \alpha_n h_n(x)$

$F_n(x)$ ⇒ $h_1 + \sum_{i=1}^{n} \alpha_i \cdot h_i(x)$

Geometric Intuition.

$\hat{y} = \sum_{i=1}^{n} \alpha_i h_i(x)$
single
Base
sub learner

Linear Reg. Vs. Gradient Boosted Decision Tree.

Q→ What are we learning?

M-Trees = 3

$F_3(x) = h_0(x) + \alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x)$

$\alpha_1 = 0.8$ $\alpha_2 = 0.3$ $\alpha_3 = 0.2$

Q→ What happens at test-time ?

A→ $\hat{y} = \sum_{i=1}^{n} \alpha_i h_i(x)$

$\hat{y} = 1.3 + 0.8 \times 1.8 + 0.3 \times 1.25 + 0.2 \times 0.85$