Bagging + Boosting

Bagging: Bootstrap Aggregating

Bootstrap sampling

$X_i \sim N(\mu, \sigma^2)$. Goal: Est μ

$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$, $E[\hat{\mu}] = \mu$. $Var[\hat{\mu}] = Var[\frac{1}{n}\sum X_i] = \frac{1}{n^2} Var[\sum X_i]$

$= \frac{1}{n^2} \cdot n \cdot Var[X_1]$

$= \frac{1}{n} \cdot \sigma^2 = \frac{\sigma^2}{n}$

Var too large...

Say we have Bn points $\sim N(\mu, \sigma^2)$

$S_1 = \{X_1, \ldots, X_n\}$, $S_2 = \{X_{n+1}, \ldots, X_{2n}\}$, $\ldots$, $S_B = \{X_{(B-1)n+1}, \ldots, X_{Bn}\}$

$\hat{\mu}^{(j)} = \frac{1}{n}\sum_{z \in S_j} z$.

$\hat{\mu}^{(avg)} = \frac{1}{B}\sum \hat{\mu}^{(j)}$. $E[\hat{\mu}^{(avg)}] = \mu$ $Var[\frac{1}{B}\sum \hat{\mu}^{(j)}] = \frac{1}{B^2} \cdot B \cdot \frac{\sigma^2}{n} = \frac{\sigma^2}{nB}$

Needs $B\times$ more data

Bootstrap sampling $\approx$ sampling w/ replacement

Dataset X_1, X_2, X_3, X_4, X_5

$S_1 = \{X_3, X_4, X_1, X_1, X_4\}$

$S_2 = \{X_5, X_5, X_3, X_1, X_2\}$

$\vdots$

$S_B = \ldots$

$\hat{\mu}^{(j)} = \frac{1}{n}\sum_{z \in S_j} z$

$\hat{\mu}^{(avg)} = \frac{1}{B}\sum_{j=1}^{B} \hat{\mu}^{(j)}$

$E[\hat{\mu}^{(avg)}] = \mu$.

Var = ?

Not independent

Still works!

Application to ML

Some approaches: high variance

Decision trees. Running DT on two splits of same dataset → different trees.

Use bagging.

1. Bootstrap sample B datasets from training set $S_1, \ldots, S_B$
2. Run learning algo on each. $\hat{f}^{(1)}, \ldots, \hat{f}^{(B)}$
3. Aggregate $\hat{f}^{(1)}, \ldots, \hat{f}^{(B)}$. How? Regression: $\hat{f}(x) = \frac{1}{B}\sum \hat{f}^{(j)}(x)$

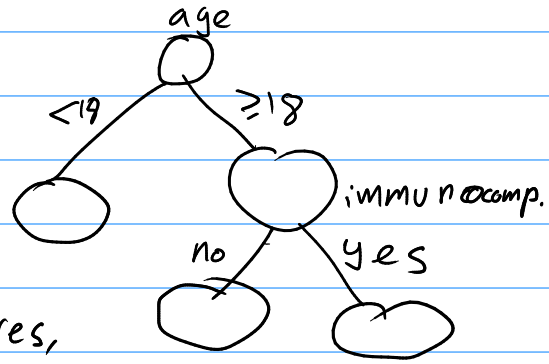
   Classification: $\hat{f}(x)$ = majority vote of $\hat{f}^{(j)}(x)$

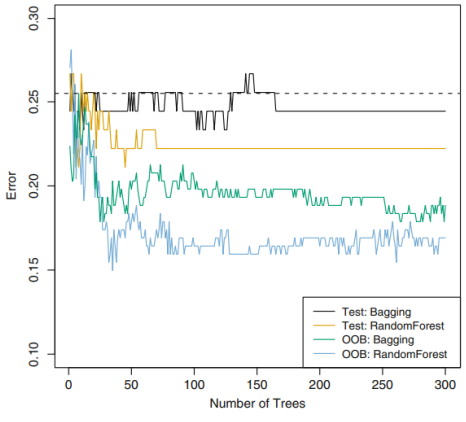
Random Forests

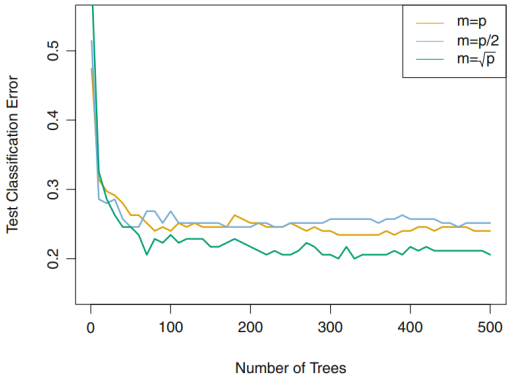
Bagging on DTs - Correlations (some)

Classic: Look at all d features, pick "best"

RFs: Choose random $m \approx \sqrt{d}$ features, split on "best"





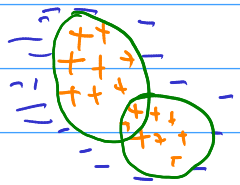


Boosting

Weak Learner: ~55% accurate (better than rand guess)

⇓

Strong Learner: "~90%+ accurate"



Online Learning with Experts

Example: Horse racing. T races. n horses.

T rounds. At round t, algo picks weights $p_1^{(t)}, \dots, p_n^{(t)}$ s.t. $\sum_{i=1}^{n} p_i^{(t)} = 1$. Algo experiences "loss" at time t: $\langle p^{(t)}, \ell^{(t)} \rangle$

↑ adversarially picked loss $[0,1]^n$

Goal: $\min \sum_{t=1}^{T} \langle p^{(t)}, \ell^{(t)} \rangle$

Benchmark: $\min_{i=1,\dots,n} \sum_{t=1}^{T} \ell_i^{(t)}$ $\quad p^{(t)} = 1$ in coord $\operatorname{arg\,min}_i \sum_{t=1}^{T} \ell_i^{(t)}$ $\forall t$

Hedge(β) $\quad \beta \in [0,1]$

Set $w^{(1)} = [1/n, \dots, 1/n] \in \mathbb{R}^n$

For $t = 1, \dots, T$:

Set $p^{(t)} = \frac{w^{(t)}}{\sum_{i=1}^{n} w_i^{(t)}}$ (Normalize)

Receive loss $\langle p^{(t)}, \ell^{(t)} \rangle$

Update: $w_i^{(t+1)} = w_i^{(t)} \beta^{\ell_i^{(t)}}$ (MWU)

$$\underbrace{\sum_{t=1}^{T} \langle p^{(t)}, \ell^{(t)} \rangle}_{\text{total loss}} \leq \frac{\log n + \log(1/\beta) \min_{i=1,\dots,n} \sum_{t=1}^{T} \ell_i^{(t)}}{1-\beta}$$

Best option in hindsight

Set β:

$$\underbrace{\frac{1}{T}\sum_{t=1}^{T} \langle p^{(t)}, l^{(t)} \rangle}_{\text{average loss}} \leq \underbrace{\frac{1}{T} \min_{i=1,\dots,n} \sum_{t=1}^{T} l_i^{(t)}}_{\text{avg best loss in hindsight}} + \underbrace{O\left(\sqrt{\frac{\log n}{T}}\right)}_{\text{"regret"}} \xrightarrow{\text{as } T \to \infty} 0$$

AdaBoost

Given algo WeakLearn that gets $>51\%$ accuracy on a training set. Can we boost this to high accuracy?

Wrong way: algorithms are "horses" (experts)
"Put large weight on good algos"

Right way: Datapoints are experts
Large weight → still must learn point.

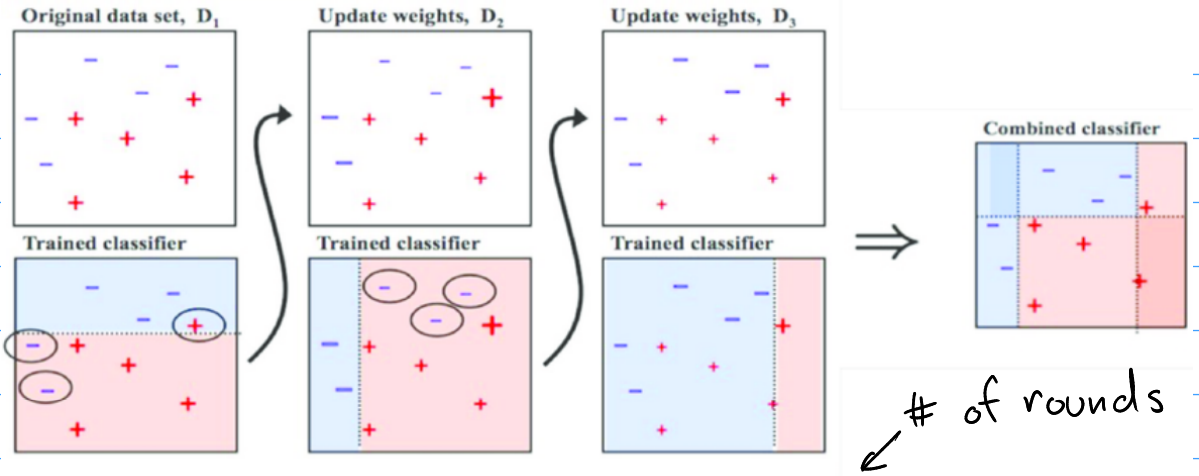
Adaboost:

Set $w^{(1)} = [1/n, \dots, 1/n]$
For $t = 1$ to T
  Set $p^{(t)} = \frac{w^{(t)}}{\sum_{i=1} w_i^{(t)}}$
  Run WeakLearn on training set (with weights $p^{(t)}$)
    ↳ Get classifier $h^{(t)} : X \to [0,1]$
  Calculate error: $\varepsilon_t = \sum_{i=1}^{n} p_i^{(t)} |h^{(t)}(x_i) - y_i|$ // $\varepsilon_t \leq 0.5$
  (avg. weighted error on pt i)

ε_t small → β_t small → reduce weights a lot

  $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$
  If $\varepsilon_t \leq \frac{1}{2}$
    Set $w_i^{(t+1)} = w_i^{(t)} \beta_t^{1 - |h^{(t)}(x_i) - y_i|}$

$$h(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \log\left(\frac{1}{\beta_t}\right) h^{(t)}(x) \geq \frac{1}{2} \sum_{t=1}^{T} \log\left(\frac{1}{\beta_t}\right) \\ 0 & \text{else} \end{cases}$$



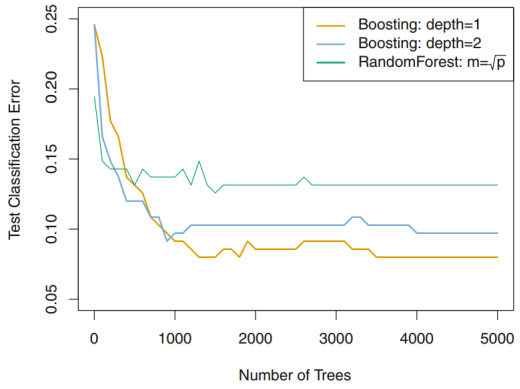
Training error: $\frac{1}{n}\sum_{i=1}^{n} 1\{h(x_i) \neq y_i\} \leq 2^T \prod_{t=1}^{T} \sqrt{\varepsilon_t(1-\varepsilon_t)}$ (# of rounds)

($\varepsilon_t < \frac{1}{2}$)

Say $\varepsilon_t \leq \frac{1}{2} - \gamma$ . . $\leq \exp(-2T\gamma^2)$.

Want: Good test error!

Generalize well w simple base classifier.



Generic + Flexible

Less interpretable.

Bagging - Parallel

Boosting - Sequential