# So-far

**Generative**

- Naive - Bayes
- G.D.A / G.N.B

**Discriminative**

- K-NN
- Decision trees
- Logistic regression
- Perceptron
- Support-vector machines.

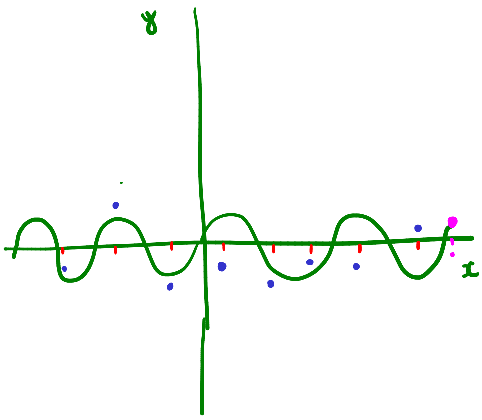
Goal :- Meta Classifiers (or) Ensemble Classifiers.

WEAK LEARNERS ⟶ STRONG LEARNERS.
(better than random)

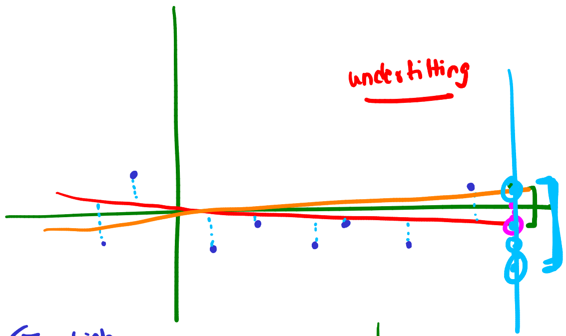
---

Overfitting - fit noise

Underfitting - missing out on structure
Thinking it is noise.

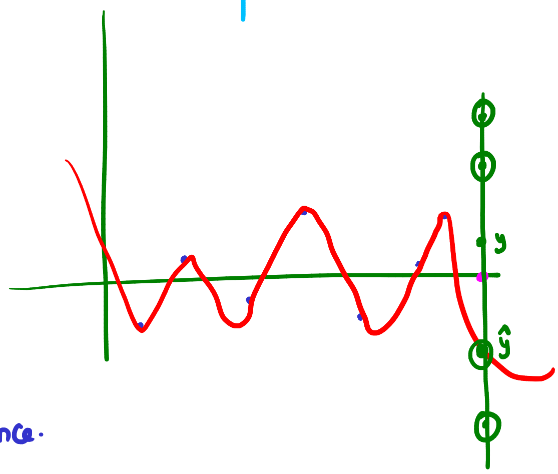



underfitting



← high bias

overfitting ← high variance


Error = bias + variance.

# BAGGING - BOOTSTRAP AGGREGATION.

$\{x_1, x_2, \ldots, x_n\} \sim \mathcal{N}(\mu, 1)$

$\hat{\mu}_1 = x_1$

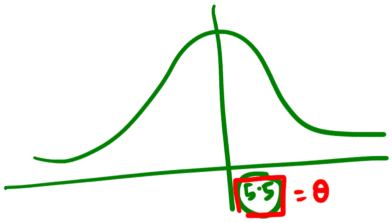
$\hat{\mu}_2 = x_2$

$\vdots$

$\hat{\mu}_n = x_n$

$\hat{\mu}_{ML} = \frac{1}{n}\sum_{i=1}^{n} x_i$

$E[\hat{\mu}_{ML}] = \mu$



$\boxed{5.5} = \theta$

$\{5, 7.3, 6.7, 8.2, \ldots, 9\}$

$\hat{\mu}_1 = 5$

$\hat{\mu}_2 = 7.3$

$\vdots$

$\hat{\mu}_n = 9$

$\boxed{E[\hat{\theta}] = \theta}$ ← unbiased estimators

$E[\hat{\mu}_1] = E[x_1] = \mu$

"Averaging reduces variance"

## Bagging

$D_1, D_2, \ldots, D_m$ — Each $D_i$ has $n$ datapoints.

$D_1 \to$ D.T $\to h_1$, $D_2 \to$ D.T $\to h_2$, $\ldots$, $D_m \to$ D.T $\to h_m$

$h_i : \mathbb{R}^d \to \{\pm 1\}$

$D_1 = \{x_1', x_2', \ldots, x_n'\}$
$\quad\;\; y_1', y_2', \ldots, y_n'$

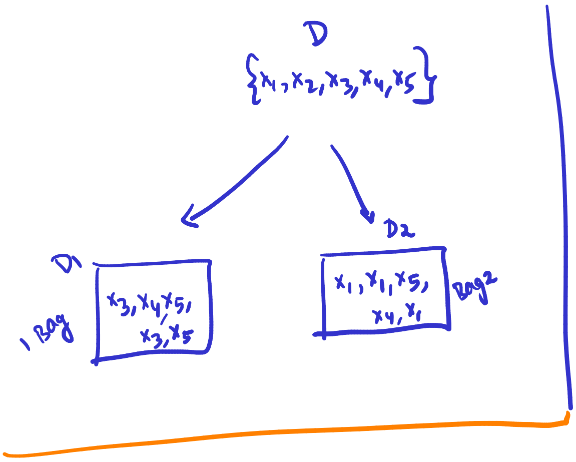
$x_i' \in \mathbb{R}^d, \; y_i' \in \{\pm 1\}$

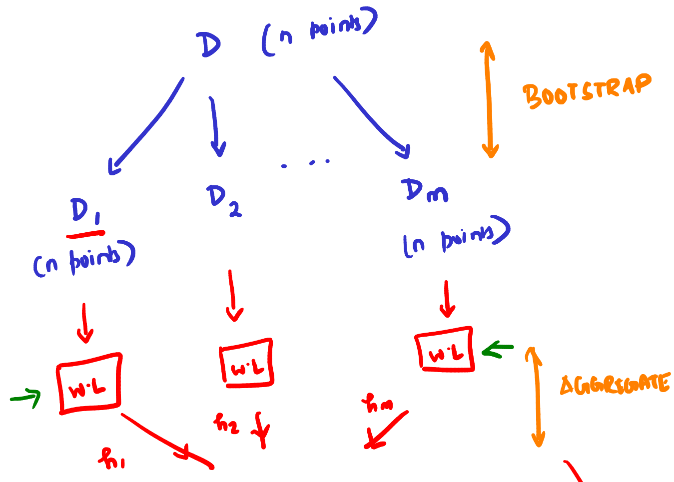
$$h^*(x) = \text{sign}\left(\frac{1}{m}\sum_{i=1}^{m} h_i(x)\right)$$

$\text{sign}(z) = +1$ if $z \geq 0$
$\quad = -1$ o/w.

Input $\quad D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ $\quad x_i \in \mathbb{R}^d$, $y_i \in \{\pm 1\}$

Bootstrapping → Sampling with replacement



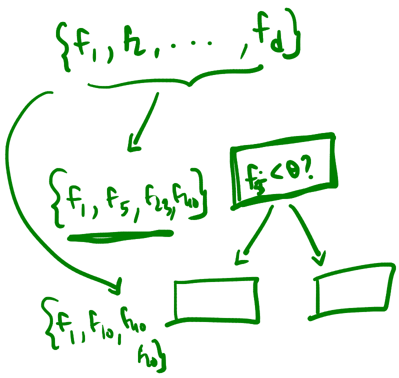

$$\hat{f}^*(x) = \operatorname{sign}\left(\frac{1}{m}\sum_{i=1}^{m} \hat{f}_i(x)\right)$$

$$Pr(x_j \in D_i) = 1 - \underbrace{\left(1 - \frac{1}{n}\right)^n} \approx 1 - \frac{1}{e}$$

$\approx$ 67% for large n

Bagging reduces variance!

## Random Forest



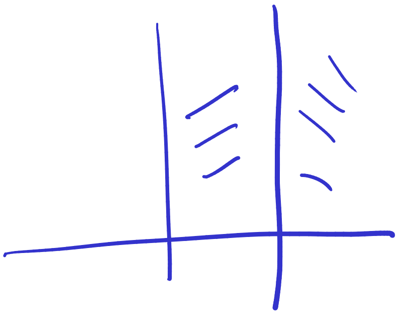

- Bags Decision trees (typically overfit trees)
- Feature bagging ($\sqrt{d}$)

- Bootstrapping – Sampling with replacement (uniform)
- Agg – Averaging

## BOOSTING

BOOSTING [ADABOOST]

L FREUND & SCHAPHIRE



Weak Learner $\longrightarrow$ Strong Learner

Decision Stumps

$\boxed{f_t < \theta}$

$+1$ $\quad$ $-1$

1-Level or 2-Level

---

ADA-BOOST ALGORITHM

↳ Adaptive.

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ $\quad x_i \in \mathbb{R}^d$, $y_i \in \{\pm 1\}$

Initialize $D_0(i) = \frac{1}{n}$

iteration

for $t = 1, \ldots, T$

- $h_t$ = Input $(S, D_t)$ to a weak learner to get $h_t$

$h_t : \mathbb{R}^d \to \{\pm 1\}$

- $$\tilde{D}_{t+1}(i) = \begin{cases} D_t(i) \cdot e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \\ D_t(i) \cdot e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \end{cases}$$

- $$D_{t+1}(i) = \frac{\tilde{D}_{t+1}(i)}{\sum_j \tilde{D}_{t+1}(j)}$$

end

$$\begin{array}{cccc} & x_1 & x_2 & x_3 \\ D_0 & [0.33 & 0.33 & 0.33] \\ D_1 & [0.3 & 0.5 & 0.2] \\ D_2 & \left[\frac{0.3 \times e^{\alpha_1}}{z}\right. & \frac{0.5\, e^{-\alpha_1}}{z} & \left.\frac{0.2\, e^{-\alpha_1}}{z}\right] \end{array}$$

$\rightarrow h_1$

✗ ✓

$z = 0.3e^{\alpha_1} + 0.5e^{-\alpha_1} + 0.2e^{-\alpha_1}$

$h_1, h_2, \ldots, h_T$

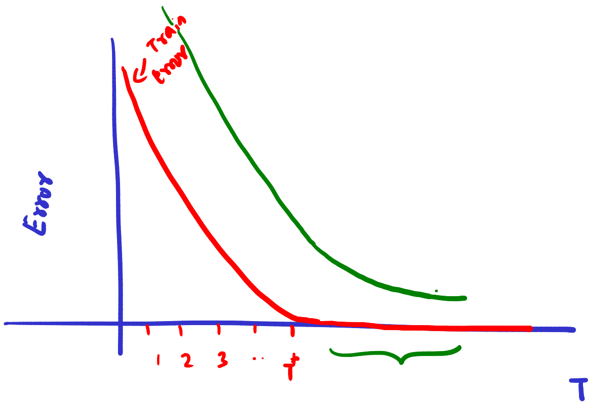
$$h_T^*(x) = \mathrm{Sign}\left(\sum_{t=1}^{T} \alpha_t\, h_t(x)\right)$$

$$\alpha_t = \ln \sqrt{\frac{1 - \mathrm{err}(h_t)}{\mathrm{err}(h_t)}}$$

One can prove

If $T \geq \dfrac{1}{2\gamma^2} \ln(2n)$, then Training error = 0

How good is my weak learner?




- Can't run in parallel