Main Idea

- Weak/Simple Learners (NB, LR, Linear SVMs, decision stump)
  - Underfit
  - High Bias
  - Low Variance

- Strong Learners (Kernel SVMs, Neural Nets, Decision Trees)
  - Overfit
  - Low Bias
  - High Variance

Ensemble Methods: Can we get the best of both?

- Bagging (multiple)
  - Reducing the Variance of Strong Learners

- Boosting (multiple)
  - Reducing the Bias of Weak Learners
Bagging = Bootstrap Averaging

Train - Time

\[ D = \{ (x_1, y_1), \ldots, (x_N, y_N) \} \]

Test - Time

\[ 2 \]

I = \{ 1, 2, 3, \ldots, N \} [set of indices]

Bootstrap Sampling

Sample with replacement N times

Committee

\[ \begin{align*}
    y_0 &= h_1(x) \\
    \hat{y}_m &= h_m(x) \\
    \hat{y}_{\text{com}} &= \frac{1}{M} \sum_{m=1}^{M} y_m \quad [\text{Regression}] \\
    \hat{y}_{\text{com}} &= \arg \max_{c} \sum_{m=1}^{M} I(y_m = c) \quad [\text{Classification}] \\
    \text{Majority Vote} \\
\end{align*} \]

Why? Because committees make better predictions [under certain assumptions]

\[ F_{\text{error}} = \mathbb{E}_{x \sim D} \left[ L(y, \hat{y}_{\text{com}}(x)) \right] \]

\[ \begin{align*}
    &= \mathbb{E} \left[ (y - \frac{1}{M} \sum_{m=1}^{M} \hat{y}_m)^2 \right] \quad [\text{For Regression}] \\
    &= \frac{1}{M^2} \mathbb{E} \left[ (My - \frac{1}{M} \sum_{m=1}^{M} \hat{y}_m)^2 \right] \quad [\text{For Regression}] \\
\end{align*} \]

Let \( e_m = y - \hat{y}_m \quad \text{Error of } m^{\text{th}} \text{ predictor} \)
\[ E_{\text{com}} = \frac{1}{M^2} E \left( \frac{\sum_{m=1}^{M} e_m^2}{\sum_{m=1}^{M} e_m} \right) \]

\[ = \frac{1}{M^2} \left[ \sum_{m=1}^{M} E[e_m^2] + \sum_{m \neq m'} E[e_{m}, e_{m}'] - E[e_{m}] \cdot E[e_{m}'] \right] \]

Now, Assume:

1. Zero mean errors: \( E[e_m] = 0 \) \( \forall m \)
2. Uncorrelated errors: \( E[e_m, e_m'] = E[e_m] \cdot E[e_m'] \)

\[ E_{\text{com}} = \frac{1}{M^2} \left[ \sum_{m=1}^{M} E[e_m^2] + \sum_{m \neq m'} E[e_{m}, e_{m}'] - E[e_{m}] \cdot E[e_{m}'] \right] \]

\[ = \frac{1}{M^2} \sum_{m=1}^{M} E[e_m^2] = \frac{1}{M^2} \sum_{m=1}^{M} E[(y - y^{(m)})^2] \]

\[ \Rightarrow E_{\text{com}} = \frac{1}{M} E_{\text{av}} \]

\[ \Rightarrow \text{Large (uncorrelated-error) committees cut error! } \frac{1}{M} \text{th of average error!} \]

Assumption 1 is okay. Why?

Assumption 2 is difficult to satisfy in practice. True for bootstrap?
Boosting: Reducing Bias of Weak learners

ie making weak learners strong.

Given: Hypothesis class $\mathcal{H} = \{ h : X \rightarrow Y \}$
collection of weak learners

Example #1: Decision Stump [for binary classification]

$$h(x) \geq t$$

$$h(x) = -1 \quad \text{if} \quad x \text{< threshold}$$
$$h(x) = +1 \quad \text{if} \quad x \geq t$$

$\mathcal{H} = \{(j, t) \mid j \in \{1, 2, \ldots, d\}, t \in \mathbb{R}\}$

Pick a feature $j$ and Pick a threshold
And you have a decision-stump classifier

Example #2: Decision Stump [for Regression]

$$h(x) = \begin{cases} 
\frac{1}{|L|} \sum_{i \in L} y_i & x_j < t \\
1 - \frac{1}{|R|} \sum_{i \in R} y_i & x_j \geq t 
\end{cases}$$

[Bucket Average]

$0^{th}$ order polynomial

$\frac{1}{|L|} \sum_{i \in L} y_i \
\frac{1}{|R|} \sum_{i \in R} y_i$
\[ h(x) = \begin{cases} \mathbf{w}_L^T x & x_j < t \\ \mathbf{w}_R^T x & x_j \geq t \end{cases} \]

Assume: Access to a Black-Box learning algorithm to pick a weak learner for a dataset, i.e.:

\[
\hat{h}^* = \text{argmin}_{h \in H} \frac{1}{N} \sum_{i=1}^{N} L(y_i, h(x_i))
\]

Can be solved

Goal of Boosting: Learn a committee of weak learners with low-loss

\[
f(x) = \sum_{t} \alpha_t h_t(x)
\]

\[
\min_{\alpha_t, h_t} \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i))
\]

Problem: Joint optimization of all weak learners is difficult

Boosting involves greedy optimization.
At "time" t fixed
\[
\sum_{i=1}^{t} \alpha_i h_i(x) + \alpha_{t+1} h_{t+1}(x) + \cdots + \alpha_t h_t(x) + \alpha_{t+1} h_{t+1}(x)
\]

\[
f_t(x) = f_{t-1}(x) + \alpha_t h_t(x)
\]

So
\[
\alpha^*_t, h^*_t = \min_{\alpha_t \in \mathbb{R}} \min_{h_t \in H} \sum_{i=1}^{N} \frac{1}{N} \sum_{i=1}^{N} \left[y_i - f_t(x_i)\right]^2
\]

\[
L_2\text{-Boost: Assume Regression is } L_2\text{-error}
\]

\[
\left(\alpha^*_t, h^*_t\right) = \min_{\alpha_t \in \mathbb{R}} \min_{h_t \in H} \frac{1}{N} \sum_{i=1}^{N} \left[y_i - f_t(x_i)\right]^2
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} \left[y_i - f_{t-1}(x_i) - \alpha_t h_t(x_i)\right]^2
\]

\[
\text{Current-Erro}r
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} \left[e_i - \alpha_t h_t(x_i)\right]^2
\]

\[
\text{Usually } \alpha_t \text{ scalar can be learned inside } h_t \text{ so we set } \alpha_t = 1
\]

\[
= \min_{h_t \in H} \frac{1}{N} \sum_{i=1}^{N} [e_i - h_t(x_i)]^2
\]

\[
\text{Black-Box Call: Regress to Error!}
\]

Train weak learner \( h(x) \) with \( GT \ y_i = e_i \)