Bagging Classifiers

[Breiman, ML journal, 1996]

Bagging = Bootstrap aggregating

Bootstrap sampling: given set $D$ containing $m$ training examples

- Create $D^i$ by drawing $m$ examples at random with replacement from $D$
- $D^i$ expected to leave out .37 of examples from $D$

Bagging:

- Create $k$ bootstrap samples $D^1 \ldots D^k$
- Train distinct classifier on each $D^i$
- Classify new instance by classifier vote (equal weights)
Bagging Experiment

[Breiman, ML journal, 1996]

Given sample $S$ of labeled data, do 100 times and report average

1. Divide $S$ randomly into test set $T$ (10 %) and training set $D$ (90 %)

2. Learn decision tree from $D$.
   - $e_S \leftarrow$ error of tree on $T$

3. Do 50 times: Create bootstrap set $D^i$, learn decision tree, prune using $D$.
   - $e_B \leftarrow$ error of majority vote using trees to classify $T$
Bagging

insert table 2 and table 10 from Brieman
Bagging - When Should this Help?

When learner is *unstable*

- Learner is unstable if small change to training set causes large change in output hypothesis (e.g., decision trees, neural networks, not \( k \) nearest neighbor)

- Experimentally, bagging can help substantially for unstable learners, can somewhat degrade results for stable learners
Bagging

Consider real valued target function, use mean instead of majority vote to combine classifier outputs

\[ \phi_A(x) = E_D \phi(x, D) \]

where \( \phi(x, D) \) is base classifier, \( \phi_A \) is aggregated classifier

\[ E_D(y - \phi(x, D))^2 = y^2 - 2yE_D\phi(x, D) + E_D\phi^2(x, D) \]

now using \( E_D\phi(x, D) = \phi_A(x) \), and \( E\varepsilon^2 \geq (E\varepsilon)^2 \),

\[ E_D(y - \phi(x, D))^2 \geq (y - \phi_A(x))^2 \]

so we expect a lower error for the bagged predictor \( \phi_A \)
Weighted Majority
$a_i$ denotes the $i^{th}$ prediction algorithm in the pool $A$ of algorithms. $w_i$ denotes the weight associated with $a_i$.

- For all $i$ initialize $w_i \leftarrow 1$
- For each training example $\langle x, c(x) \rangle$
  * Initialize $q_0$ and $q_1$ to 0
  * For each prediction algorithm $a_i$
    - If $a_i(x) = 0$ then $q_0 \leftarrow q_0 + w_i$
    - If $a_i(x) = 1$ then $q_1 \leftarrow q_1 + w_i$
  * If $q_1 > q_0$ then predict $c(x) = 1$
    - If $q_0 > q_1$ then predict $c(x) = 0$
    - If $q_1 = q_0$ then predict 0 or 1 at random for $c(x)$
  * For each prediction algorithm $a_i$ in $A$ do
    - If $a_i(x) \neq c(x)$ then $w_i \leftarrow \beta w_i$
Weighted Majority

[Relative mistake bound for WEIGHTED-MAJORITY] Let $D$ be any sequence of training examples, let $A$ be any set of $n$ prediction algorithms, and let $k$ be the minimum number of mistakes made by any algorithm in $A$ for the training sequence $D$. Then the number of mistakes over $D$ made by the WEIGHTED-MAJORITY algorithm using $\beta = \frac{1}{2}$ is at most

$$2.4(k + \log_2 n)$$