Ensemble Learning

How to combine multiple classifiers into a single one

Works well if the classifiers are complementary

This class: two types of ensemble methods:

   Bagging
   Boosting
Bagging (Bootstrap AGgregatING)

**Input:** n labelled training examples \((x_i, y_i), i = 1, \ldots, n\)

**Algorithm:**
Repeat k times:
- Select m samples out of n with replacement to get training set \(S_i\)
- Train classifier (decision tree, k-NN, perceptron, etc) \(h_i\) on \(S_i\)

**Output:** Classifiers \(h_1, \ldots, h_k\)

**Classification:** On test example \(x\), output majority\((h_1, \ldots, h_k)\)
Example

**Input:** $n$ labelled training examples $(x_i, y_i), i = 1, \ldots, n$

**Algorithm:**

Repeat $k$ times:

- Select $m$ samples out of $n$ **with replacement** to get training set $S_i$
- Train classifier (decision tree, $k$-NN, perceptron, etc) $h_i$ on $S_i$

**How to pick $m$?**

- Popular choice: $m = n$
- Still different from working with entire training set. Why?
Bagging

**Input:** $n$ labelled training examples $S = \{(x_i, y_i)\}, i = 1, \ldots, n$

Suppose we select $n$ samples out of $n$ with replacement to get training set $S_i$

Still different from working with entire training set. Why?

$$\Pr(S_i = S) = \frac{n!}{n^n} \quad \text{(tiny number, exponentially small in } n)$$

$$\Pr((x_i, y_i) \text{ not in } S_i) = \left(1 - \frac{1}{n}\right)^n \approx e^{-1}$$

For large data sets, about 37% of the data set is left out!
Bias and Variance

Classification error  = Bias + Variance
Bias and Variance

Classification error  = Bias + Variance

Bias is the true error of the best classifier in the concept class (e.g., best linear separator, best decision tree on a fixed number of nodes).

Bias is high if the concept class cannot model the true data distribution well, and does not depend on training set size.
Bias

High Bias

Low Bias

Underfitting: when you have high bias
Bias and Variance

Classification error $= \text{Bias} + \text{Variance}$

Variance is the error of the trained classifier with respect to the best classifier in the concept class.

Variance depends on the training set size. It decreases with more training data, and increases with more complicated classifiers.
Variance

Overfitting: when you have extra high variance
Bias and Variance

Classification error  = Bias + Variance

If you have high bias, both training and test error will be high.

If you have high variance, training error will be low, and test error will be high.
Bias Variance Tradeoff

If we make the concept class more complicated (e.g, linear classification to quadratic classification, or increase number of nodes in the decision tree), then bias decreases but variance increases.

Thus there is a bias-variance tradeoff
Why is Bagging useful?

Bagging reduces the variance of the classifier, doesn't help much with bias.
Ensemble Learning

How to combine multiple classifiers into a single one

Works well if the classifiers are complementary

This class: two types of ensemble methods:

  Bagging

  Boosting
Boosting

**Goal:** Determine if an email is spam or not based on text in it

**From:** Yuncong Chen
**Text:** 151 homeworks are all graded...

**From:** Work from home solutions
**Text:** Earn money without working!

Sometimes it is:
* Easy to come up with simple rules-of-thumb classifiers,
* Hard to come up with a single high accuracy rule
**Boosting**

**Goal:** Detect if an image contains a face in it

Sometimes it is:
* Easy to come up with simple rules-of-thumb classifiers,
* Hard to come up with a single high accuracy rule

Is the black region darker on average than the white?
Boosting

**Weak Learner:** A simple rule-of-the-thumb classifier that doesn’t necessarily work very well

**Strong Learner:** A good classifier

**Boosting:** How to combine many weak learners into a strong learner?
Boosting

Procedure:

1. Design a method for finding a good rule-of-thumb
2. Apply method to training data to get a good rule-of-thumb
3. Modify the training data to get a 2nd data set
4. Apply method to 2nd data set to get a good rule-of-thumb
5. Repeat T times...
Boosting

1. How to get a good rule-of-thumb?
   Depends on application
   e.g., single node decision trees

2. How to choose examples on each round?
   Focus on the **hardest examples** so far --
   namely, examples misclassified most often by
   previous rules of thumb

3. How to combine the rules-of-thumb to a prediction rule?
   Take a weighted majority of the rules
Some Notation

Let $D$ be a distribution over examples, and $h$ be a classifier. Error of $h$ with respect to $D$ is:

$$err_D(h) = Pr((X,Y) \sim D (h(X) \neq Y))$$

Example:

Below $X$ is uniform over $[0, 1]$, and $Y = 1$ if $X > 0.5$, 0 otherwise.

$$err_D(h) = 0.25$$
Let \( D \) be a distribution over examples, and \( h \) be a classifier. Error of \( h \) with respect to \( D \) is:

\[
\text{err}_D(h) = \Pr_{(X,Y) \sim D}(h(X) \neq Y)
\]

\( h \) is called a **weak learner** if \( \text{err}_D(h) < 0.5 \)

If you guess completely randomly, then the error is 0.5
Some Notation

Given training examples \( \{(x_i, y_i)\}, i=1,\ldots,n \), we can assign weights \( w_i \) to the examples. If the \( w_i \)s sum to 1, then we can think of them as a distribution \( W \) over the examples.

The error of a classifier \( h \) wrt \( W \) is:

\[
err_W(h) = \sum_{i=1}^{n} w_i 1(h(x_i) \neq y_i)
\]

Note: \( 1 \) here is the indicator function
Boosting

Given training set \( S = \{(x_1, y_1), \ldots, (x_n, y_n)\}, \ y \in \{-1, 1\} \)

For \( t = 1, \ldots, T \)

Construct distribution \( D_t \) on the examples

Find weak learner \( h_t \) which has small error \( err_{D_t}(h_t) \) wrt \( D_t \)

Output final classifier

Initially, \( D_1(i) = 1/n \), for all \( i \) (uniform)

Given \( D_t \) and \( h_t \):

\[
D_{t+1}(i) = \frac{D_t(i)}{Z_t} \exp(-\alpha_t y_i h_t(x_i))
\]

where:

\[
\alpha_t = \frac{1}{2} \ln \left( \frac{1 - err_{D_t}(h_t)}{err_{D_t}(h_t)} \right)
\]

\( Z_t = \) normalization constant
Boosting

Given training set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $y$ in \{-1, 1\}

For $t = 1, \ldots, T$

Construct distribution $D_t$ on the examples

Find weak learner $h_t$ which has small error $err_{D_t}(h_t)$ wrt $D_t$

Output final classifier

Initially, $D_1(i) = 1/n$, for all $i$ (uniform)

Given $D_t$ and $h_t$:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \exp(-\alpha_t y_i h_t(x_i))$$

where:

$$\alpha_t = \frac{1}{2} \ln \left( \frac{1 - err_{D_t}(h_t)}{err_{D_t}(h_t)} \right)$$

$Z_t$ = normalization constant

$D_{t+1}(i)$ goes down if $x_i$ is classified correctly by $h_t$, up otherwise

High $D_{t+1}(i)$ means hard example
Boosting

Given training set \( S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), \( y \) in \{-1, 1\}

For \( t = 1, \ldots, T \)
   - Construct distribution \( D_t \) on the examples
   - Find weak learner \( h_t \) which has small error \( \text{err}_{D_t}(h_t) \) wrt \( D_t \)

Output final classifier

Initially, \( D_1(i) = 1/n \), for all \( i \) (uniform)

Given \( D_t \) and \( h_t \):
\[
D_{t+1}(i) = \frac{D_t(i)}{Z_t} \exp(-\alpha_t y_i h_t(x_i))
\]

where:
\[
\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \text{err}_{D_t}(h_t)}{\text{err}_{D_t}(h_t)} \right)
\]
Higher if \( h_t \) has low error wrt \( D_t \), lower otherwise. >0 if \( \text{err}_{D_t}(h_t) < 0.5 \)

\( Z_t \) = normalization constant
Boosting

Given training set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $y$ in $\{-1, 1\}$

For $t = 1, \ldots, T$
- Construct distribution $D_t$ on the examples
- Find weak learner $h_t$ which has small error $err_{D_t}(h_t)$ wrt $D_t$
- Output final classifier

Initially, $D_1(i) = 1/n$, for all i (uniform)

Given $D_t$ and $h_t$:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \exp(-\alpha_t y_i h_t(x_i))$$

where:

$$\alpha_t = \frac{1}{2} \ln \left( \frac{1 - err_{D_t}(h_t)}{err_{D_t}(h_t)} \right)$$

$Z_t = \text{normalization constant}$

Final classifier: $\text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right)$
Boosting: Example

weak classifiers: horizontal or vertical half-planes

Schapire, 2011
Boosting: Example

weak classifiers: horizontal or vertical half-planes

Schapire, 2011
Boosting: Example

weak classifiers: horizontal or vertical half-planes

Schapire, 2011
Boosting: Example

weak classifiers: horizontal or vertical half-planes

Schapire, 2011
The Final Classifier

\[ H_{\text{final}} = \text{sign} (0.42 + 0.65 + 0.92) \]

weak classifiers: horizontal or vertical half-planes

Schapire, 2011
How to Find Stopping Time

Given training set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $y \in \{-1, 1\}$

For $t = 1, \ldots, T$
   - Construct distribution $D_t$ on the examples
   - Find weak learner $h_t$ which has small error $\text{err}_{D_t}(h_t)$ wrt $D_t$

Output final classifier

To find stopping time, use a validation dataset. Stop when the error on the validation dataset stops getting better, or when you can’t find a good rule of thumb.