Boosting (almost) by hand
Magic: \[ \alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right) \]

- Initialize weights to uniform dist: \( D_1(j) = 1/N \)
- For \( t = 1 \ldots T \)
  - Train weak learner \( h_t \) on distribution \( D_t \) over the data
  - Choose weight \( \alpha_t \)
  - Update weights:
    - For each \( j \):
      \[ D_{t+1}(j) = \frac{D_t(j) \exp(-\alpha_t y^j h_t(x^j))}{Z_t} \]
- Where \( Z_t \) is normalizer:
  \[ Z_t = \sum_{j=1}^{N} D_t(j) \exp(-\alpha_t y^j h_t(x^j)) \]
  - So weights add up to 1
- Output final classifier:
  \[ H(x) = \text{Sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right) \]
(from Rob Schapire)

N = 10
Whats error train here?

$$D_1(j) = \frac{1}{N} = 0.1$$

$$\epsilon_t = \sum_{j=1}^{N} D_t(j) 1[\text{sign}(h_t(x^j)) \neq y^j]$$
What's the error train here?

\[ D_1(j) = \frac{1}{N} = 0.1 \]

\[ \epsilon_t = \sum_{j=1}^{N} D_t(j) 1[\text{sign}(h_t(x^j) \neq y^j] \]

\[ \epsilon_1 = 0.3 \]

\[ \alpha_1 = \frac{1}{2} \ln \left( \frac{0.7}{0.3} \right) \approx 0.42 \]
New weights

\[ D_{t+1}(j) = \frac{D_t(j) \exp(-\alpha_t y^j h_t(x^j))}{Z_t} \]

When is \( y^j h_t(x^j) = 1 \)?
When is \( y^j h_t(x^j) = -1 \)?
New weights

\[ D_{t+1}(j) = \frac{D_t(j) \exp(-\alpha_t y^j h_t(x^j))}{Z_t} \]

\[ D_2(right) = \frac{0.1\exp(-0.42)}{Z_t} \approx 0.071 \]

\[ D_2(wrong) = \frac{0.1\exp(0.42)}{Z_t} \approx 0.166 \]
Step 2

\[ \epsilon_2 = 0.071 \times 3 \approx 0.21 \]

\[ \alpha_2 = \frac{1}{2} \ln \left( \frac{0.79}{0.21} \right) \approx 0.65 \]

Notice I still get 3 examples wrong, but they are worth less now.
New weights

\[
D_{t+1}(j) = \frac{D_t(j) \exp(-\alpha_t y^j h_t(x^j))}{Z_t}
\]

\[
D_3(\text{small + and -}) = \frac{0.071 \exp(-0.65)}{Z_t} \approx 0.045
\]

\[
D_3(\text{medium +}) = \frac{0.166 \exp(-0.65)}{Z_t} \approx 0.1
\]

\[
D_3(\text{large -}) = \frac{0.166 \exp(-0.65)}{Z_t} \approx 0.17
\]
Step 3

\[ \epsilon_3 = 0.045 \times 3 \approx 0.14 \]

\[ \alpha_3 = \frac{1}{2} \ln\left( \frac{0.86}{0.14} \right) \approx 0.92 \]

Notice I still get 3 examples wrong.
Output final classifier:

\[ H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right) \]

\[
= \text{sign} \left( 0.42 + 0.65 + 0.92 \right)
= \text{sign} (1.99)
\]
Evaluation Metrics

• 0-1 error on test set: \[ \sum_{j=1}^{N} 1[\text{sign}(h_t(x^j)) \neq y^j] \]

• 1 – (0-1 error)/N = accuracy

• Accuracy is just % of test samples I get right.
Let’s go back to that millionaire...

- Let’s say the millionaire asks you to build a classifier to identify other millionaires.
Let’s go back to that millionaire...

• Let’s say the millionaire asks you to build a classifier to identify other millionaires.

• You build a fancy classifier, and get accuracy = 80% in some test data. Pretty good, right?
Let’s go back to that millionaire...

• Let’s say the millionaire asks you to build a classifier to identify other millionaires.

• You build a fancy classifier, and get accuracy = 80% in some test data. Pretty good, right?

• Your competition has a O(1) classifier that has accuracy ~ 99.9%. Should the millionaire fire you right away?
Let’s go back to that millionaire...

• Let’s say the millionaire asks you to build a classifier to identify other millionaires.

• You build a fancy classifier, and get accuracy = 80% in some test data. Pretty good, right?

• Your competition has a O(1) classifier that has accuracy ~ 99.9%. Should the millionaire fire you right away?

• His classifier just always predicts ‘not millionaire’.
# Confusion Matrix

<table>
<thead>
<tr>
<th>Your Result</th>
<th>Gold standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X: true positive (tp), false positive (fp) type I error</td>
</tr>
<tr>
<td>Y</td>
<td>false negative (fn), true negative (tn) type II error</td>
</tr>
</tbody>
</table>

- **True Positive (tp)**: Your result correctly identifies an actual positive case.
- **False Positive (fp)**: Your result incorrectly identifies a negative case as positive.
- **False Negative (fn)**: Your result incorrectly identifies a positive case as negative.
- **True Negative (tn)**: Your result correctly identifies a negative case.
Confusion matrix

<table>
<thead>
<tr>
<th>Gold standard</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>X</td>
<td>tp</td>
<td>fp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>fn</td>
<td>tn</td>
</tr>
</tbody>
</table>

accuracy = \frac{tp + tn}{tp + tn + fp + fn}

precision = \frac{tp}{tp + fp}

error = \frac{fp + fn}{tp + tn + fp + fn}

recall = \frac{tp}{tp + fn}
Examples:

Let’s say the net is trying to pick only blue fish. What’s the precision and the recall?

I got this figure from http://www.lucidatainc.com/2012/10/recall-and-precision-understanding-relevancy-in-ediscovery/
Examples:

Let’s say the net is trying to pick only blue fish. What’s the precision and the recall?

$P = 1/4$

$R = 1/5$

I got this figure from
Examples:

Let’s say the net is trying to pick only red fish. What’s the precision and the recall?

I got this figure from http://www.lucidatainc.com/2012/10/recall-and-precision-understanding-relevancy-in-ediscovery/
Examples:

Let’s say the net is trying to pick only red fish. What’s the precision and the recall?

\[ P = \frac{3}{4} \]
\[ R = \frac{3}{5} \]

I got this figure from http://www.lucidatainc.com/2012/10/recall-and-precision-understanding-relevancy-in-ediscovery/
A single metric?

• $F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

• $F_{beta}$

• AUC

• ...
Which one is better?

• Our millionaire identification scenario?
Which one is better?

- Spam classification
Which one is better?

- Medical classifier: $Y = \text{(operate, don’t operate)}$
Which one is better?

- Search engine: query = legal
Which one is better?

- Search engine: query = “Husky football”
- By the way: why does google show more than 1 page?
Which one is better?

- It depends on the task
- Is there imbalance?
- Are the misclassification costs the same?
- ...
- ...
- Think about evaluation when doing your projects!