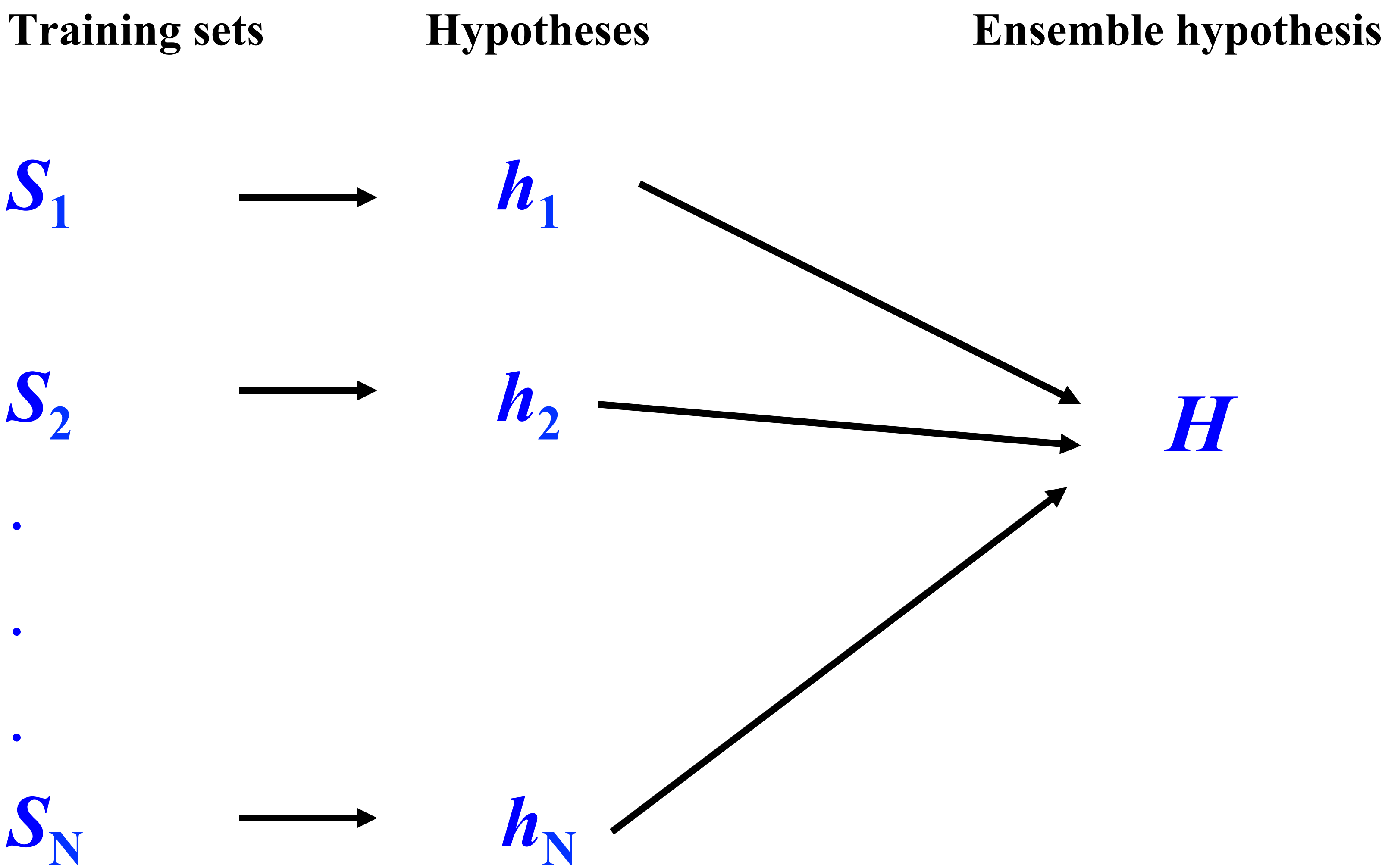
Ensemble Learning

Reading:

R. Schapire, A brief introduction to boosting

Ensemble learning



Advantages of ensemble learning

- Can be very effective at reducing generalization error! (E.g., by voting.)

- Ideal case: the h_i have independent errors

Example

Given three hypotheses, h_1, h_2, $h_{3,}$ with $h_i(\mathbf{x}) \in \{-1,1\}$

Suppose each h_i has 60% generalization accuracy, and assume errors are independent.

Now suppose $H(\mathbf{x})$ is the majority vote of h_1, h_2, and h_3 . What is probability that H is correct?

h_1	h_2	h_3	$\boldsymbol{H}$	*probability*
C	C	C	**C**	
C	C	I	**C**	
C	I	I	I	
C	I	C	**C**	
I	C	C	**C**	
I	I	C	I	
I	C	I	I	
I	I	I	I	
				Total probability correct:

h_1	h_2	h_3	***H***	*probability*
C	C	C	**C**	.216
C	C	I	**C**	.144
C	I	I	I	.096
C	I	C	**C**	.144
I	C	C	**C**	.144
I	I	C	I	.096
I	C	I	I	.096
I	I	I	I	.064
				Total probability correct: .648

Another Example

Again, given three hypotheses, h_1, h_2, $h_{3.}$

Suppose each h_i has 40% generalization accuracy, and assume errors are independent.

Now suppose we classify **x** as the majority vote of h_1, h_2, and h_3 . What is probability that the classification is correct?

h_1	h_2	h_3	$\boldsymbol{H}$	*probability*
C	C	C	**C**	.064
C	C	I	**C**	.096
C	I	I	I	.144
C	I	C	**C**	.096
I	C	C	**C**	.096
I	I	C	I	.144
I	C	I	I	.144
I	I	I	I	.261
				Total probability correct: .352

General case

In general, if hypotheses $h_1, \ldots, h_M$ all have generalization accuracy **A**, what is probability that a majority vote will be correct?

Possible problems with ensemble learning

- Errors are typically not independent
- Training time and classification time are increased by a factor of *M*.
- Hard to explain how ensemble hypothesis does classification.
- How to get enough data to create M separate data sets, $S_1, \ldots, S_M$?

- Three popular methods:

  - **Voting:**
    - Train classifier on M different training sets S_i to obtain M different classifiers h_i.
    - For a new instance x, define $H(x)$ as:

$$H(x) = \sum_{i=1}^{M} \alpha_i h_i(x)$$

where α_i is a confidence measure for classifier h_i .

- **Bagging (Breiman, 1990s):**
  - To create S_i, create "bootstrap replicates" of original training set $\boldsymbol{S}$

- **Boosting (Schapire & Freund, 1990s)**
  - To create S_i, reweight examples in original training set $\boldsymbol{S}$ as a function of whether or not they were misclassified on the previous round.

Adaptive Boosting (Adaboost)

A method for combining different weak hypotheses (training error close to but less than 50%) to produce a strong hypothesis (training error close to 0%)

Sketch of algorithm

Given examples S and learning algorithm L, with $| S | = N$

- Initialize probability distribution over examples $\mathbf{w}_1(i) = 1/N$.

- Repeatedly run L on training sets $S_t \subset$ S to produce $h_1, h_2, \ldots, h_K$.
  - At each step, derive S_t from S by choosing examples probabilistically according to probability distribution $\mathbf{w}_t$. Use S_t to learn h_t.

- At each step, derive $\mathbf{w}_{t+1}$ by giving more probability to examples that were misclassified at step t.

- The final ensemble classifier H is a weighted sum of the h_t's, with each weight being a function of the corresponding h_t's error on its training set.

Adaboost algorithm

- Given $S = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N)\}$ where $\mathbf{x} \in \boldsymbol{X}$, $y_i \in \{+1, -1\}$
- Initialize $\mathbf{w}_1(i) = 1/N$. (Uniform distribution over data)

- For $t = 1, ..., K$:
  - Select new training set S_t from S with replacement, according to $\mathbf{w}_t$
  - Train L on S_t to obtain hypothesis h_t
  - Compute the training error ε_t of h_t on $\boldsymbol{S}$:

$$\varepsilon_t = \sum_{j=1}^{N} \mathbf{w}_t(j)\, \delta(y_j \neq h_t(\mathbf{x}_j)), \text{ where}$$

$$\delta(y_j \neq h_t(\mathbf{x}_j)) = \begin{cases} 1 \text{ if } y_j \neq h_t(\mathbf{x}_j) \\ 0 \text{ otherwise} \end{cases}$$

  - Compute coefficient

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

– Compute new weights on data:

For $i = 1$ to N

$$\mathbf{w}_{t+1}(i) = \frac{\mathbf{w}_t(i)\ \exp(-\alpha_t y_i h_t(\mathbf{x}_i))}{Z_t}$$

where Z_t is a normalization factor chosen so that $\mathbf{w}_{t+1}$ will be a probability distribution:

$$Z_t = \sum_{i=1}^{N} \mathbf{w}_t(i)\ \exp(-\alpha_t y_i h_t(\mathbf{x}_i))$$

- At the end of K iterations of this algorithm, we have

  $h_1, h_2, \ldots, h_K$

  We also have

  $\alpha_1, \alpha_2, \ldots, \alpha_K$, where

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

- Ensemble classifier:

$$H(\mathbf{x}) = \operatorname{sgn}\sum_{t=1}^{K}\alpha_t h_t(\mathbf{x})$$

- Note that hypotheses with higher accuracy on their training sets are weighted more strongly.

A Hypothetical Example

$$S = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8,\}$$

where $\{ \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4 \}$ are class +1

$\{\mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8 \}$ are class −1

$t = 1$:

$\mathbf{w}_1 = \{1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8\}$

$S_1 = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_2, \mathbf{x}_5, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8\}$ (notice some repeats)

Train classifier on S_1 to get h_1

Run h_1 on **S**. Suppose classifications are: $\{\mathbf{1}, \mathbf{-1}, \mathbf{-1}, \mathbf{-1}, \mathbf{-1}, \mathbf{-1}, \mathbf{-1}, \mathbf{-1}\}$

- Calculate error: $$\varepsilon_1 = \sum_{j=1}^{N} \mathbf{w}_t(j)\delta(y_j \neq h_t(\mathbf{x}_j)) = \frac{1}{8}(3) = .375$$

Calculate α*'s*:

$$\alpha_1 = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) = .255$$

Calculate new **w***'s*:

$$\mathbf{w}_{t+1}(i) = \frac{\mathbf{w}_t(i)\ \exp(-\alpha_t y_i h_t(\mathbf{x}_i))}{Z_t}$$

$$\hat{\mathbf{w}}_2(1) = (.125)\exp(-.255(1)(1)) = 0.1$$
$$\hat{\mathbf{w}}_2(2) = (.125)\exp(-.255(1)(-1)) = 0.16$$
$$\hat{\mathbf{w}}_2(3) = (.125)\exp(-.255(1)(-1)) = 0.16$$
$$\hat{\mathbf{w}}_2(4) = (.125)\exp(-.255(1)(-1)) = 0.16$$
$$\hat{\mathbf{w}}_2(5) = (.125)\exp(-.255(-1)(-1)) = 0.1$$
$$\hat{\mathbf{w}}_2(6) = (.125)\exp(-.255(-1)(-1)) = 0.1$$
$$\hat{\mathbf{w}}_2(7) = (.125)\exp(-.255(-1)(-1)) = 0.1$$
$$\hat{\mathbf{w}}_2(8) = (.125)\exp(-.255(-1)(-1)) = 0.1$$

$$\mathbf{w}_2(1) = 0.1/.98 = 0.102$$
$$\mathbf{w}_2(2) = 0.163$$
$$\mathbf{w}_2(3) = 0.163$$
$$\mathbf{w}_2(4) = 0.163$$
$$\mathbf{w}_2(5) = 0.102$$
$$\mathbf{w}_2(6) = 0.102$$
$$\mathbf{w}_2(7) = 0.102$$
$$\mathbf{w}_2(8) = 0.102$$

$$Z_1 = \sum_i \hat{\mathbf{w}}_2(i) = .98$$

$t = 2$

$\mathbf{w}_2 = \{0.102, 0.163, 0.163, 0.163, 0.102, 0.102, 0.102, 0.102\}$

$S_2 = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_4, \mathbf{x}_7, \mathbf{x}_8\}$

Learn classifier on S_2 to get h_2

Run h_2 on **S**. Suppose classifications are: {**1**, **1**, **1**, **1**, **1**, **1**, **1**, **1**}

Calculate error:

$$\varepsilon_2 = \sum_{j=1}^{N} \mathbf{w}_t(j)\,\delta(y_j \neq h_t(\mathbf{x}_j))$$

$$= (.102) \times 4 = 0.408$$

Calculate α's: $$\alpha_2 = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) = .186$$

Calculate **w**'s:

$$\mathbf{w}_{t+1}(i) = \frac{\mathbf{w}_t(i)\ \exp(-\alpha_t y_i h_t(\mathbf{x}_i))}{Z_t}$$

$$\hat{\mathbf{w}}_3(1) = (.102)\exp(-.186(1)(1)) = 0.08$$
$$\hat{\mathbf{w}}_3(2) = (.163)\exp(-.186(1)(1)) = 0.135$$
$$\hat{\mathbf{w}}_3(3) = (.163)\exp(-.186(1)(1)) = 0.135$$
$$\hat{\mathbf{w}}_3(4) = (.163)\exp(-.186(1)(1)) = 0.135$$
$$\hat{\mathbf{w}}_3(5) = (.102)\exp(-.186(-1)(1)) = 0.122$$
$$\hat{\mathbf{w}}_3(6) = (.102)\exp(-.186(-1)(1)) = 0.122$$
$$\hat{\mathbf{w}}_3(7) = (.102)\exp(-.186(-1)(1)) = 0.122$$
$$\hat{\mathbf{w}}_3(8) = (.102)\exp(-.186(-1)(1)) = 0.122$$

$$\mathbf{w}_3(1) = 0.08 / .973 = 0.082$$
$$\mathbf{w}_3(2) = 0.139$$
$$\mathbf{w}_3(3) = 0.139$$
$$\mathbf{w}_3(4) = 0.139$$
$$\mathbf{w}_3(5) = 0.125$$
$$\mathbf{w}_3(6) = 0.125$$
$$\mathbf{w}_3(7) = 0.125$$
$$\mathbf{w}_3(8) = 0.125$$

$$Z_2 = \sum_i \hat{\mathbf{w}}_2(i) = .973$$

$t = 3$

$\mathbf{w}_3 = \{0.082, 0.139, 0.139, 0.139, 0.125, 0.125, 0.125, 0.125\}$

$S_3 = \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_3, \mathbf{x}_3, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8\}$

Run classifier on S_3 to get h_3

Run h_3 on **S**. Suppose classifications are: $\{\mathbf{1, 1, -1, 1, -1, -1, 1, -1}\}$

Calculate error:

$$\varepsilon_3 = \sum_{j=1}^{N} \mathbf{w}_t(i)\,\delta(y_j \neq h_t(\mathbf{x}_j))$$

$$= (.139) + (.125) = 0.264$$

- Calculate α's:

$$\alpha_3 = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) = .512$$

- Ensemble classifier:

$$H(\mathbf{x}) = \operatorname{sgn}\sum_{t=1}^{K}\alpha_t h_t(\mathbf{x})$$
$$= \operatorname{sgn}\left(.255 \times h_1(\mathbf{x}) + .186 \times h_2(\mathbf{x}) + .512 \times h_3(\mathbf{x})\right)$$

Example	Actual class	h_1	h_2	h_3
$\mathbf{x}_1$	1	1	1	1
$\mathbf{x}_2$	1	−1	1	1
$\mathbf{x}_3$	1	−1	1	−1
$\mathbf{x}_4$	1	1	1	1
$\mathbf{x}_5$	−1	−1	1	−1
$\mathbf{x}_6$	−1	−1	1	−1
$\mathbf{x}_7$	−1	1	1	1
$\mathbf{x}_8$	−1	−1	1	−1

Recall the training set:

$$S = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8,\}$$

where $\{ \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, x_4 \}$ are class +1

$\{\mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8 \}$ are class −1

$$H(\mathbf{x}) = \operatorname{sgn} \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

$$= \operatorname{sgn}\left(.255 \times h_1(\mathbf{x}) + .186 \times h_2(\mathbf{x}) + .512 \times h_3(\mathbf{x})\right)$$

What is the accuracy of H on the training data?

Adaboost seems to reduce both bias and variance.

Adaboost does not seem to overfit for increasing K.

Optional: Read about "Margin-theory" explanation of success of Boosting