

AdaBoost

Lecturer:
Jan Šochman

Authors:
Jan Šochman, Jiří Matas

Centre for Machine Perception
Czech Technical University, Prague
<http://cmp.felk.cvut.cz>



Outline:

- ◆ AdaBoost algorithm
 - Why is of interest?
 - How it works?
 - Why it works?
- ◆ AdaBoost variants

What is AdaBoost?

AdaBoost is an algorithm for constructing a “strong” classifier as linear combination

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

of “simple” “weak” classifiers $h_t(x)$.

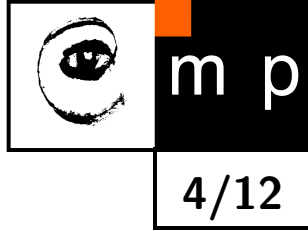
Terminology

- ◆ $h_t(x)$... “weak” or basis classifier, hypothesis, “feature”
- ◆ $H(x) = \text{sign}(f(x))$... “strong” or final classifier/hypothesis

Interesting properties

- ◆ AB is a linear classifier with all its desirable properties.
- ◆ AB output converges to the logarithm of likelihood ratio.
- ◆ AB has good generalisation properties.
- ◆ AB is a feature selector with a principled strategy (minimisation of upper bound on empirical error).
- ◆ AB close to sequential decision making (it produces a sequence of gradually more complex classifiers).

The AdaBoost Algorithm

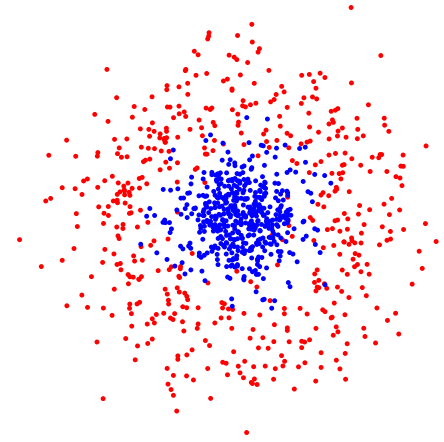


Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

The AdaBoost Algorithm

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

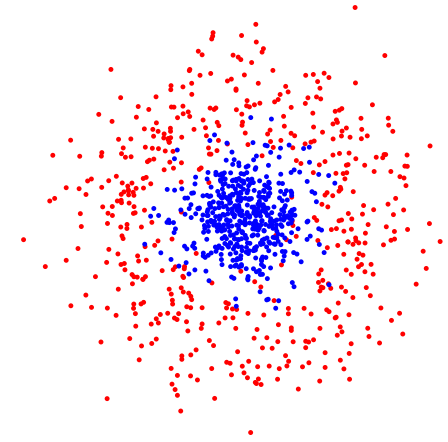


The AdaBoost Algorithm

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:



The AdaBoost Algorithm

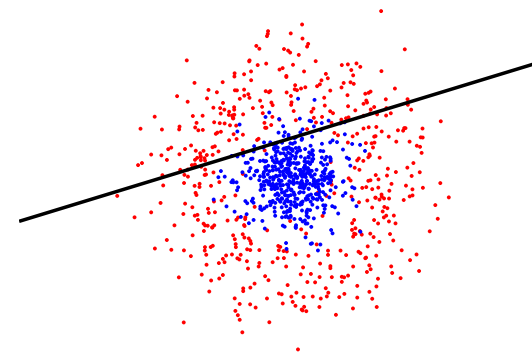
Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$

$t = 1$



The AdaBoost Algorithm

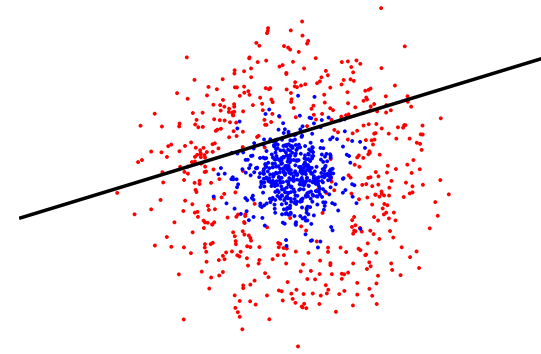
Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop

$t = 1$



The AdaBoost Algorithm

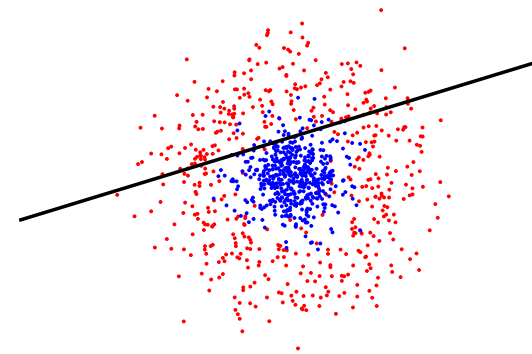
Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

$t = 1$



The AdaBoost Algorithm

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

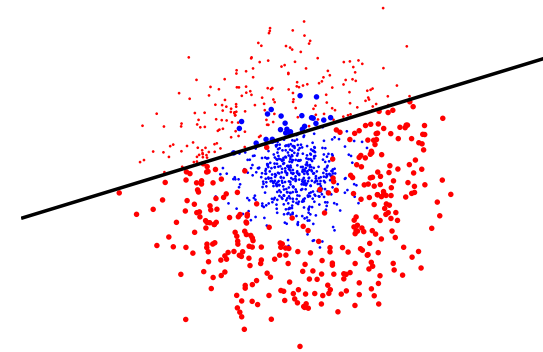
Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

$t = 1$



The AdaBoost Algorithm

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

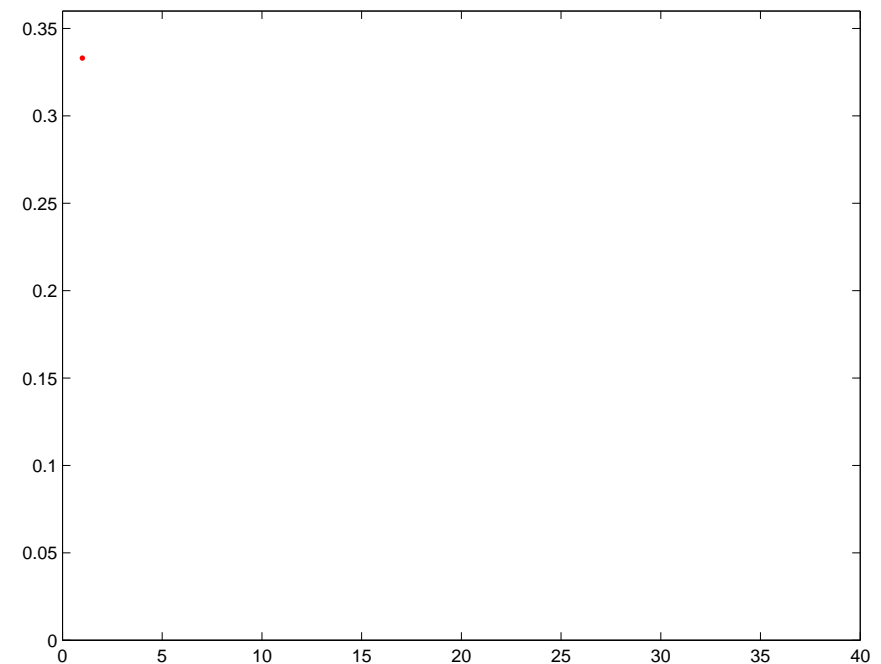
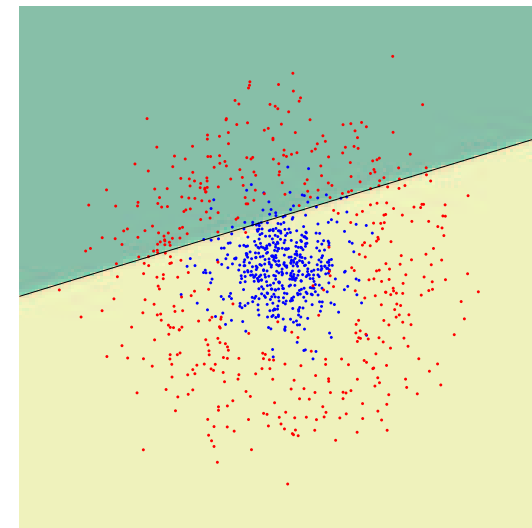
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \llbracket y_i \neq h_j(x_i) \rrbracket$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 1$



The AdaBoost Algorithm

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

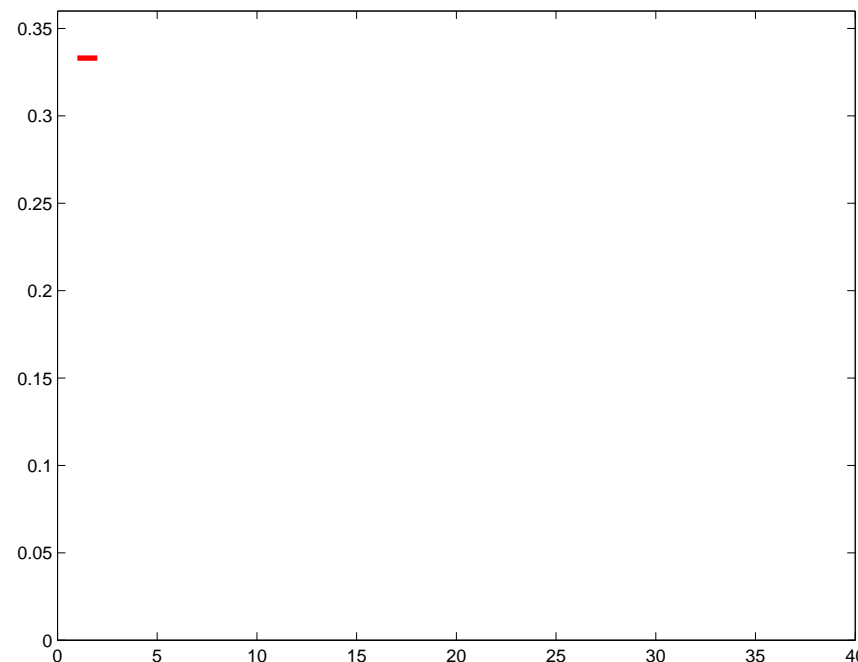
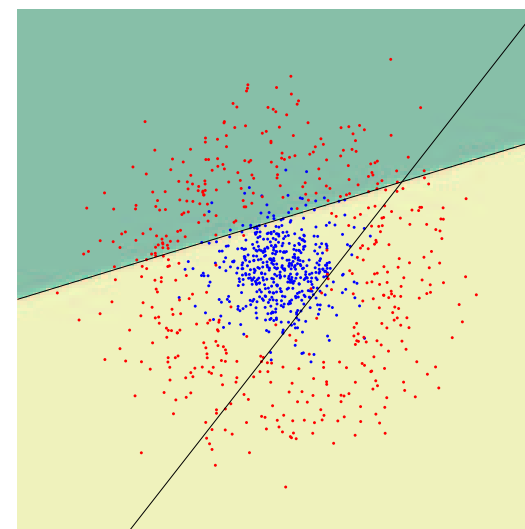
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 2$



The AdaBoost Algorithm

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

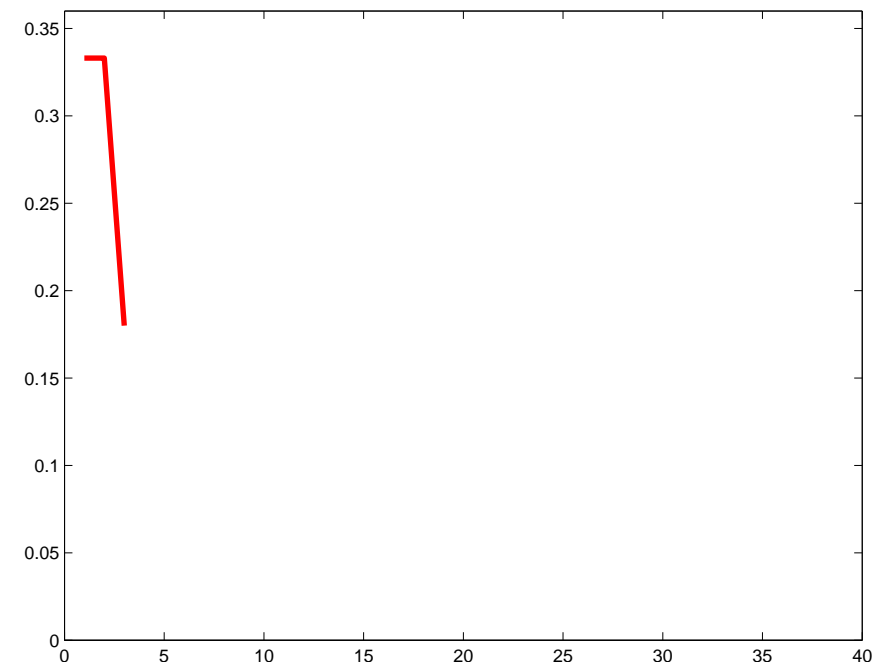
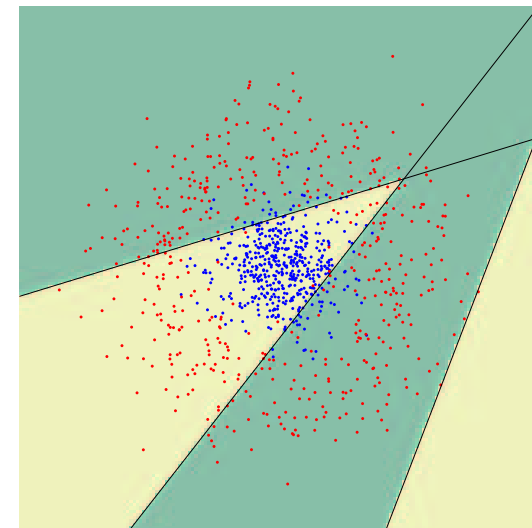
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 3$



The AdaBoost Algorithm

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

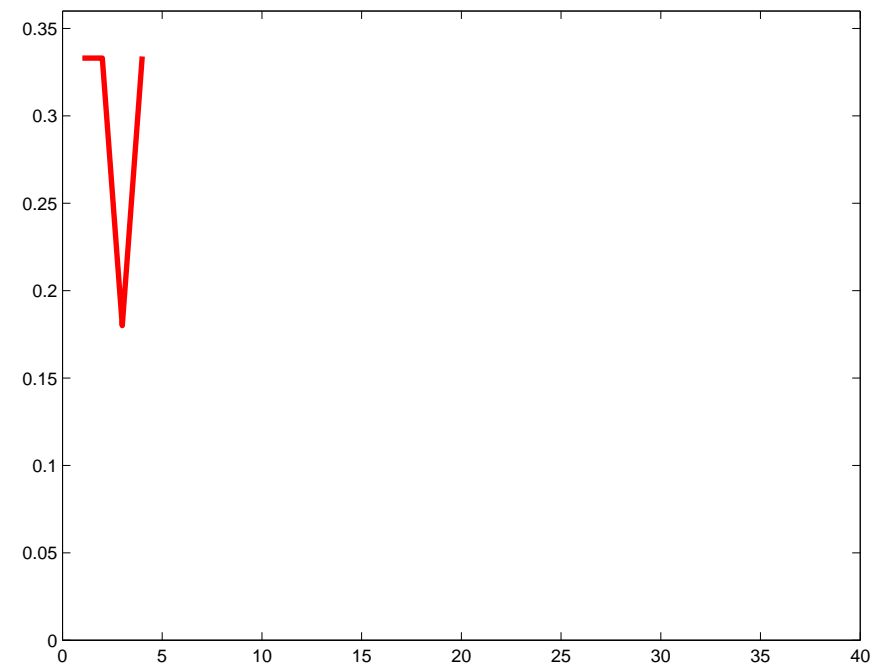
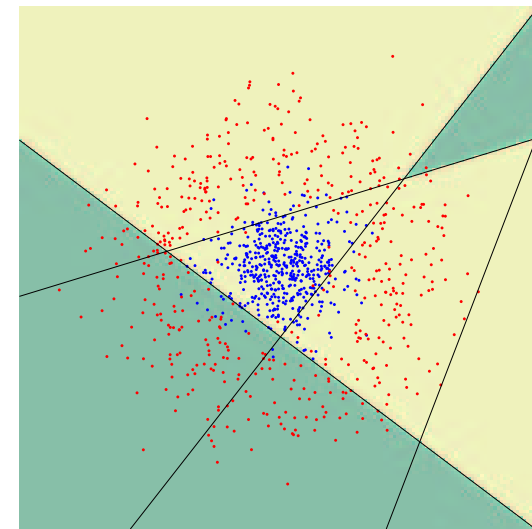
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 4$



The AdaBoost Algorithm

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

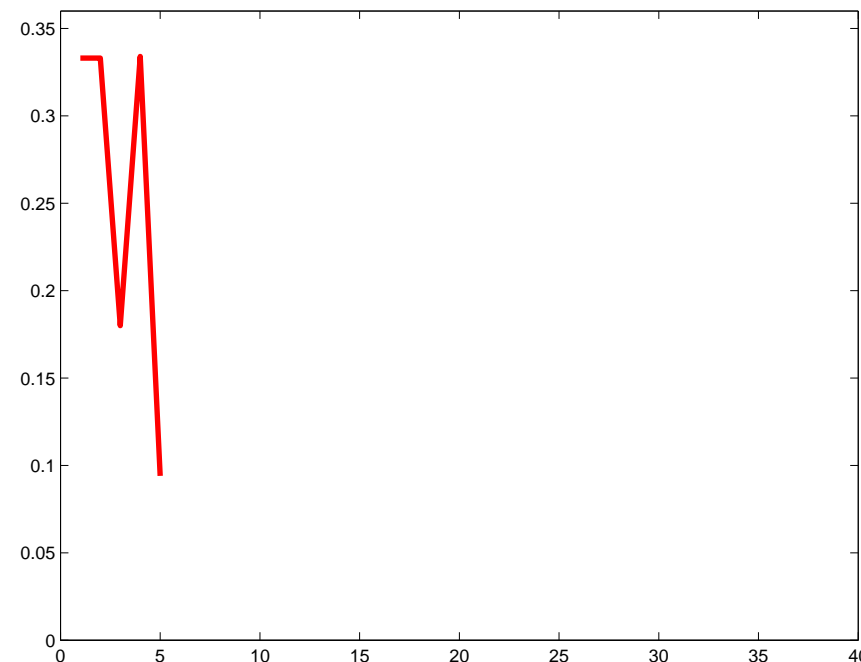
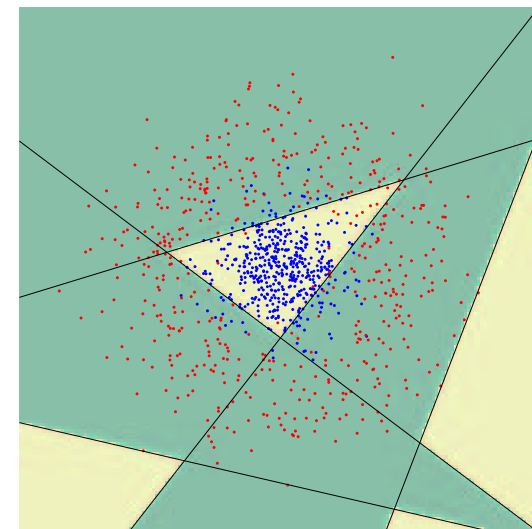
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 5$



The AdaBoost Algorithm

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

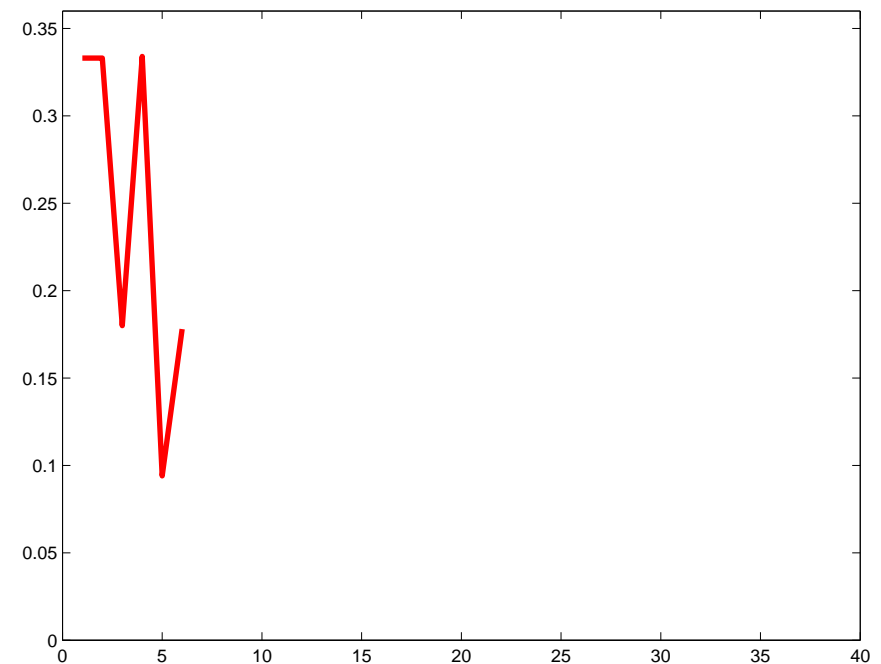
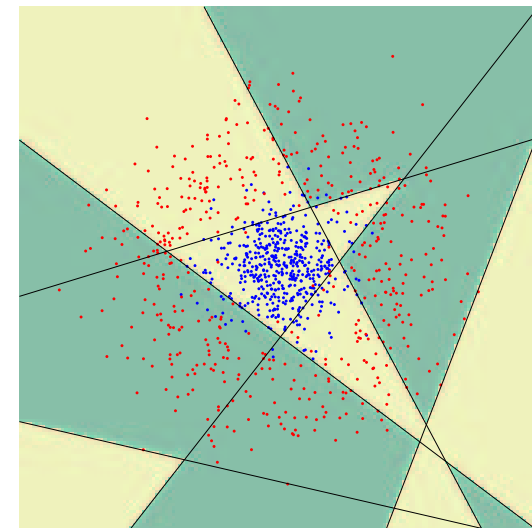
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 6$



The AdaBoost Algorithm

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

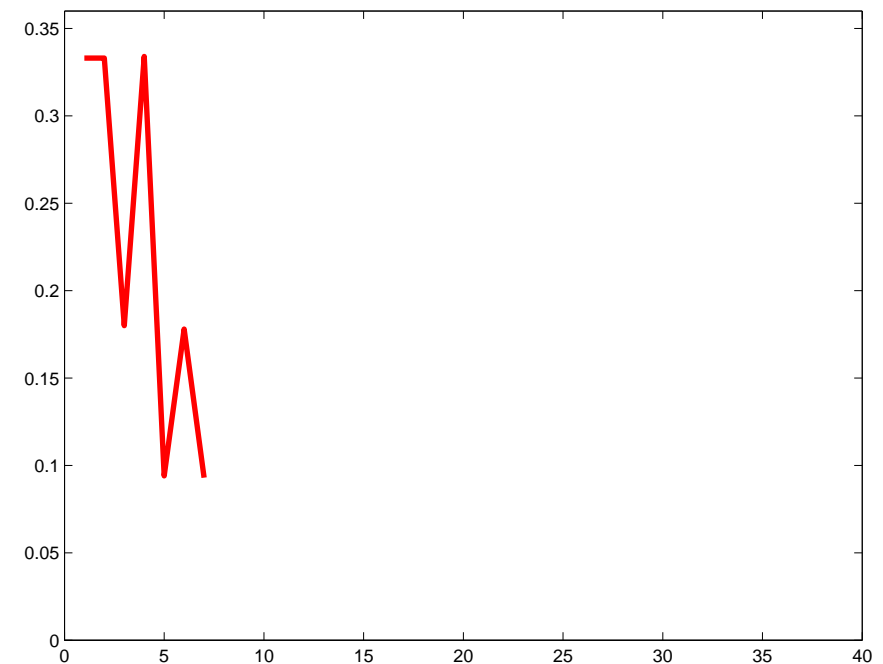
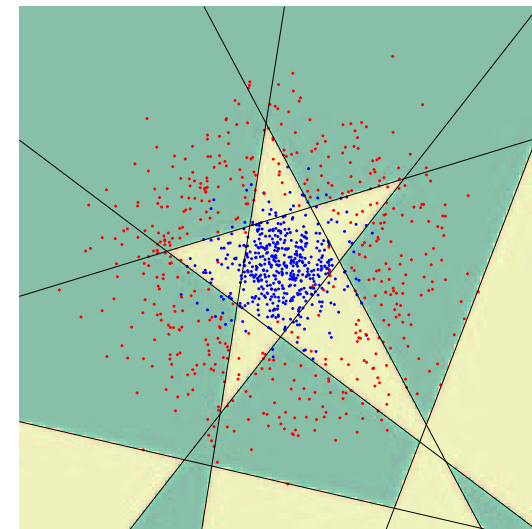
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 7$



The AdaBoost Algorithm

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

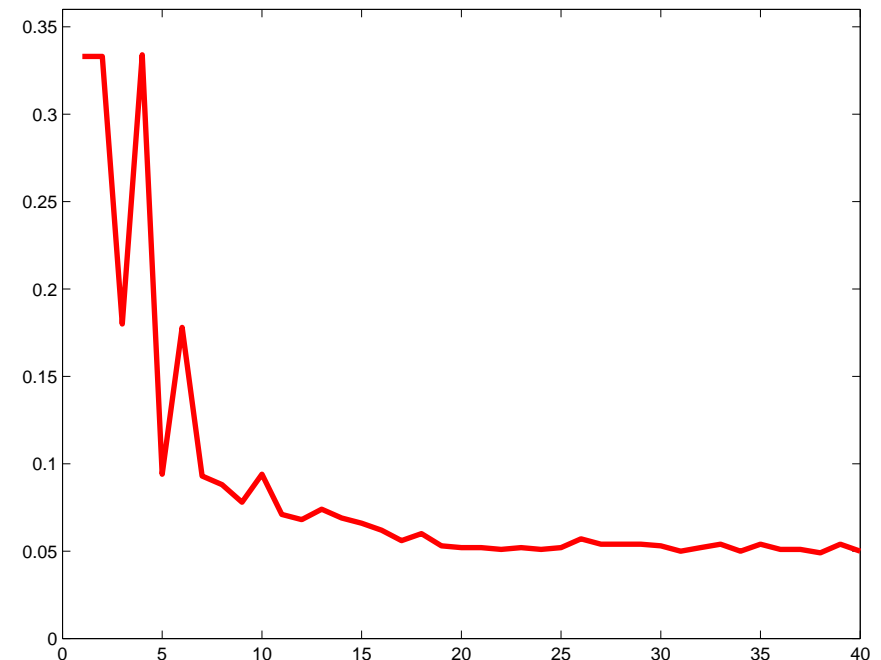
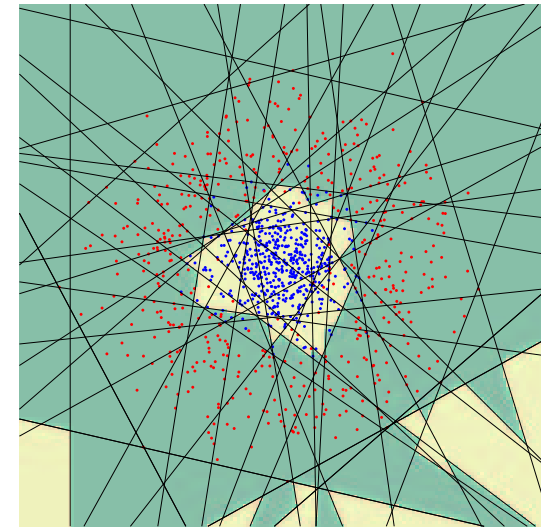
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 40$

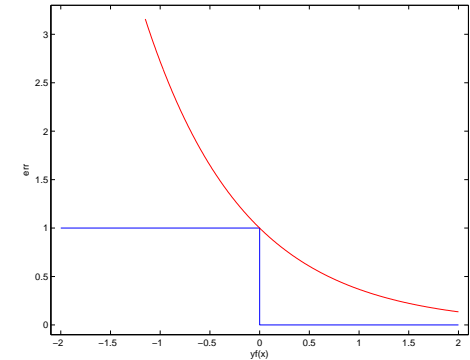


Reweighting

Effect on the training set

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

$$\exp(-\alpha_t y_i h_t(x_i)) \begin{cases} < 1, & y_i = h_t(x_i) \\ > 1, & y_i \neq h_t(x_i) \end{cases}$$



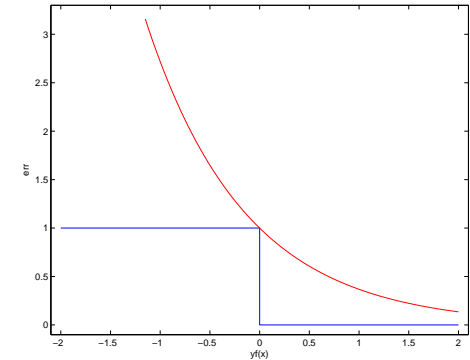
- ⇒ Increase (decrease) weight of wrongly (correctly) classified examples
- ⇒ The weight is the upper bound on the error of a given example.
- ⇒ All information about previously selected “features” is captured in D_t .

Reweighting

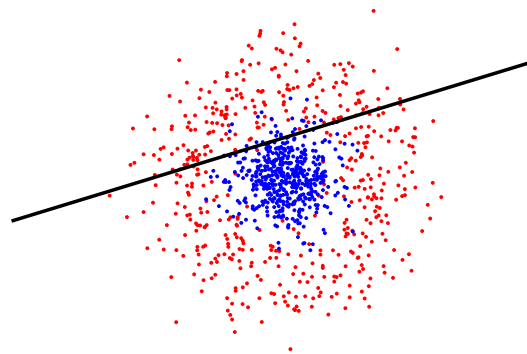
Effect on the training set

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

$$\exp(-\alpha_t y_i h_t(x_i)) \begin{cases} < 1, & y_i = h_t(x_i) \\ > 1, & y_i \neq h_t(x_i) \end{cases}$$



- ⇒ Increase (decrease) weight of wrongly (correctly) classified examples
- ⇒ The weight is the upper bound on the error of a given example.
- ⇒ All information about previously selected “features” is captured in D_t .

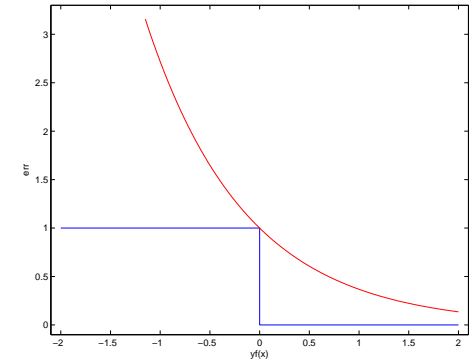


Reweighting

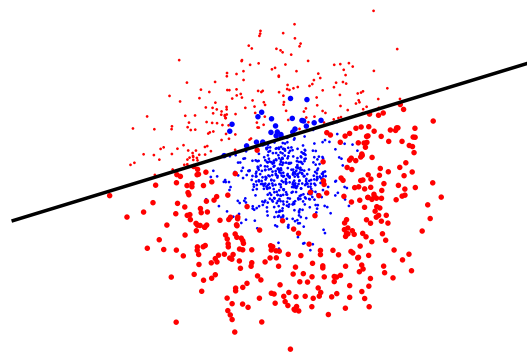
Effect on the training set

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

$$\exp(-\alpha_t y_i h_t(x_i)) \begin{cases} < 1, & y_i = h_t(x_i) \\ > 1, & y_i \neq h_t(x_i) \end{cases}$$



- ⇒ Increase (decrease) weight of wrongly (correctly) classified examples
- ⇒ The weight is the upper bound on the error of a given example.
- ⇒ All information about previously selected “features” is captured in D_t .

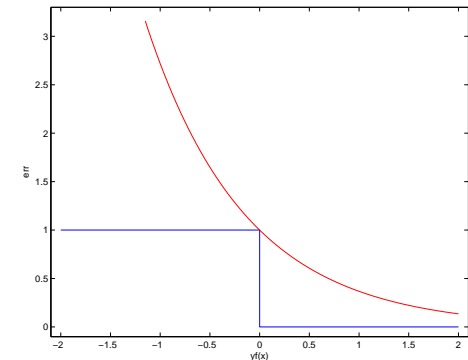


Reweighting

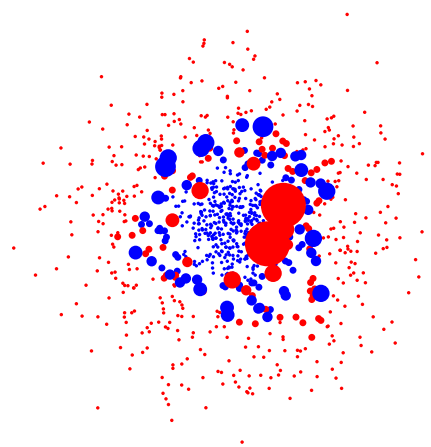
Effect on the training set

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

$$\exp(-\alpha_t y_i h_t(x_i)) \begin{cases} < 1, & y_i = h_t(x_i) \\ > 1, & y_i \neq h_t(x_i) \end{cases}$$



- ⇒ Increase (decrease) weight of wrongly (correctly) classified examples
- ⇒ The weight is the upper bound on the error of a given example.
- ⇒ All information about previously selected “features” is captured in D_t .



Upper Bound Theorem

Theorem: The following upper bound holds on the training error of H

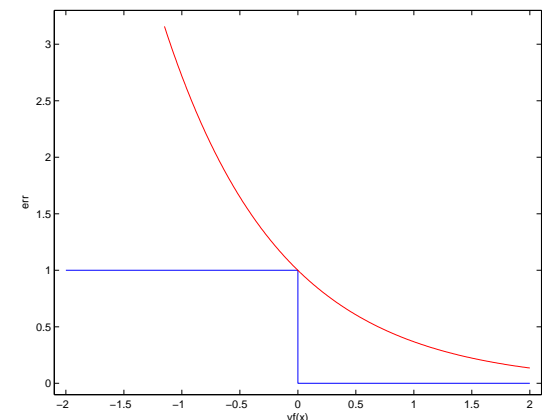
$$\frac{1}{m} |\{i : H(x_i) \neq y_i\}| \leq \prod_{t=1}^T Z_t$$

Proof: By unravelling the update rule

$$\begin{aligned} D_{T+1}(i) &= \frac{\exp(-\sum_t \alpha_t y_i h_t(x_i))}{m \prod_t Z_t} \\ &= \frac{\exp(-y_i f(x_i))}{m \prod_t Z_t}. \end{aligned}$$

If $H(x_i) \neq y_i$ then $y_i f(x_i) \leq 0$ implying that $\exp(-y_i f(x_i)) > 1$, thus

$$\begin{aligned} \mathbb{1}[H(x_i) \neq y_i] &\leq \exp(-y_i f(x_i)) \\ \frac{1}{m} \sum_i \mathbb{1}[H(x_i) \neq y_i] &\leq \frac{1}{m} \sum_i \exp(-y_i f(x_i)) \\ &= \sum_i \left(\prod_t Z_t \right) D_{T+1}(i) = \prod_t Z_t \end{aligned}$$



Consequences of the Theorem

- ◆ Instead of minimising the training error, its upper bound can be minimised.
- ◆ This can be done by minimising Z_t in each training round by:
 - Choosing optimal h_t , and
 - Finding optimal α_t .
- ◆ Can be proved to maximise margin.

Choosing α_t

We attempt to minimise $Z_t = \sum_i D_t(i) \exp(-\alpha_t y_i h_t(x_i))$:

$$\begin{aligned} \frac{dZ}{d\alpha} &= - \sum_{i=1}^m D(i) y_i h(x_i) e^{-y_i \alpha h(x_i)} = 0 \\ - \sum_{i: y_i = h(x_i)} D(i) e^{-\alpha} + \sum_{i: y_i \neq h(x_i)} D(i) e^{\alpha} &= 0 \\ -e^{-\alpha}(1 - \epsilon) + e^{\alpha}\epsilon &= 0 \\ \alpha &= \frac{1}{2} \log \frac{1 - \epsilon}{\epsilon} \end{aligned}$$

\Rightarrow The minimisator of the upper bound is $\alpha_t = \frac{1}{2} \log \frac{1 - \epsilon_t}{\epsilon_t}$.

Choosing h_t

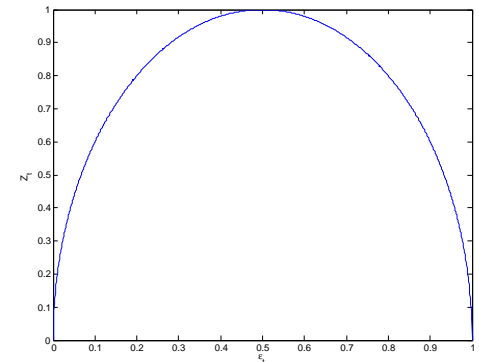
Weak classifier examples

- ◆ Decision tree, Perceptron – \mathcal{H} infinite
- ◆ Selecting the best one from given *finite* set \mathcal{H}

Justification of the weighted error minimisation

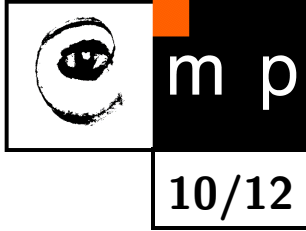
Having $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$

$$\begin{aligned}
 Z_t &= \sum_{i=1}^m D_t(i) e^{-y_i \alpha_t h_t(x_i)} \\
 &= \sum_{i: y_i = h_t(x_i)} D_t(i) e^{-\alpha_t} + \sum_{i: y_i \neq h_t(x_i)} D_t(i) e^{\alpha_t} \\
 &= (1 - \epsilon_t) e^{-\alpha_t} + \epsilon_t e^{\alpha_t} \\
 &= 2\sqrt{\epsilon_t(1 - \epsilon_t)}
 \end{aligned}$$



$\Rightarrow Z_t$ is minimised by selecting h_t with minimal weighted error ϵ_t .

The Algorithm Recapitulation

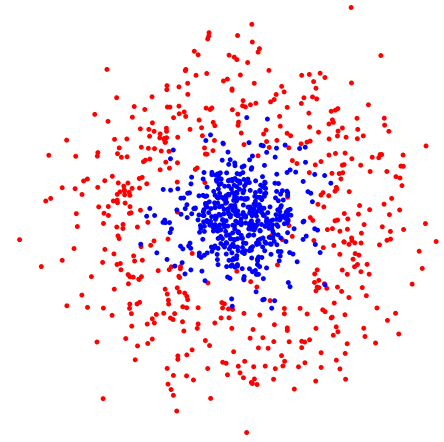


Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

The Algorithm Recapitulation

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

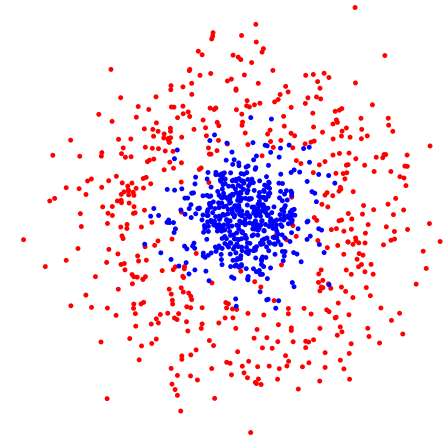


The Algorithm Recapitulation

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:



The Algorithm Recapitulation

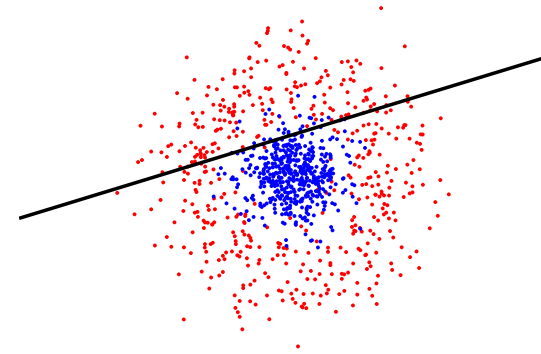
Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

- Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$

$t = 1$



The Algorithm Recapitulation

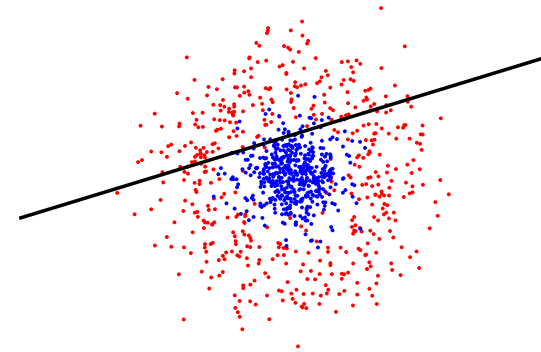
Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop

$t = 1$



The Algorithm Recapitulation

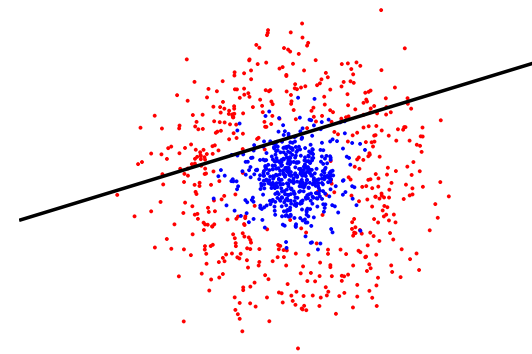
Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

$t = 1$



The Algorithm Recapitulation

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

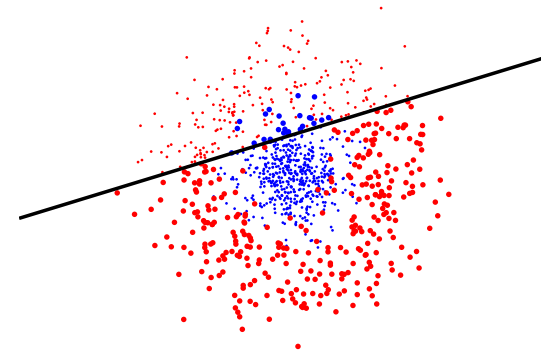
Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \llbracket y_i \neq h_j(x_i) \rrbracket$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

$t = 1$



The Algorithm Recapitulation

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

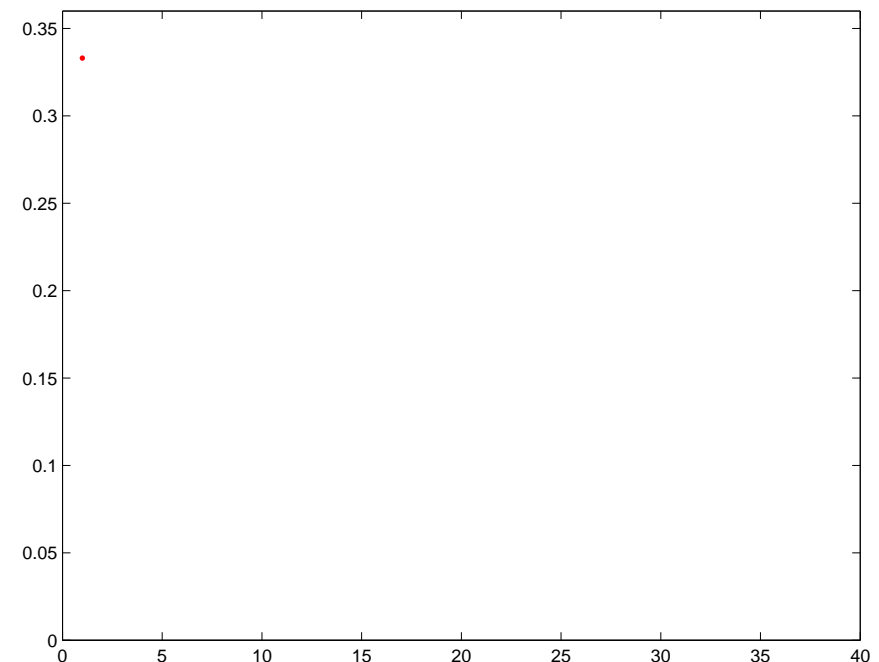
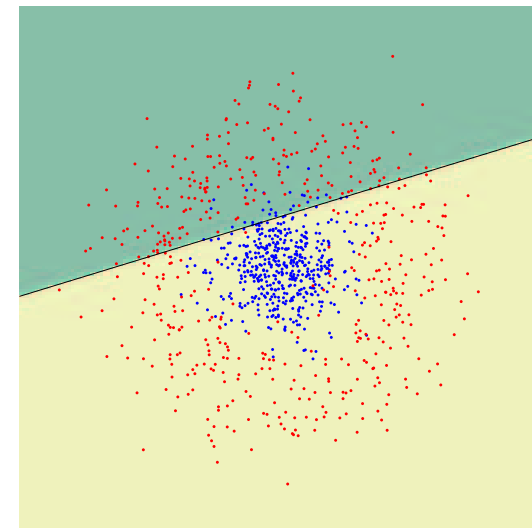
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \llbracket y_i \neq h_j(x_i) \rrbracket$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 1$



The Algorithm Recapitulation

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

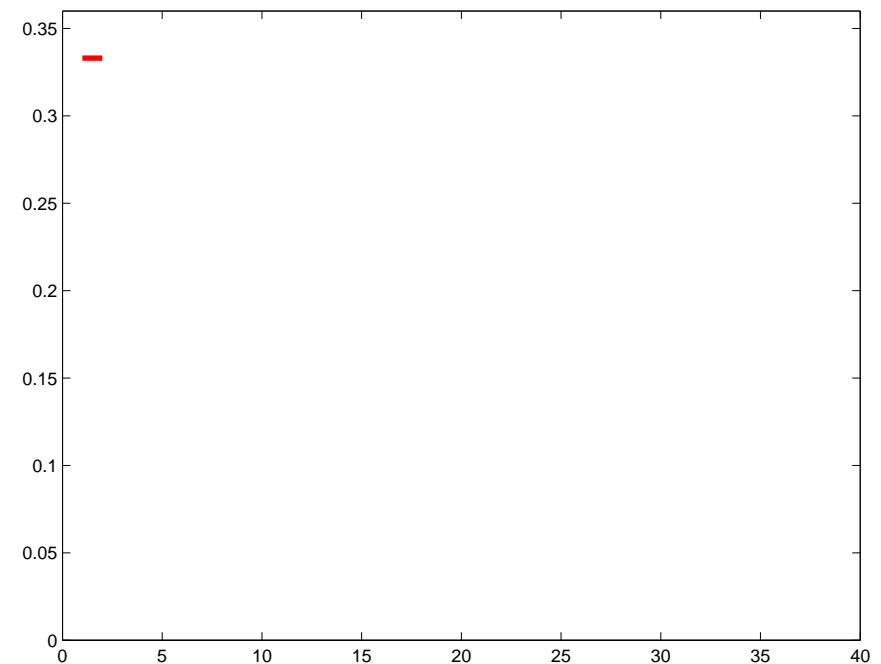
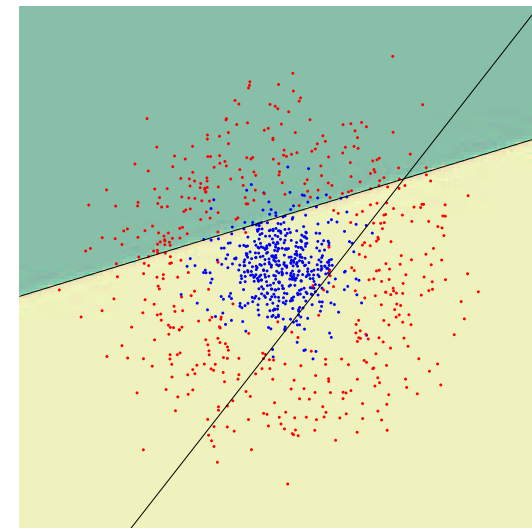
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \llbracket y_i \neq h_j(x_i) \rrbracket$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 2$



The Algorithm Recapitulation

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

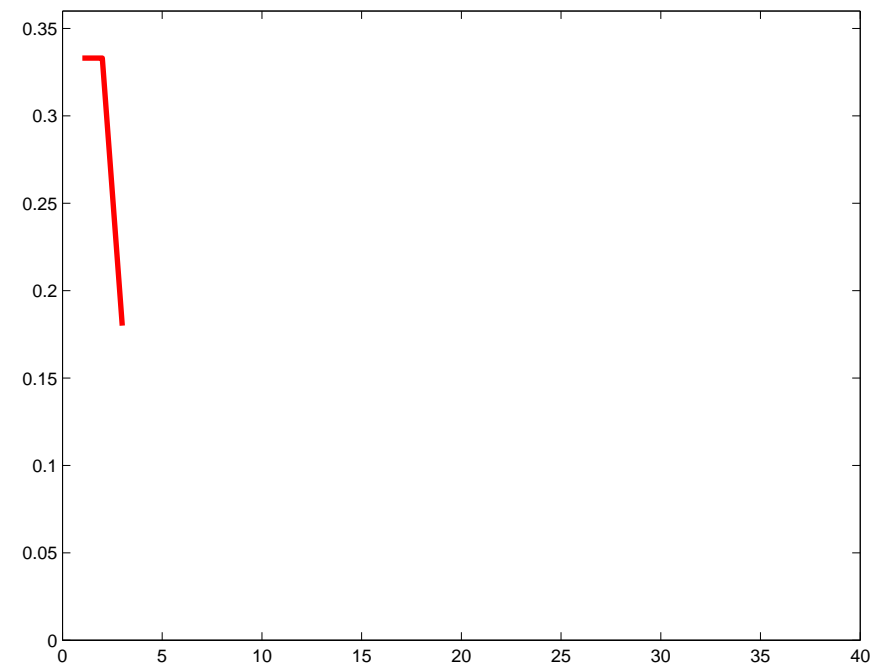
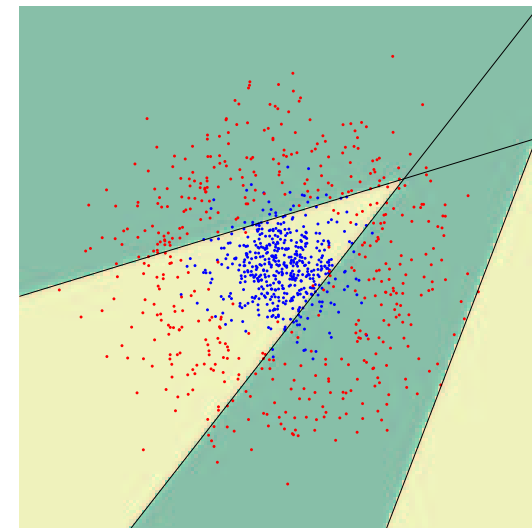
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \llbracket y_i \neq h_j(x_i) \rrbracket$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 3$



The Algorithm Recapitulation

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

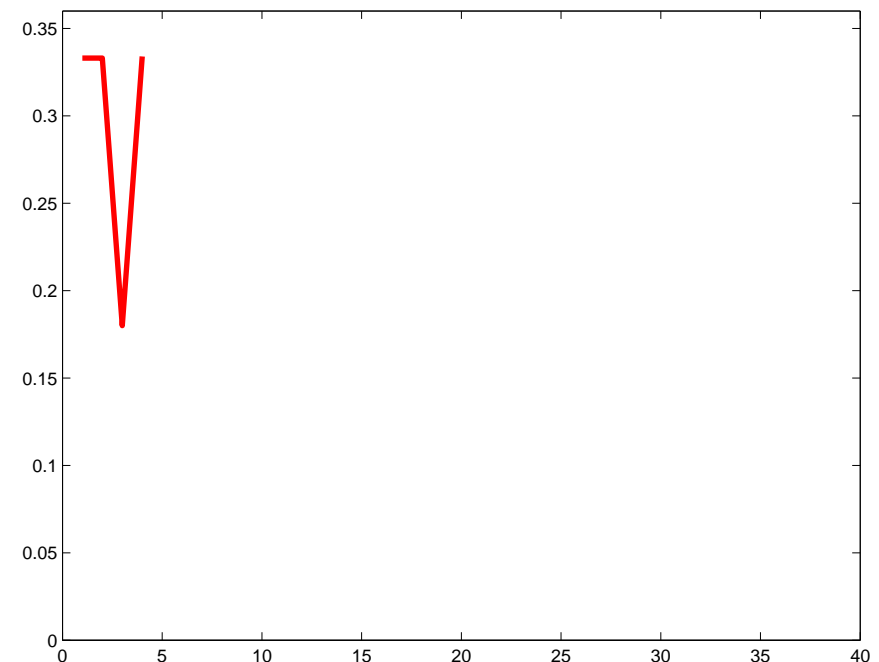
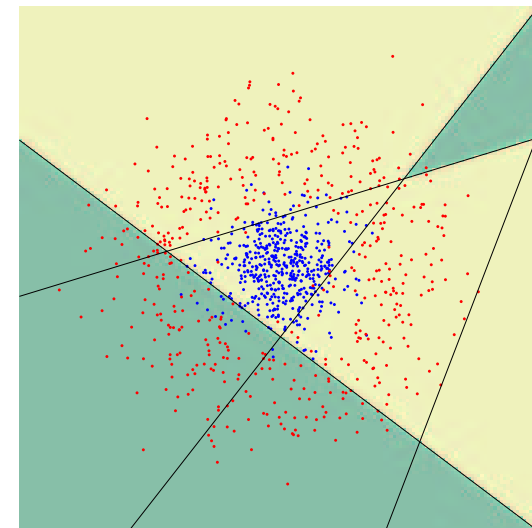
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \llbracket y_i \neq h_j(x_i) \rrbracket$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 4$



The Algorithm Recapitulation

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

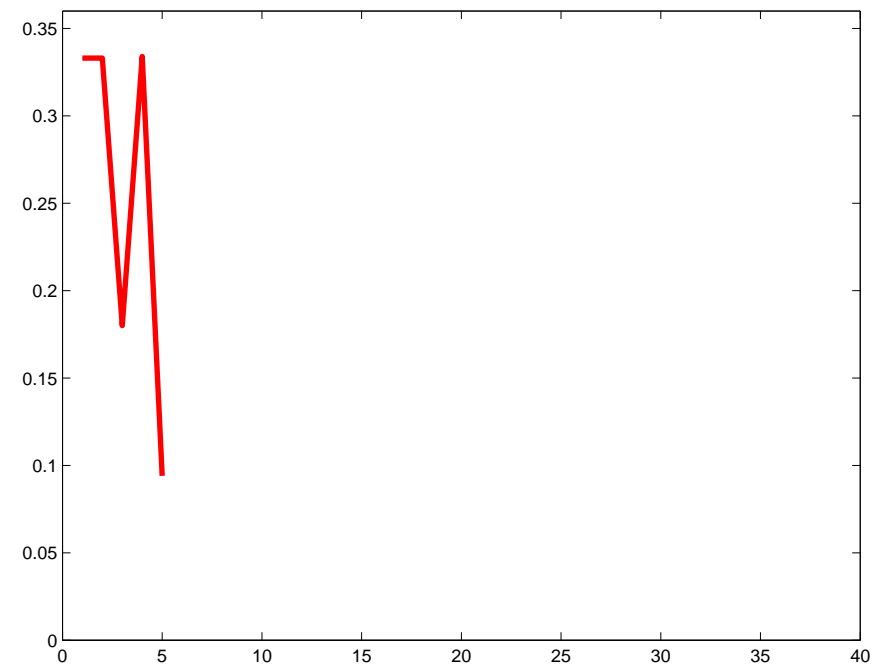
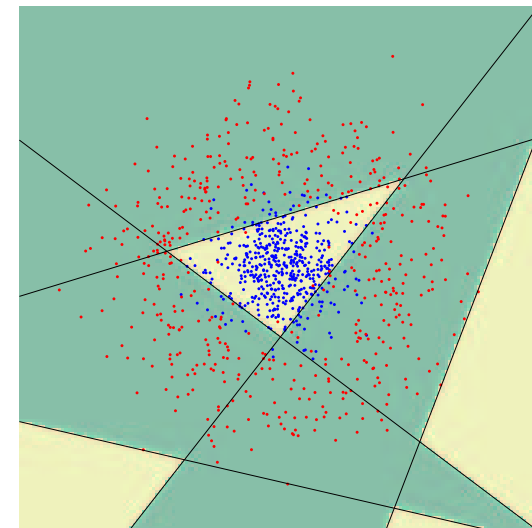
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \llbracket y_i \neq h_j(x_i) \rrbracket$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 5$



The Algorithm Recapitulation

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

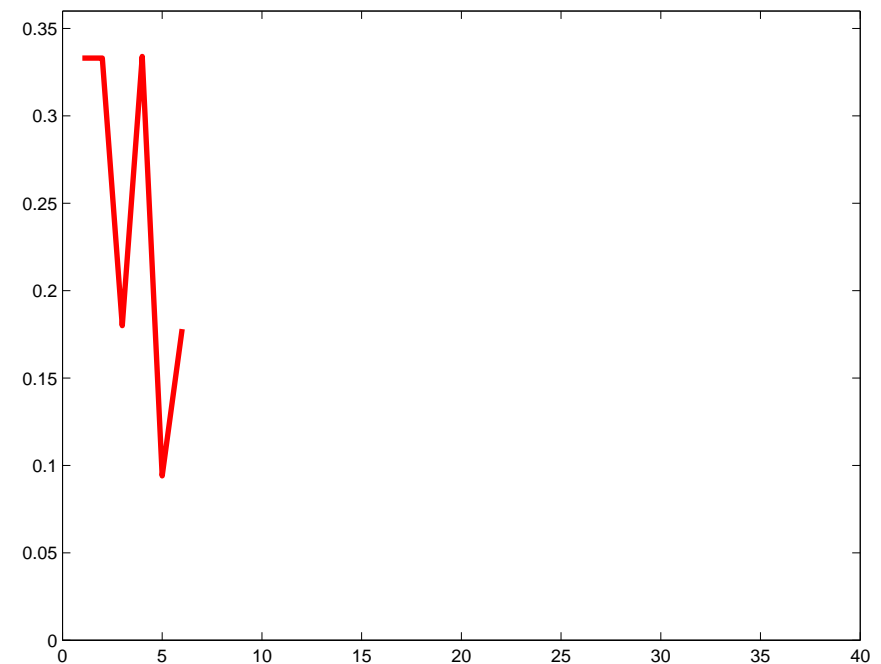
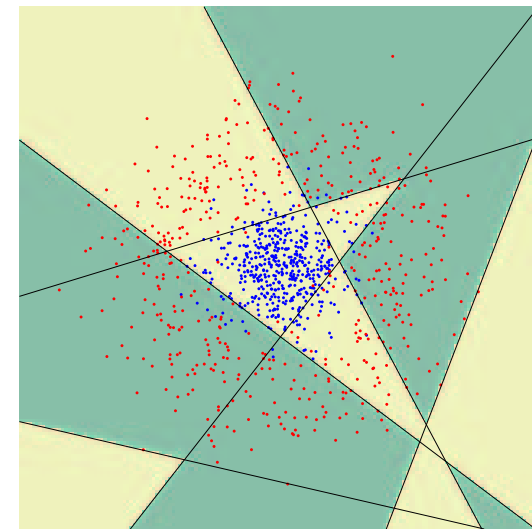
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 6$



The Algorithm Recapitulation

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

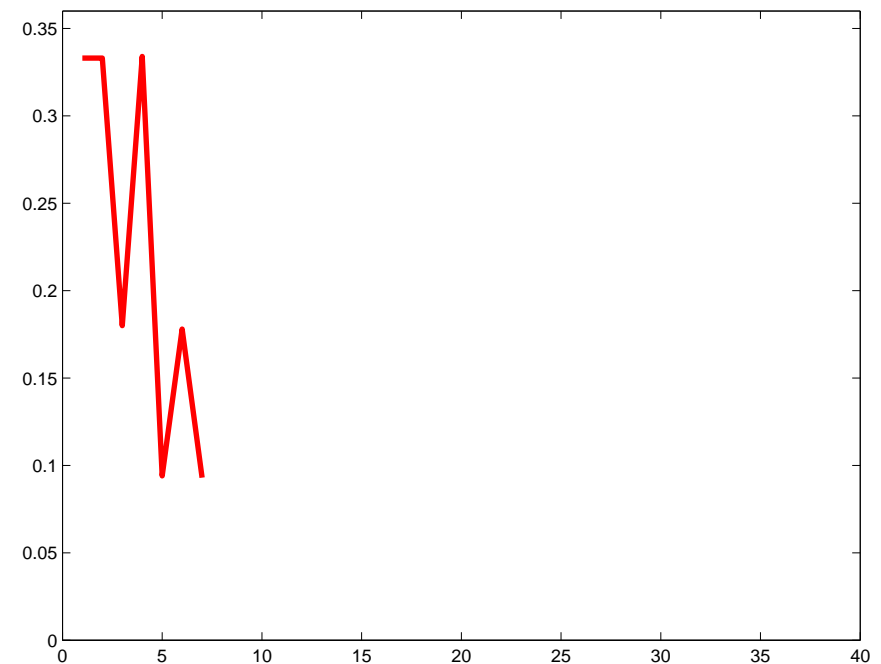
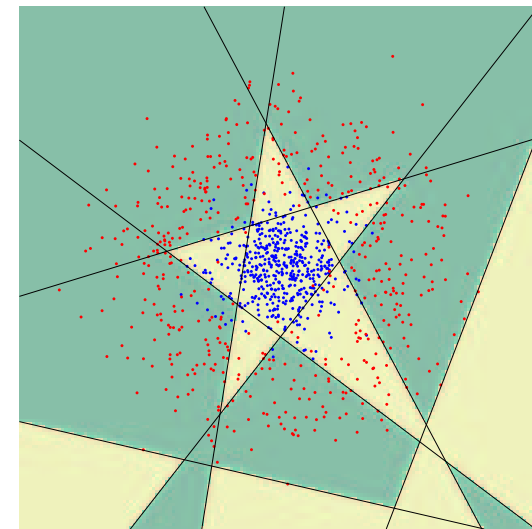
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \mathbb{I}[y_i \neq h_j(x_i)]$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 7$



The Algorithm Recapitulation

Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

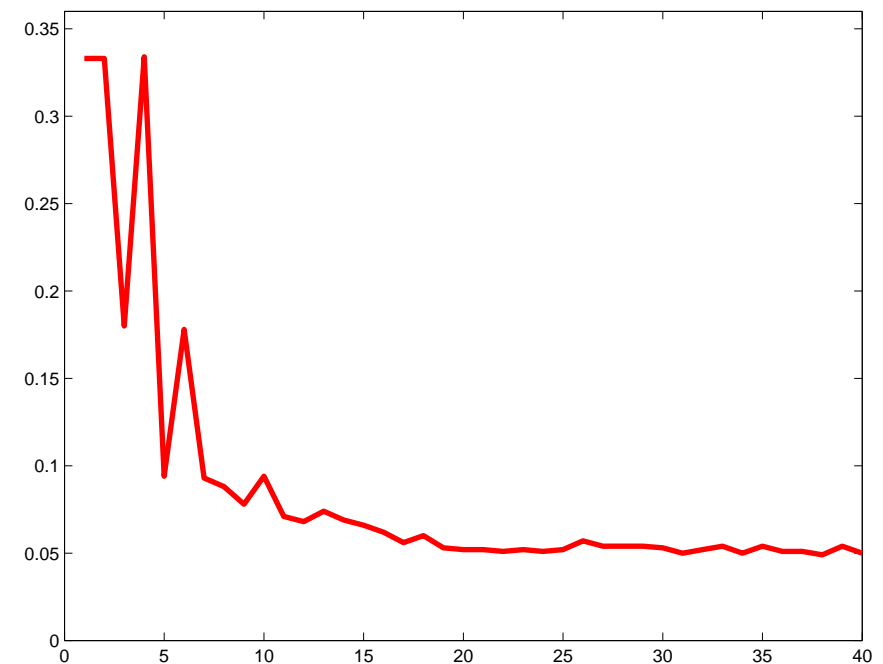
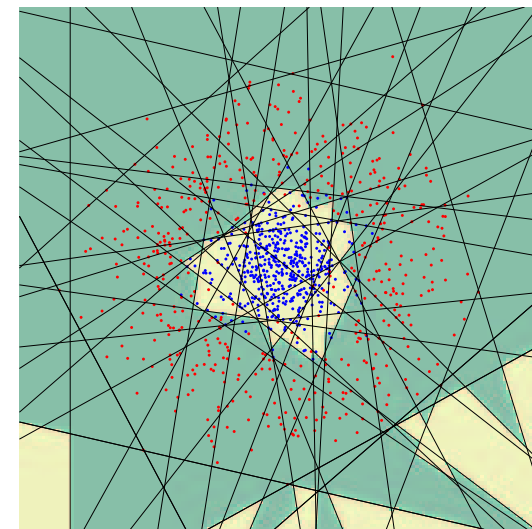
- ◆ Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^m D_t(i) \llbracket y_i \neq h_j(x_i) \rrbracket$
- ◆ If $\epsilon_t \geq 1/2$ then stop
- ◆ Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- ◆ Update

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

Output the final classifier:

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

$t = 40$



AdaBoost variants

Freund & Schapire 1995

- ◆ Discrete ($h : \mathcal{X} \rightarrow \{0, 1\}$)
- ◆ Multiclass AdaBoost.M1 ($h : \mathcal{X} \rightarrow \{0, 1, \dots, k\}$)
- ◆ Multiclass AdaBoost.M2 ($h : \mathcal{X} \rightarrow [0, 1]^k$)
- ◆ Real valued AdaBoost.R ($Y = [0, 1], h : \mathcal{X} \rightarrow [0, 1]$)

Schapire & Singer 1997

- ◆ Confidence rated prediction ($h : \mathcal{X} \rightarrow R$, two-class)
- ◆ Multilabel AdaBoost.MR, AdaBoost.MH (different formulation of minimised loss)

... Many other modifications since then (WaldBoost, cascaded AB, online AB, ...)

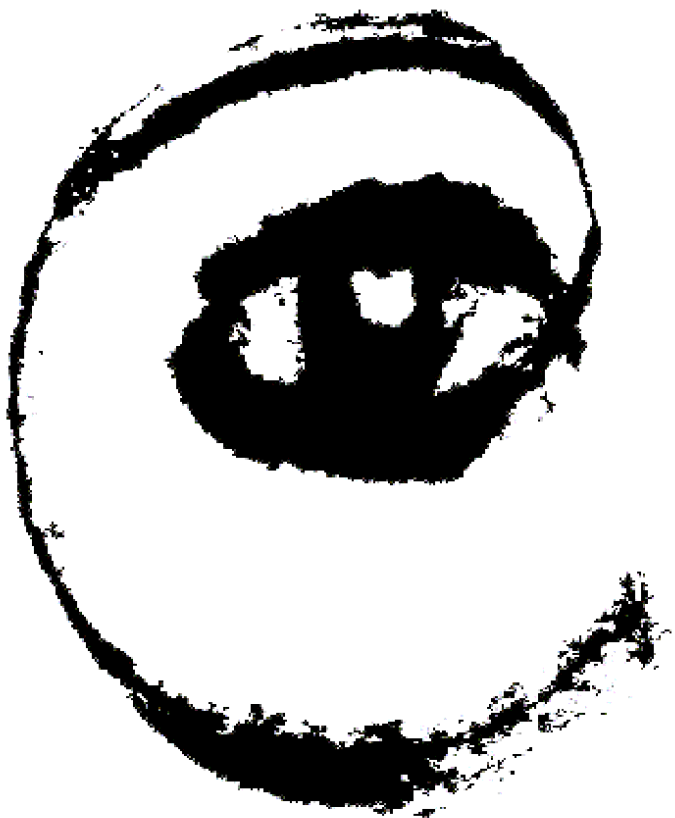
Pros and cons of AdaBoost

Advantages

- ◆ Very simple to implement
- ◆ General learning scheme - can be used for various learning tasks
- ◆ Feature selection on very large sets of features
- ◆ Fairly good generalisation

Disadvantages

- ◆ Suboptimal solution (greedy learning)
- ◆ Can overfit in presence of noise (has been addressed recently)



m p

