Machine learning: Boosting

1. Basic definition:

Assume again we have a classification task (e.g. cancer classification) with data

$$D = \{\mathbf{x}_i, y_i\}_i.$$

and $y_i = \pm 1$.

Boosting

Assume we have a classifier $p(\mathbf{x})$ which takes feature vector $\mathbf{x}$, and classify

$$\begin{cases} y = 1 & \text{if } p(\mathbf{x}) > 0 \\ y = -1 & \text{if } p(\mathbf{x}) < 0 \end{cases}.$$

Assume $p(\mathbf{x})$ is 'weak' - i.e., its predictions are not always correct.

Now assume a family $\{p_j(\mathbf{x})\}_{j=1}^{m}$ of different weak classifiers.

Boosting

A *boosting classifier* takes a linear combination of these classifiers to form a better one, for example,

$$f(\mathbf{x}) = \sum_{i=1}^{m} a_i p_i(\mathbf{x}),$$

where, e.g., $f(\mathbf{x}) \geq 0$ selects the $+$ class, and otherwise the $-$ class.

2. AdaBoost (adaptive boosting) algorithm (Freund & Schaphire, 1997)

Consider data $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$. Weigh each data point $(\mathbf{x}_i, y_i)$ equally with weight $W_1(i) = \frac{1}{n}$.

Assume $\mathbf{x}_i \in F =$ feature space.

Goal: build a classifier $f(\mathbf{x})$ which generalizes data set D to predict class y of $\mathbf{x}$.

Let $\mathcal{H} =$ space of allowed classifier functions f

AdaBoost: introduction

Idea: We will take random samples from data set D (with repetition) according the weight distribution W_1, and later some distributions W_2, $W_3, \dots$ which emphasize the examples $\mathbf{x}_i$ we have misclassified previously.

Specifically: train initial classifier $h_1(\mathbf{x}) : \mathbf{x} \rightarrow \{\pm 1\}$ which minimizes error with respect to weight distribution W_1.

That is:

AdaBoost: introduction

h_1 = function h which minimizes weighted number of errors

= function h which minimizes error $\sum_{i=1}^{n} W_1(i) I(y_i \neq h(\mathbf{x}_i))$

$$= \operatorname*{argmin}_{h \in \mathcal{H}} \sum_{i=1}^{n} W_1(i) I(y_i \neq h(\mathbf{x}_i)) \quad (1)$$

where

$$I(y_i \neq h(\mathbf{x}_i)) = \begin{cases} 1 & \text{if } y_i \neq h(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases}.$$

3. The algorithm:

1. Define

$\epsilon_1 =$ minimal *value* of error in (1).

(i.e. smallest value of the sum (1))

Require: $\epsilon_1 \leq .5$; otherwise stop (then have a bad family of classifiers h; need at least 50% accuracy).

AdaBoost algorithm

2. Choose α_1; typically

$$\alpha_1 = \frac{1}{2} \ln \frac{1 - \epsilon_1}{\epsilon_1} \tag{2}$$

3. Update weigths W:

$$W_2(i) = \frac{W_1(i) e^{-\alpha_1 y_i h_1(i)}}{Z_1(\alpha_1)},$$

where $Z_1 =$ normalizing constant to make $\{W_2(i)\}_i$ a probability distribution (over i) which adds up to 1.

AdaBoost algorithm

Now $\{W_2(i)\}_i$ form a family of weights which we can use to 're-sample' the data set D, and form a new classifier h_2.

Specifically

$$h_2 = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{n} W_2(i) I(y_i \neq h(\mathbf{x}_i))$$

(note that if weight $W_2(i) > \frac{1}{n}$ this is equivalent to 'oversampling' data point $\mathbf{x}_i$; otherwise 'undersampling' $\mathbf{x}_i$)

AdaBoost algorithm

4. Generally, define

$$h_t = \operatorname*{argmin}_{h \in \mathcal{H}} \sum_{i=1}^{n} W_t(i) I(y_i \neq h(\mathbf{x}_i)) \qquad (2a)$$

Again let

$\epsilon_t =$ smallest value of the sum in (2a)

and

$$\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}. \qquad (3)$$

AdaBoost algorithm

Then define

$$W_{t+1}(i) = \frac{W_t(i)e^{-\alpha_t y_i h_t(i)}}{Z_t}, \qquad \text{(3a)}$$

and

Z_t = normalizing constant as earlier.

AdaBoost algorithm

Then after T steps form final classifier

$$f(\mathbf{x}) = \mathrm{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right).$$

4. Observations about AdaBoost

Note that the updating equation (3a) has property

$$e^{-\alpha_{t-1} y_i h_{t-1}(\mathbf{x}_i)} \begin{cases} < 1 & \text{if } y_i = h_{t-1}(\mathbf{x}_i) \\ > 1 & \text{if } y_i \neq h_{t-1}(\mathbf{x}_i) \end{cases}.$$

Thus after we find best classifier h_{t-1} for sample distribution W_{t-1}, examples $\mathbf{x}_i$ which h_{t-1} identified incorrectly are weighted more for the selection of h_t.

Thus the classifier based on distribution W_t will better identify examples that the previous weak classifier missed.

AdaBoost: additional observations

Recall normalization constant

$$Z_t(\alpha_t) = \sum_i W_t(i) e^{-\alpha_t y_i h_t(i)}. \tag{4}$$

This is weighted measure of the total error at the previous step, since

$$y_i h_t(i) = \begin{cases} 1 & \text{if } h_t(i) = y_i \text{ (correct prediction)} \\ -1 & \text{if } h_t(i) \neq y_i \text{ (incorrect prediction)} \end{cases}$$

which means $e^{-\alpha_t y_i h_t(i)}$ large if $h_t(i) \neq y_i$.

AdaBoost: additional observations

Can show our definition (3) for α_t minimizes $Z_t(\alpha_t)$ at each step (just use calculus to minimize (4) above w/ respect to the variable α_t).

Note: can also show

$$Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$$

for above choice of α_t.

Recall reweighting formula:

AdaBoost: additional observations

$$W_{t+1}(i) = \frac{W_t(i)e^{-\alpha_t y_i h_t(\mathbf{x}_i)}}{Z_t} \stackrel{\text{recursion}}{=} \frac{e^{-\left(y_i \sum_{q=1}^{t} \alpha_q h_q(\mathbf{x}_i)\right)}}{n \prod_{q=1}^{t} Z_q}.$$

(recall initial value $D_1(i) = \frac{1}{n}$).

AdaBoost: additional observations

Can also show: fact that α_t are chosen to minimize Z_t implies that

$$\sum_{i:h_t(\mathbf{x}_i)=y_i} W_{t+1}(i) = \sum_{i:h_t(\mathbf{x}_i)\neq y_i} W_{t+1}(i);$$

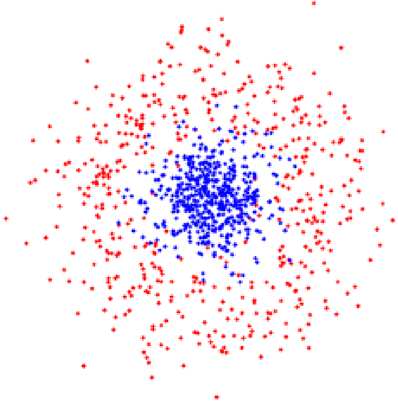
this shows that half the *new* weight is focused on the misclassified examples for previous classifier h_t (where $h_t(\mathbf{x}_i) \neq y_i$).

AdaBoost: examples

Example 1:

Training set:

AdaBoost: examples



J. Matas, J. Sochman

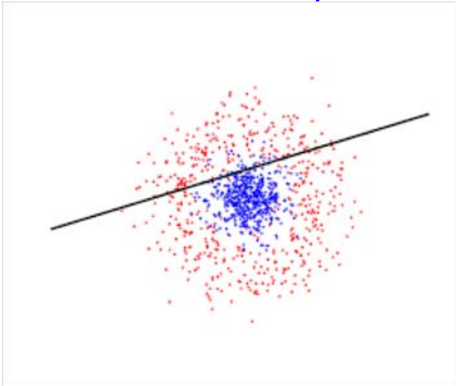
Blue: $N(0,1)$ **Red:** $\frac{1}{r\sqrt{8r^2}}e^{-1/2(r-4)^2}$

AdaBoost: examples

(distribution is radial with uniform angular distribution)

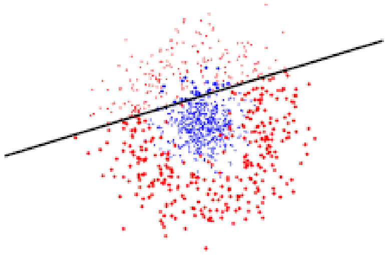
Select weak (always linear) classifier with smallest weighted error (all points equal at this time). $W_1(i) = \frac{1}{n}$.

AdaBoost: examples



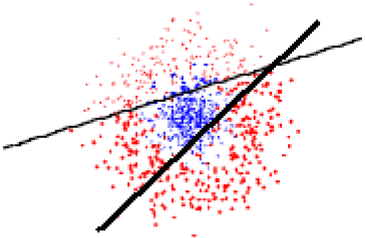
Re-weighting formula: find $W_2(i)$. Give more weight to misclassified examples;

AdaBoost: examples



AdaBoost: examples

new classifer (focuses more on higher weights):



Reweigh again based on previously misclassified examples, etc.

AdaBoost: examples

Summary: first classifier at $t = 1$ and error rate:

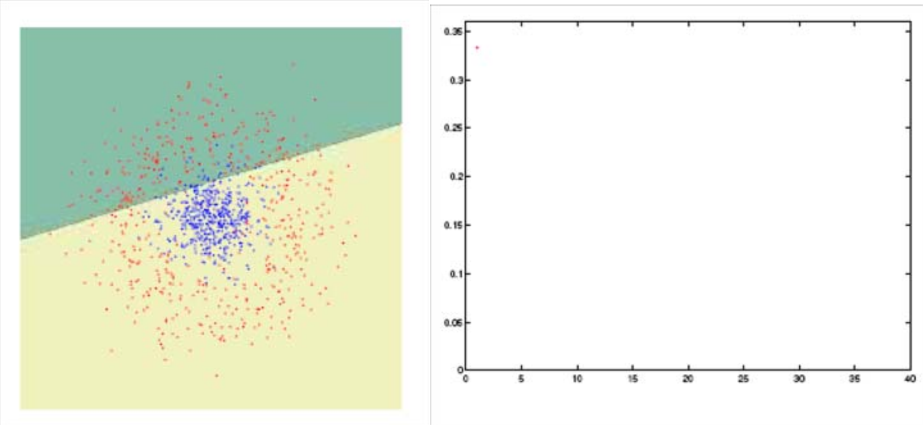
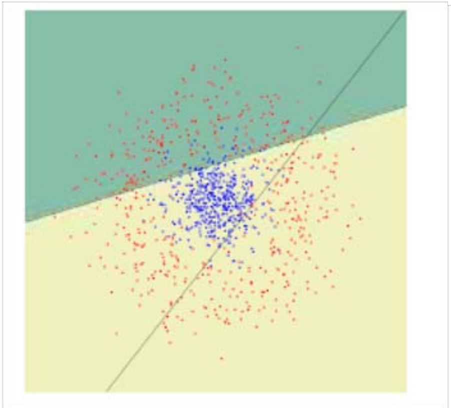
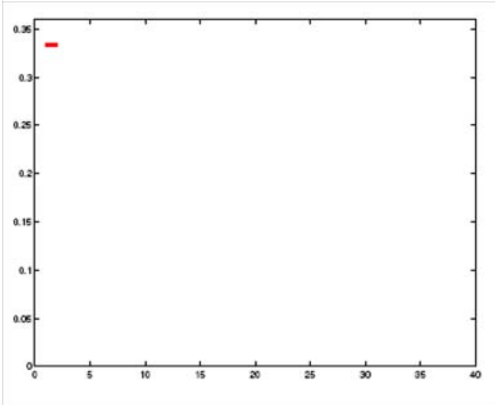


Fig. 1: Here and below shaded part is currently classified red (figures due to J. Matas, J. Sochman)

AdaBoost: examples

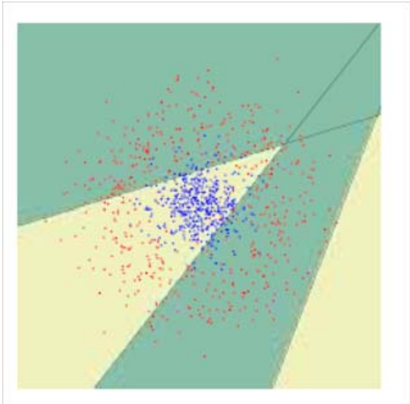
Second classifier: $t = 2$ and error rate:

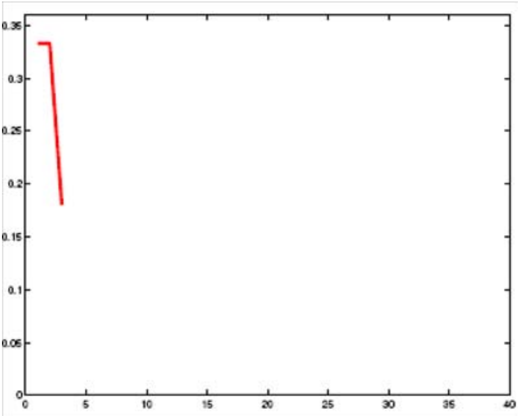




AdaBoost: examples

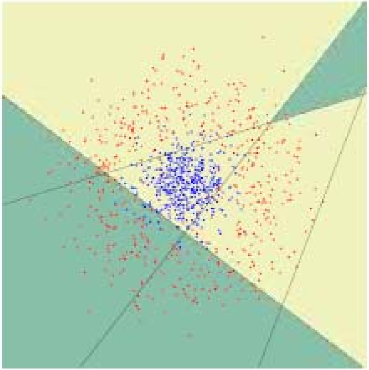
$t = 3:$

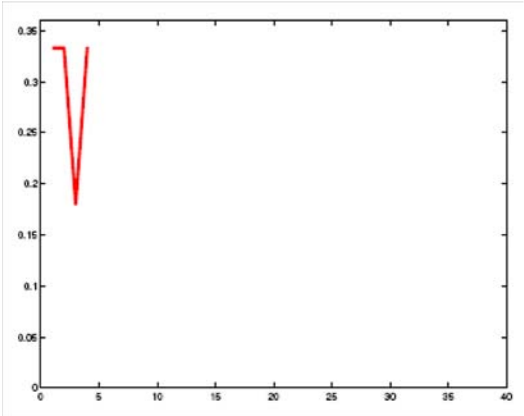




AdaBoost: examples

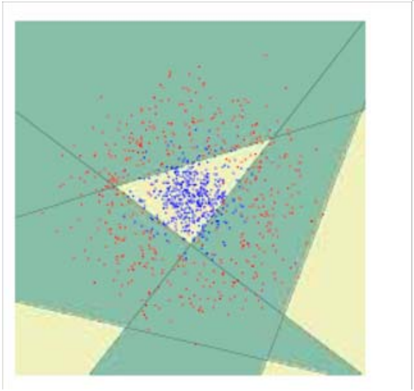
$t = 4:$

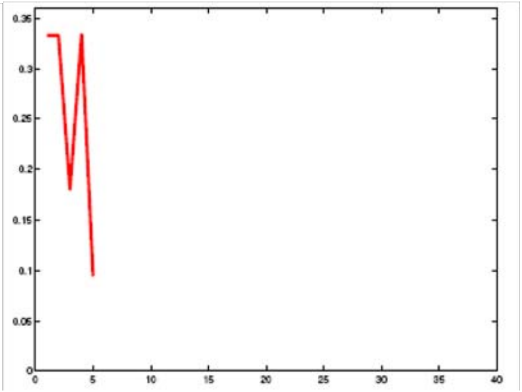




AdaBoost: examples

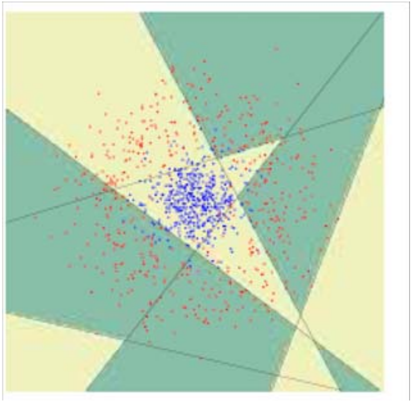
$t = 5:$

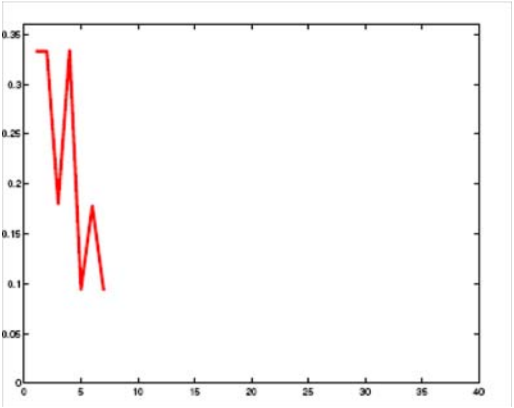




AdaBoost: examples

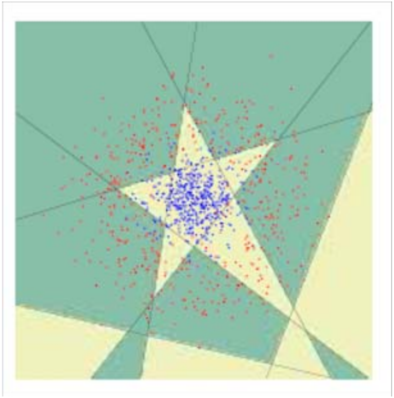
$t = 6:$

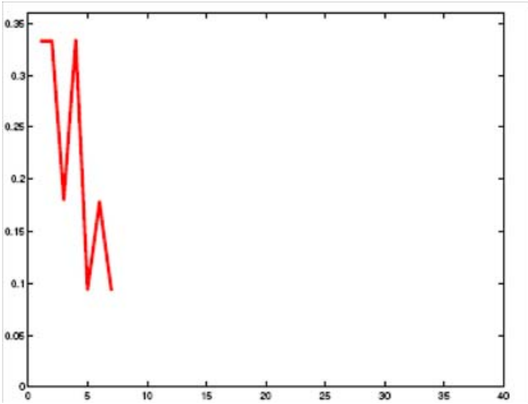




AdaBoost: examples

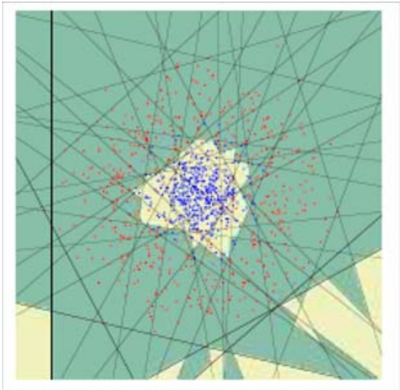
$t = 7:$

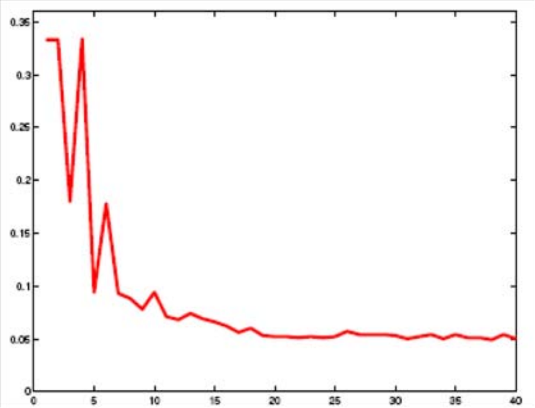




AdaBoost: examples

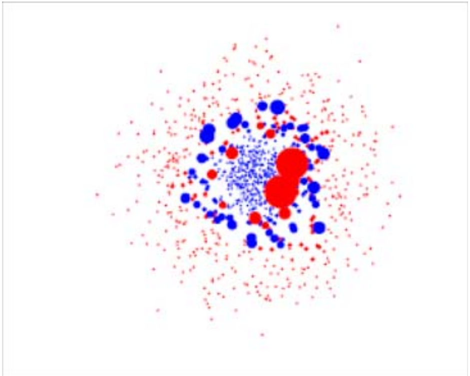
$t = 40$:





AdaBoost: examples

Note misclassified examples closest to the boundary are the ones which at end get highest weight:



AdaBoost: examples

Data from the IDA repository (Ratsch:2000):

	Input dimension	Training patterns	Testing patterns	Number of realizations
Banana	2	400	4900	100
Breast cancer	9	200	77	100
Diabetes	8	468	300	100
German	20	700	300	100
Heart	13	170	100	100
Image segment	18	1300	1010	20
Ringnorm	20	400	7000	100
Flare solar	9	666	400	100
Splice	60	1000	2175	20
Thyroid	5	140	75	100
Titanic	3	150	2051	100
Twonorm	20	400	7000	100
Waveform	21	400	4600	100

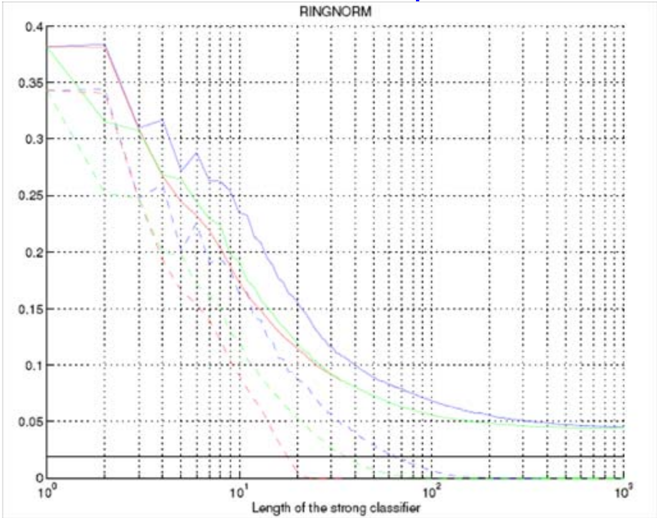
Baseline: AdaBoost with RBF network weak classifiers from (Ratsch-ML: 2000).

AdaBoost: examples

Blue: Adaboost (discrete)
Green: Continuous Adaboost
Red: Adaboost with totally corrective step (TCA)
Cyan: Continuous Adaboost with totally corrective step

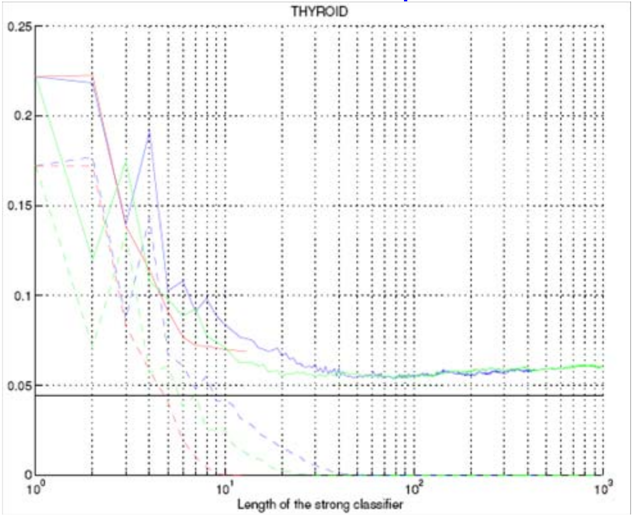
Dashed lines: training (leave one out) error; solid - test error

AdaBoost: examples



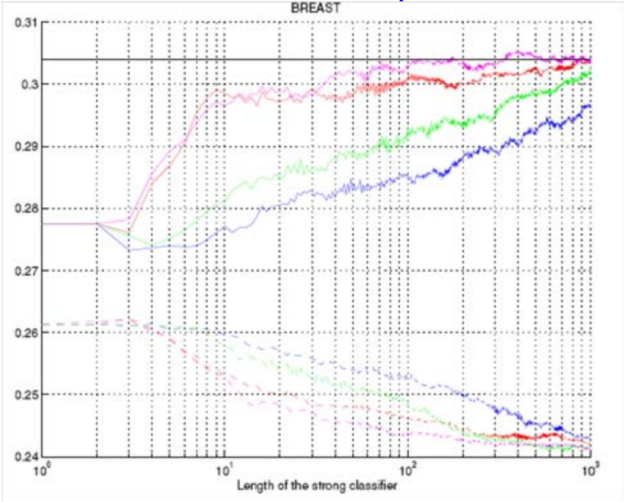
Above dual distribution example

AdaBoost: examples



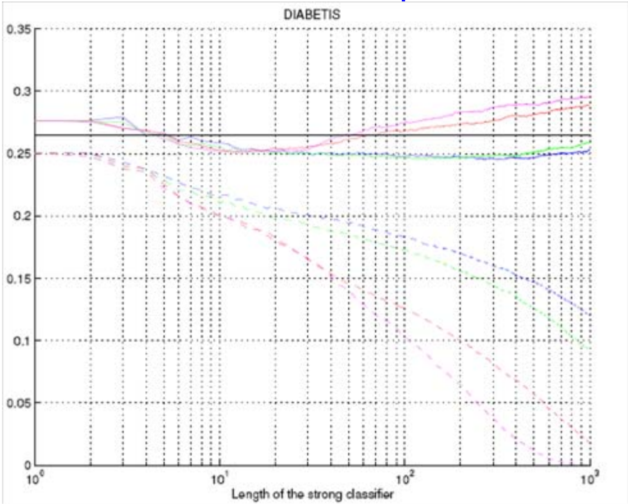
(Serum antibodies as predictors of thyroiditis)

AdaBoost: examples



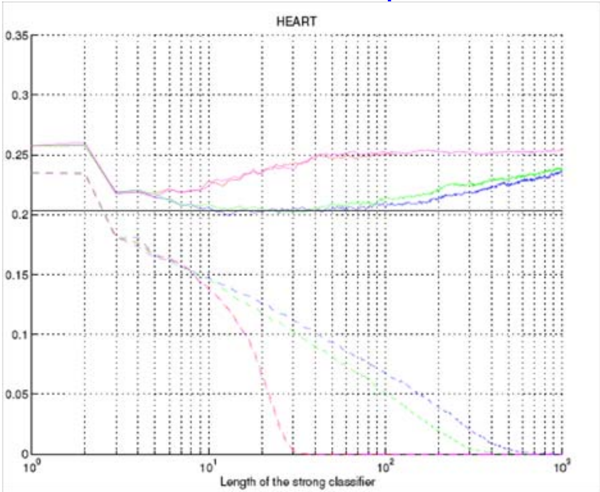
Breast cancer classification

AdaBoost: examples



Diabetes from blood markers

AdaBoost: examples



Heart disease from blood markers

All figures: J. Matas and J. Sochman (2005)

AdaBoost: examples

Algorithm	Gene limit	ALL-AML	HCC	ER	Colon	LN	Brain
Adaboost	10	6.2	7.8	19.9	25.3	40.4	42.3
Adaboost-VC	10	3.9	5.6	18.1	24.4	43.8	41.1
Adaboost-NR	10	3.5	6.0	19.5	25.1	42.7	41.2
Adaboost-PL	10	7.0	7.2	20.6	23.4	36.5	41.9
Arc-x4-RW	10	6.5	8.2	19.8	25.0	39.1	41.4
Arc-x4-RW-NR	10	3.3	5.5	17.8	24.7	42.1	40.7
SVM-RFE	10	13.4	8.6	20.9	19.2	48.4	39.2
Wilcoxon/SVM	10	6.4	6.7	23.2	24.3	35.4	39.3
Adaboost	100	5.2	6.9	16.1	23.4	35.4	38.2
Adaboost-VC	100	2.8	4.8	13.8	22.6	42.8	38.2
Adaboost-NR	100	2.7	4.9	13.2	21.9	40.6	36.5
Adaboost-PL	100	5.0	5.4	17.2	23.2	36.2	38.6
Arc-x4-RW	100	5.4	7.4	16.6	23.7	36.9	38.0
Arc-x4-RW-NR	100	2.6	4.8	12.8	21.6	41.1	36.1
SVM-RFE	100	6.5	6.7	12.6	20.7	48.1	35.7
Wilcoxon/SVM	100	3.3	4.1	17.5	23.6	40.4	37.8

Long and Vega (2003)

Adaboost vs. other classifiers in microarray cancer diagnosis (cross-validation error rates)

AdaBoost: examples

HCC = Hepatocellular carcinoma

ER = estrogen response in breast cancer

LN = lymph node spread of cancer