Ensembles

Léon Bottou

COS 424 – 4/8/2010

Readings

- T. G. Dietterich (2000)
  "Ensemble Methods in Machine Learning".

- R. E. Schapire (2003):
  "The Boosting Approach to Machine Learning".
  Sections 1,2,3,4,6.

Summary

1. Why ensembles?
2. Combining outputs.
3. Constructing ensembles.
4. Boosting.

I. Ensembles

Ensemble of classifiers

Ensemble of classifiers

– Consider a set of classifiers $h_1, h_2, \ldots, h_L$.
– Construct a classifier by combining their individual decisions.
– For example by voting their outputs.

Accuracy

– The ensemble works if the classifiers have low error rates.

Diversity

– No gain if all classifiers make the same mistakes.
– What if classifiers make different mistakes?

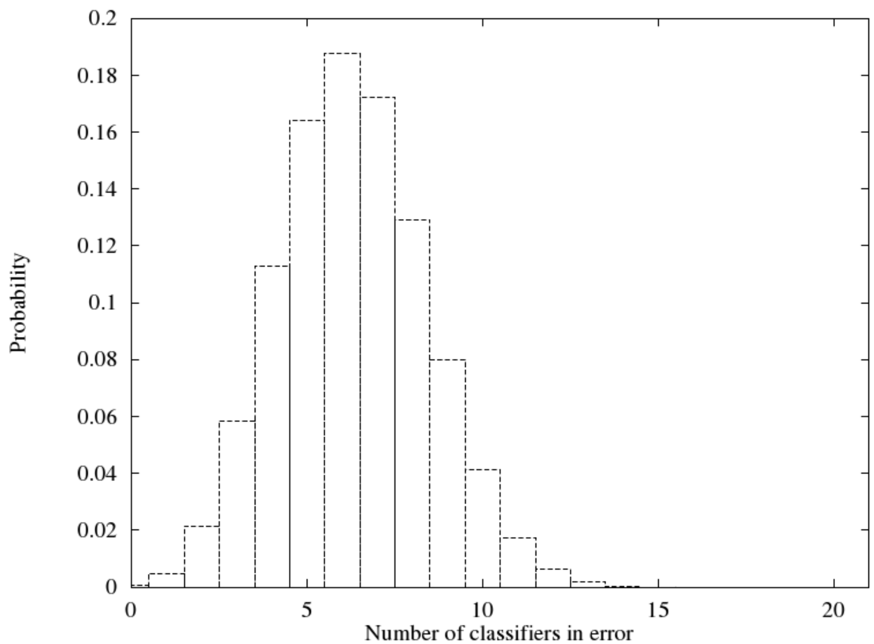
Uncorrelated classifiers

Assume $\forall r \neq s \quad \text{Cov}\left[\, \mathbb{1}\{h_r(x) = y\}\,,\; \mathbb{1}\{h_s(x) = y\}\,\right] = 0$

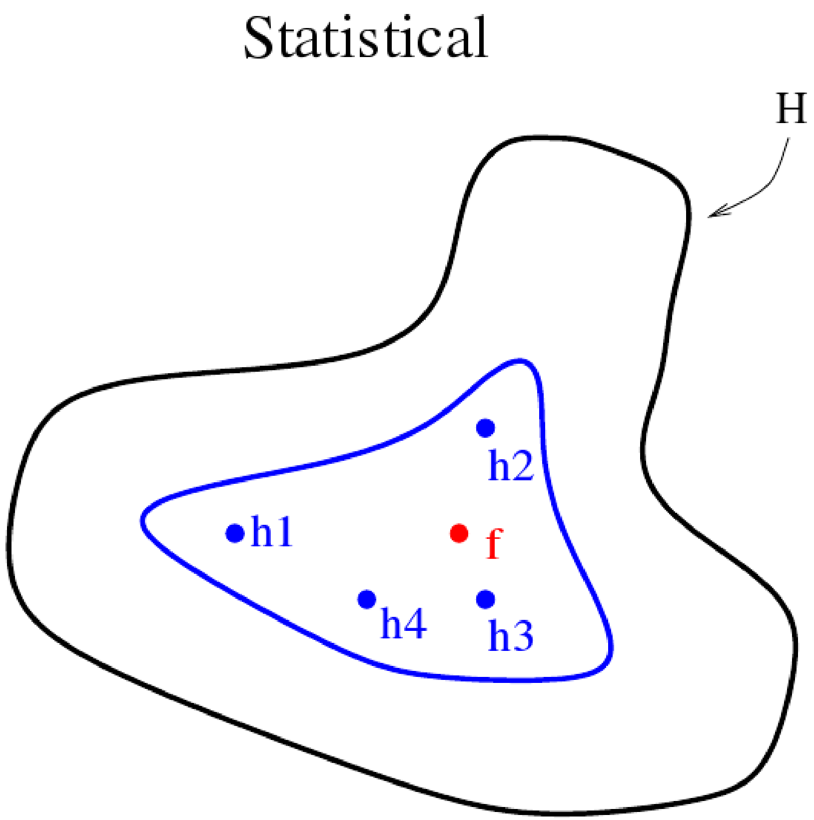
The tally of classifier votes follows a binomial distribution.

Example

Twenty-one uncorrelated classifiers with 30% error rate.

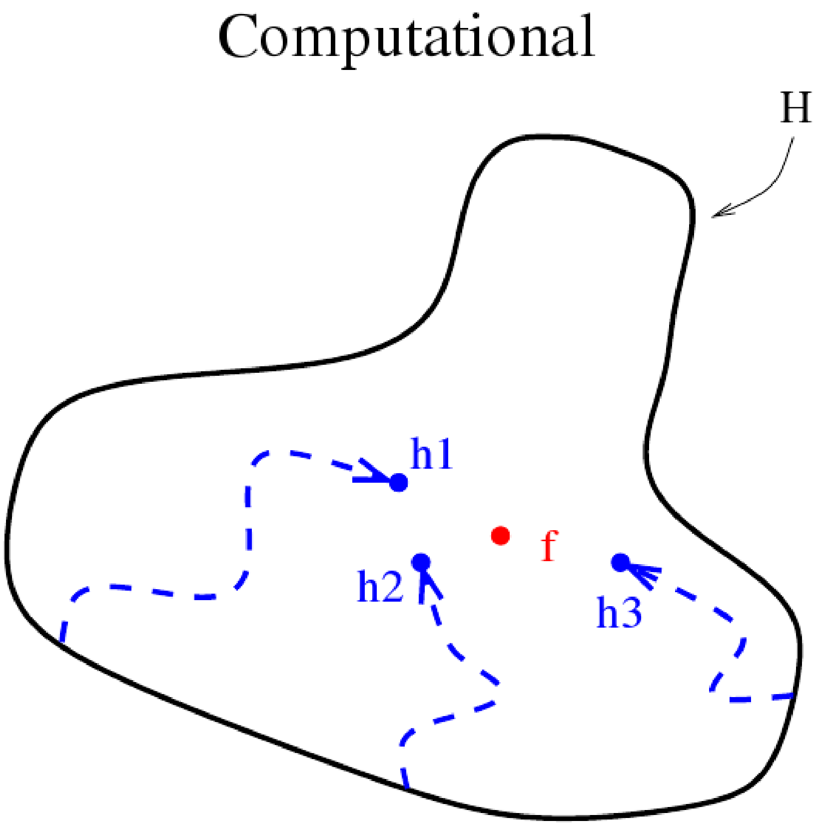


Statistical motivation



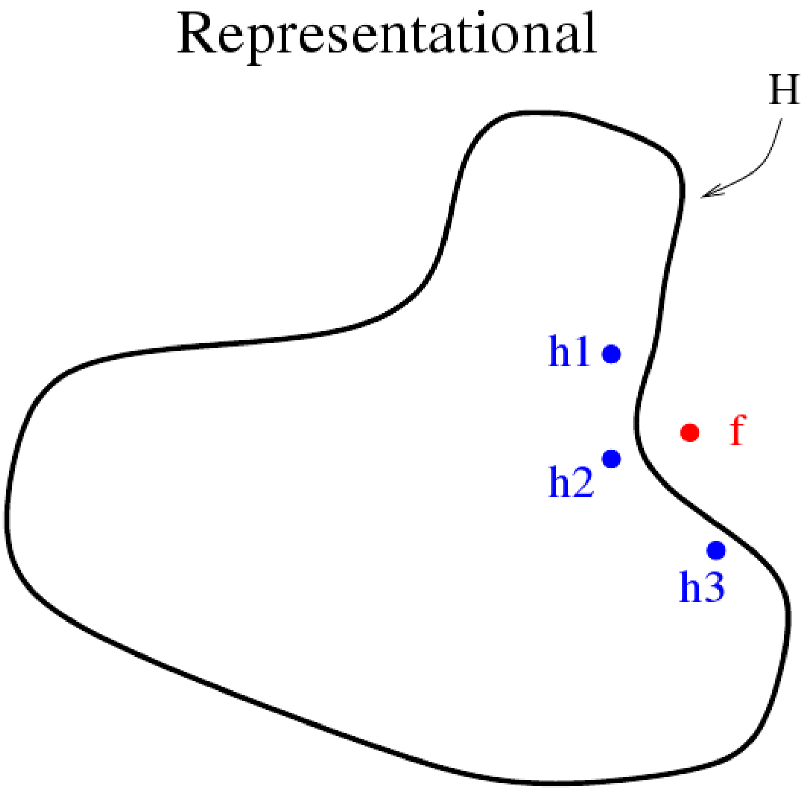
blue : classifiers that work well on the training set(s)
f : best classifier.

Computational motivation



blue : classifier search may reach local optima
f : best classifier.

Representational motivation



blue : classifier space may not contain best classifier
f : best classifier.

Practical success

Recommendation system

– Netflix "movies you may like".

– Customers sometimes rate movies they rent.

– Input: (movie, customer)

– Output: rating

Netflix competition

– 1M$ for the first team to do 10% better than their system.

Winner: BellKor team and friends

– Ensemble of more than 800 rating systems.

Runner-up: everybody else

– Ensemble of all the rating systems built by the other teams.

Bayesian ensembles

Let D represent the training data.

Enumerating all the classifiers

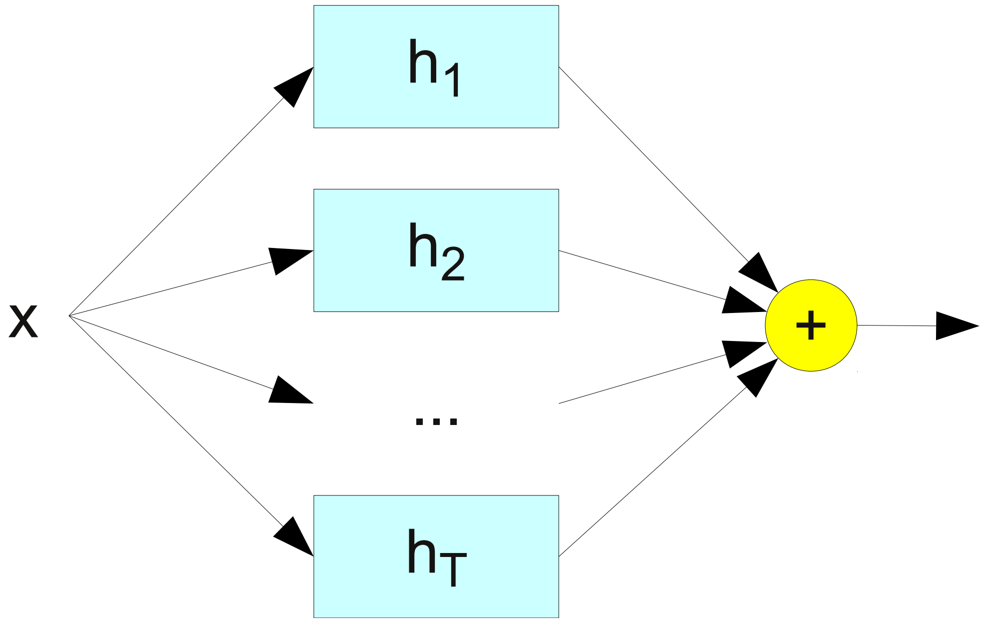
$$\begin{aligned}
P(y|x,D) &= \sum_h P(y,h|x,D) \\
&= \sum_h P(h|x,D)\,P(y|h,x,D) \\
&= \sum_h P(h|D)\,P(y|x,h)
\end{aligned}$$

$P(h|D)$: how well does h match the training data.
$P(y|x,h)$: what h predicts for pattern x.

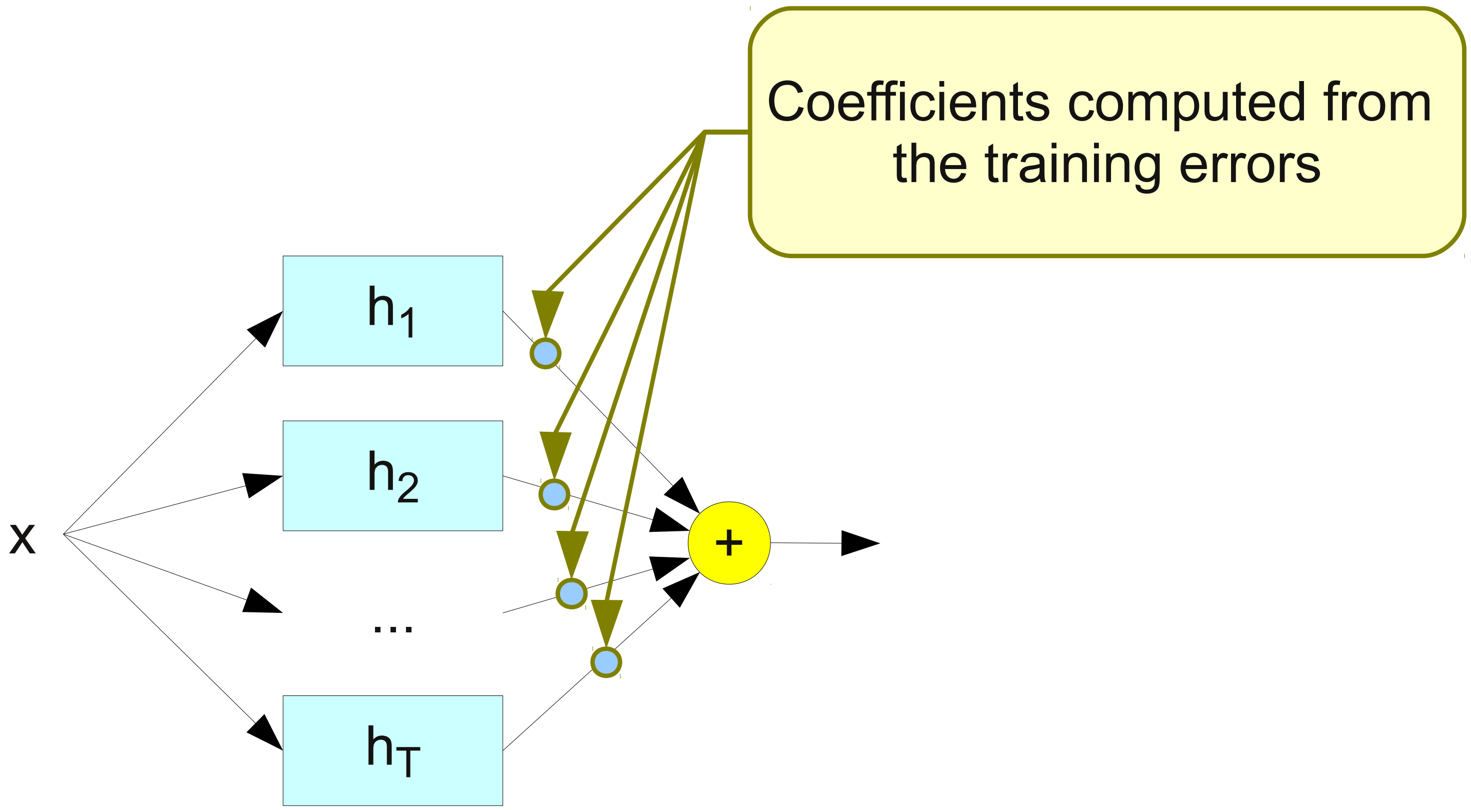
Note that this is a weighted average.

II. Combining Outputs

Simple averaging

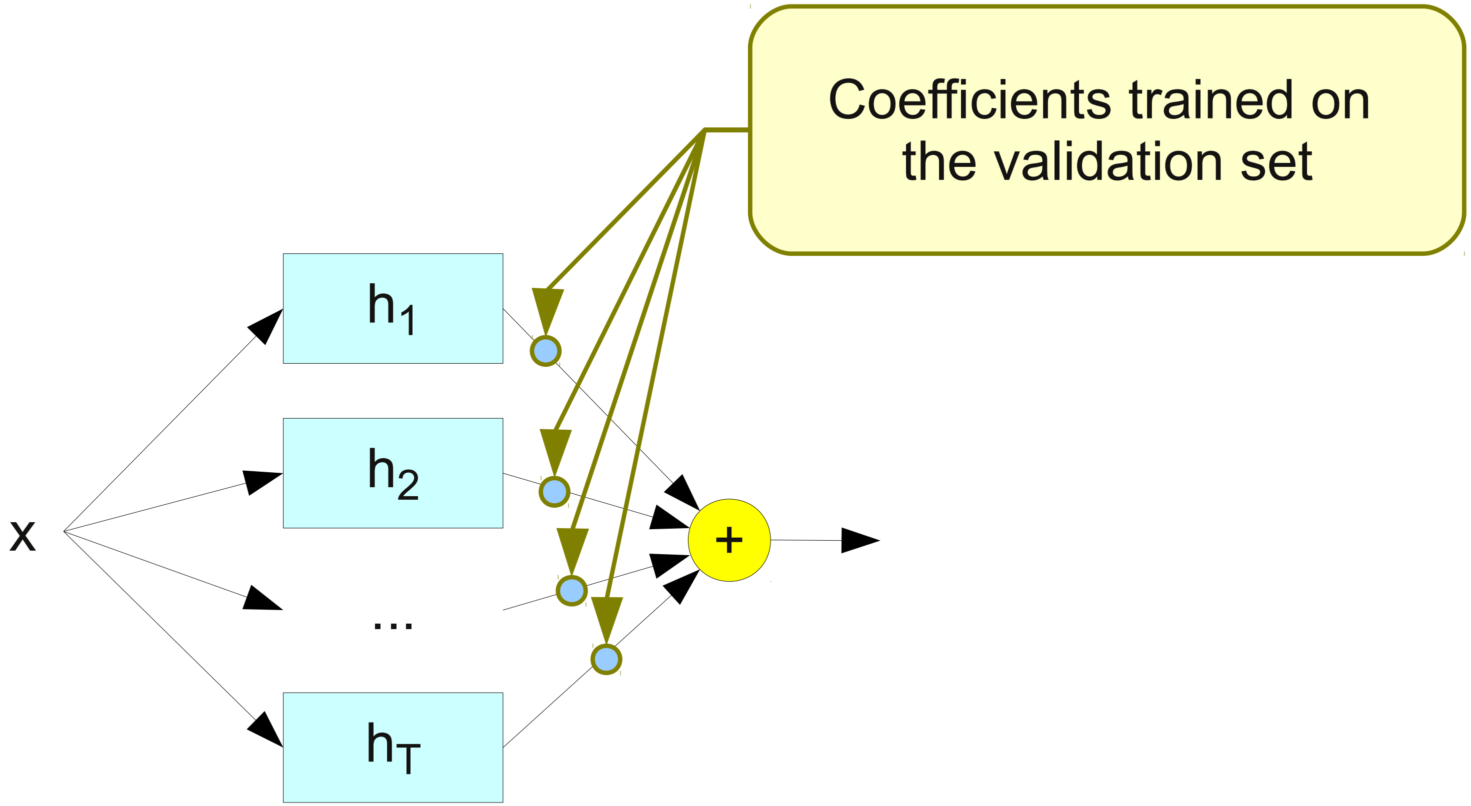


Weighted averaging a priori



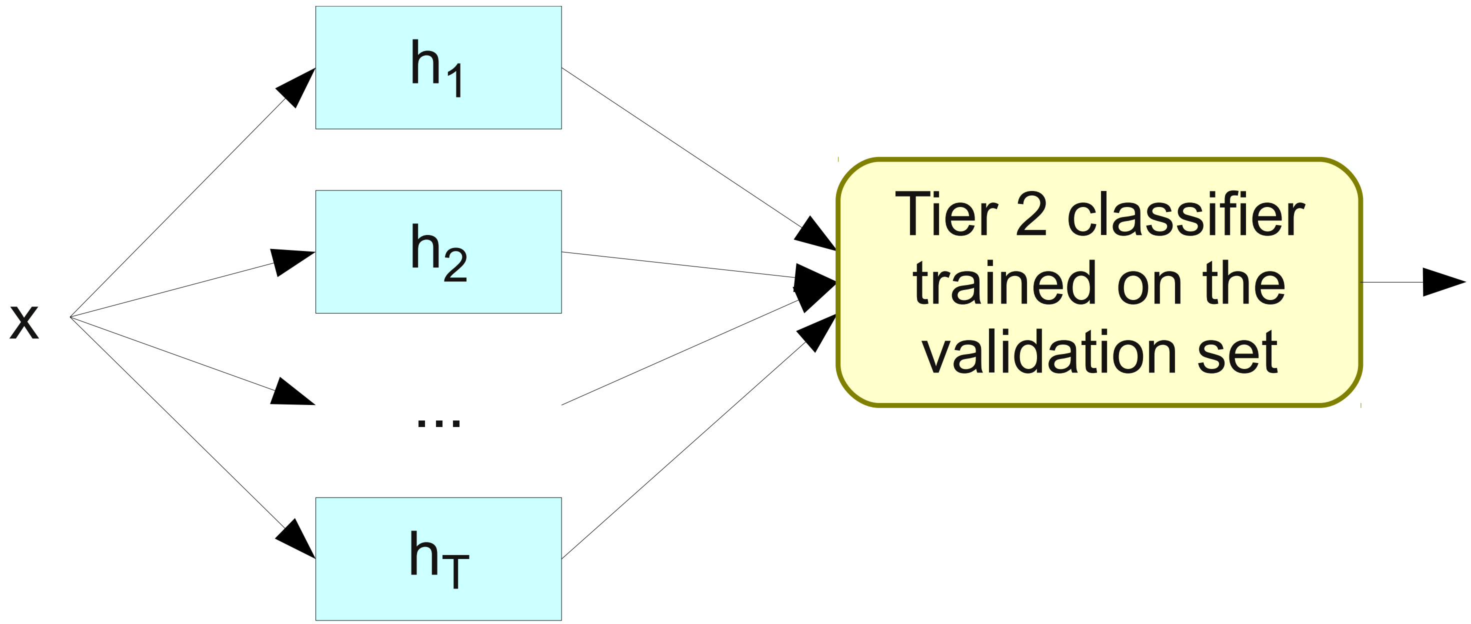
Weights derived from the training errors, e.g. $\exp(-\beta\, TrainingError(h_t))$.
Approximate Bayesian ensemble.

Weighted averaging with trained weights



Train weights on the validation set.
Training weights on the training set overfits easily.
You need another validation set to estimate the performance!

Stacked classifiers



Second tier classifier trained on the validation set.

You need another validation set to estimate the performance!

III. Constructing Ensembles

Diversification

Cause of the mistake	Diversification strategy
Pattern was difficult.	hopeless
Overfitting ($\star$)	vary the training sets
Some features were noisy	vary the set of input features
Multiclass decisions were inconsistent	vary the class encoding

Manipulating the training examples

Bootstrap replication simulates training set selection

– Given a training set of size n, construct a new training set by sampling n examples with replacement.
– About 30% of the examples are excluded.

Bagging

– Create bootstrap replicates of the training set.
– Build a decision tree for each replicate.
– Estimate tree performance using out-of-bootstrap data.
– Average the outputs of all decision trees.

Boosting

– See part IV.

Manipulating the features

Random forests

– Construct decision trees on bootstrap replicas.
  Restrict the node decisions to a small subset of features
  picked randomly for each node.
– Do not prune the trees.
  Estimate tree performance using out-of-bootstrap data.
  Average the outputs of all decision trees.

Multiband speech recognition

– Filter speech to eliminate a random subset of the frequencies.
– Train speech recognizer on filtered data.
– Repeat and combine with a second tier classifier.
– Resulting recognizer is more robust to noise.

Manipulating the output codes

Reducing multiclass problems to binary classification

– We have seen one versus all.

– We have seen all versus all.

Error correcting codes for multiclass problems

– Code the class numbers with an error correcting code.

– Construct a binary classifier for each bit of the code.

– Run the error correction algorithm on the binary classifier outputs.

IV. Boosting

Motivation

- Easy to come up with rough rules of thumb for classifying data
  - email contains more than 50% capital letters.
  - email contains expression "buy now".
- Each alone isnt great, but better than random.
- Boosting converts rough rules of thumb into an accurate classier.

  Boosting was invented by Prof. Schapire.

Adaboost

Given examples $(x_1, y_1) \dots (x_n, y_n)$ with $y_i = \pm 1$.

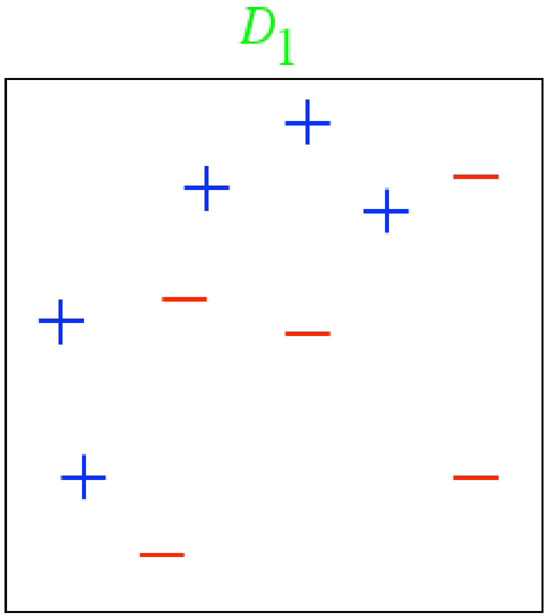
Let $D_1(i) = 1/n$ for $i = 1 \dots n$.

For $t = 1 \dots T$ do

- Run weak learner using examples with weights D_t.
- Get weak classifier h_t
- Compute error: $\varepsilon_t = \sum_i D_t(i)\, \mathbb{I}(h_t(x_i) \neq y_i)$
- Compute magic coefficient $\alpha_t = \dfrac{1}{2} \log\left(\dfrac{1-\varepsilon_t}{\varepsilon_t}\right)$
- Update weights $D_{t+1}(i) = \dfrac{D_t(i)\, e^{-\alpha_t\, y_i\, h_t(x_i)}}{Z_t}$

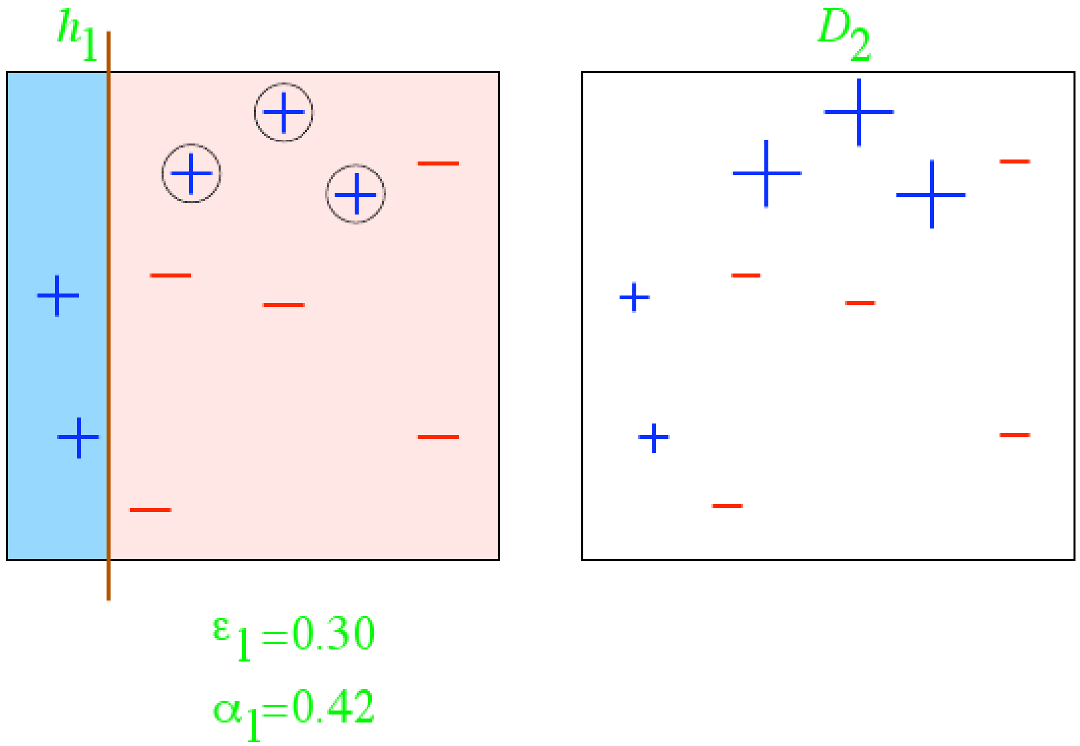
Output the final classifier $f_T(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Toy example

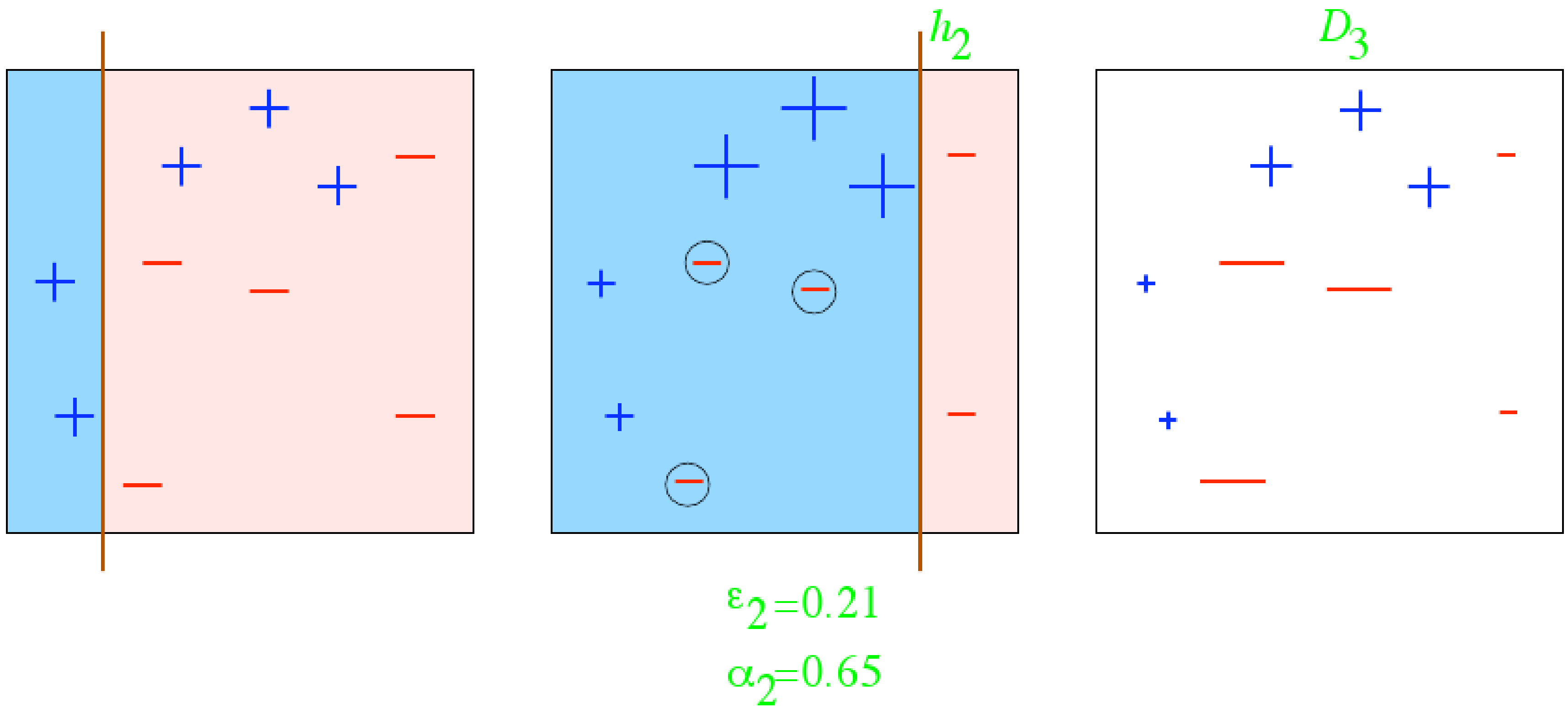


Weak classifiers: vertical or horizontal half-planes.

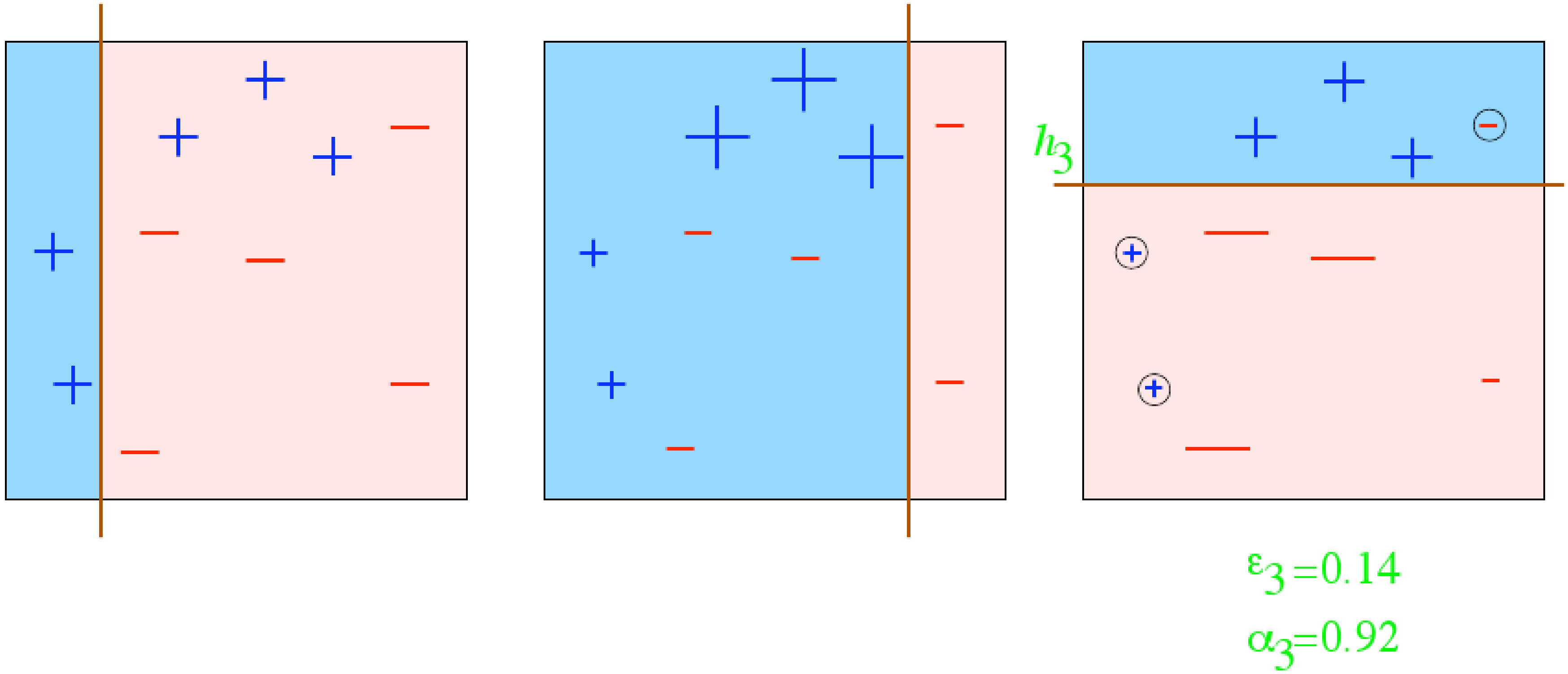
Adaboost round 1



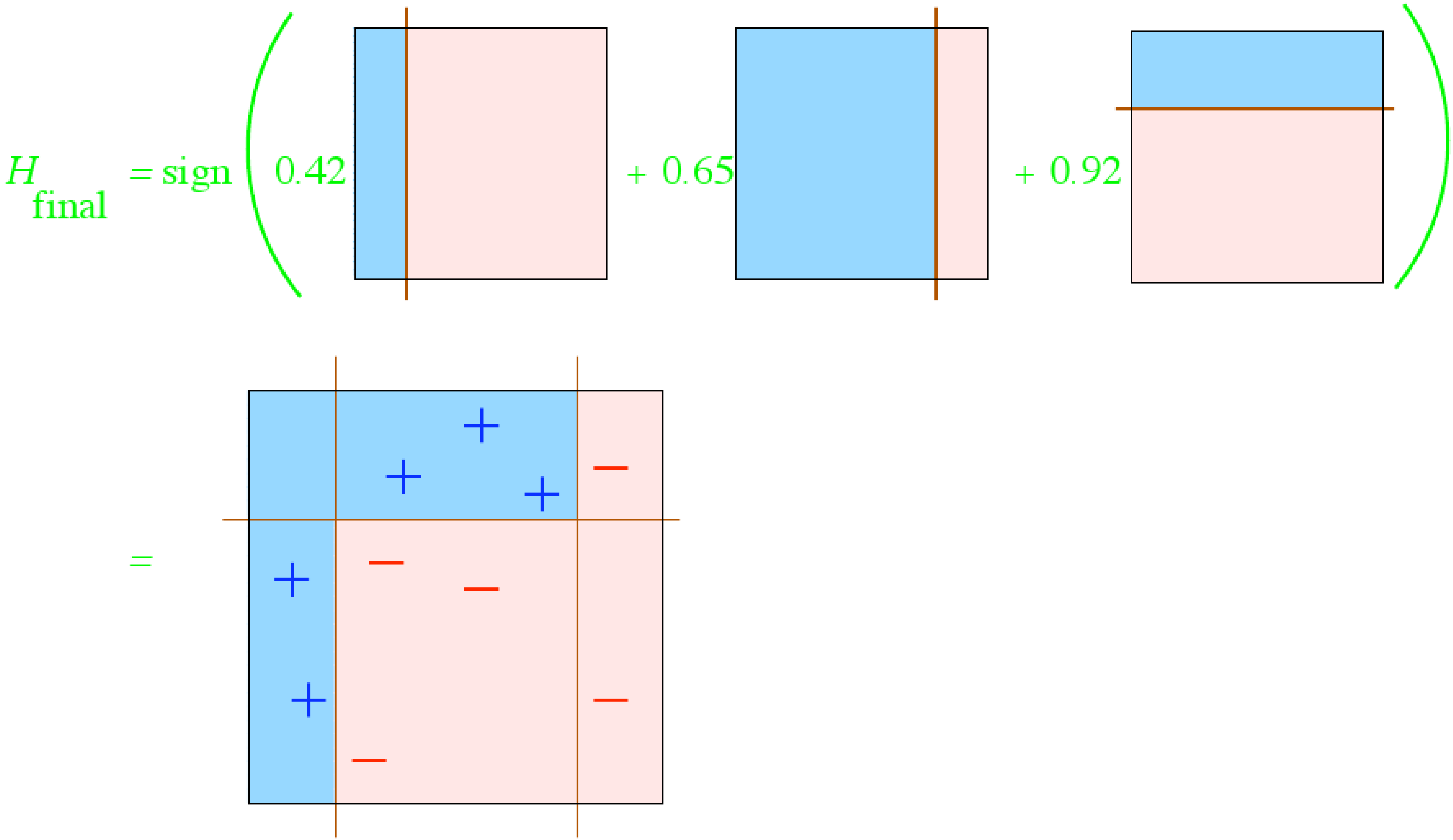
Adaboost round 2



Adaboost round 3



Adaboost final classifier



From weak learner to strong classifier (1)

Preliminary

$$D_{T+1}(i) \;=\; D_1(i)\frac{e^{-\alpha_1 y_i h_1(x_i)}}{Z_1}\cdots\frac{e^{-\alpha_T y_i h_T(x_i)}}{Z_T} \;=\; \frac{1}{n}\,\frac{e^{-y_i f_T(x_i)}}{\prod_t Z_t}$$

Bounding the training error

$$\frac{1}{n}\sum_i \mathbb{I}\{f_T(x_i) \neq y_i\} \;\leq\; \frac{1}{n}\sum_i e{-y_i f_T(x_i)} \;=\; \frac{1}{n}\sum_i D_{T+1}(i)\prod_t Z_t \;=\; \prod_t Z_t$$

Idea: make Z_t as small as possible.

$$Z_t \;=\; \sum_{i=1}^{n} D_t(i)e^{-\alpha_t y_i h_t(x_i)} \;=\; n\,(1-\varepsilon_t)\,e^{-\alpha_t} + n\,\varepsilon_t\,e^{\alpha_t}$$

1. Pick h_t to minimize ε_t.
2. Pick α_t to minimize Z_t.

From weak learner to strong classifier (2)

Pick α_t to minimize Z_t (the magic coefficient)

$$\frac{\partial Z_t}{\partial \alpha_t} = -(1-\varepsilon_t)\, e^{-\alpha_t} + \varepsilon_t\, e^{\alpha_t} = 0 \quad \Longrightarrow \quad \alpha_t = \frac{1}{2} \log \frac{1-\varepsilon_t}{\varepsilon_t}$$

Weak learner assumption: $\gamma_t = \frac{1}{2} - \varepsilon_t$ is positive and small.

$$Z_t \;=\; (1-\varepsilon)\sqrt{\frac{\varepsilon}{1-\varepsilon}} + \varepsilon\sqrt{\frac{1-\varepsilon}{\varepsilon}} \;=\; \sqrt{4\varepsilon(1-\varepsilon)} = \sqrt{1-4\gamma_t^2} \;\leq\; \exp\left(-2\gamma_t^2\right)$$

$$\text{TrainingError}(f_T) \;\leq\; \prod_{t=1}^{T} Z_t \;\leq\; \exp\left(-2\sum_{t=1}^{T}\gamma_t^2\right)$$

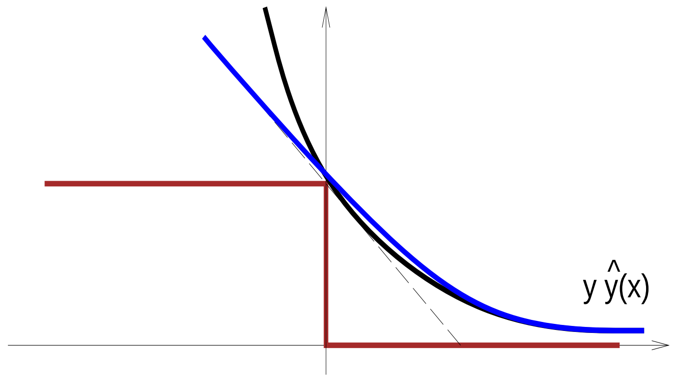
The training error decreases exponentially if $\inf \gamma_t > 0$.

But that does not happen beyond a certain point...

Boosting and exponential loss

Proofs are instructive

We obtain the bound



$$\text{TrainingError}(f_T) \;\leq\; \frac{1}{n}\sum_i e^{-y_i H(x_i)} \;=\; \prod_{t=1}^{T} Z_t$$

- without saying how D_t relates to h_t
- without using the value of α_t

Conclusion

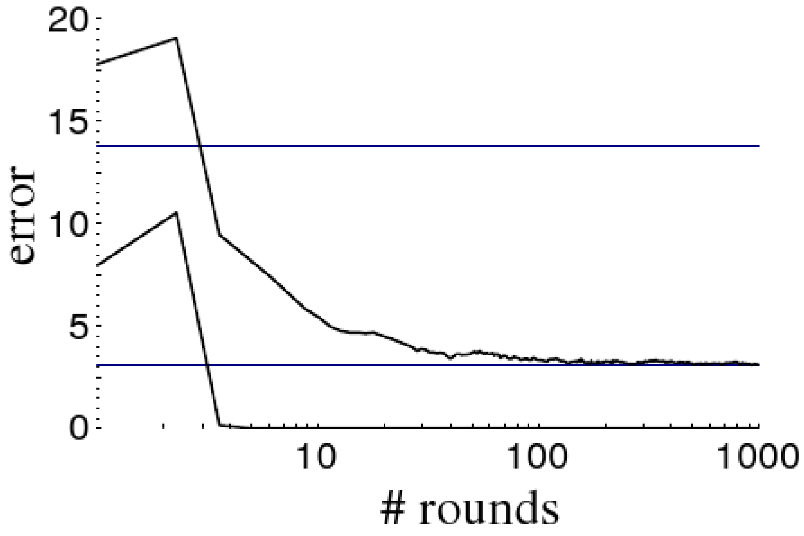
– Round T chooses the h_T and α_T
that maximize the exponential loss reduction from f_{T-1} to f_T.

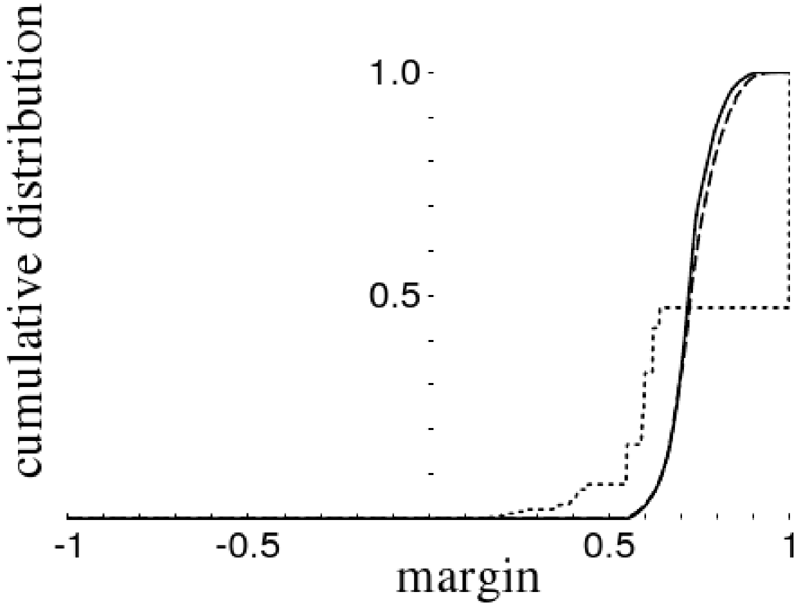
Exercise

– Tweak Adaboost to minimize the log loss instead of the exp loss.

Boosting and margins

$$margin_H(x,y) \;=\; \frac{y\,H(x)}{\sum_t |\alpha_t|} \;=\; \frac{\sum_t \alpha_t\, y\, h_t(x)}{\sum_t |\alpha_t|}$$





Remember support vector machines?