Boosting

Recitation 9

Oznur Tastan

Outline

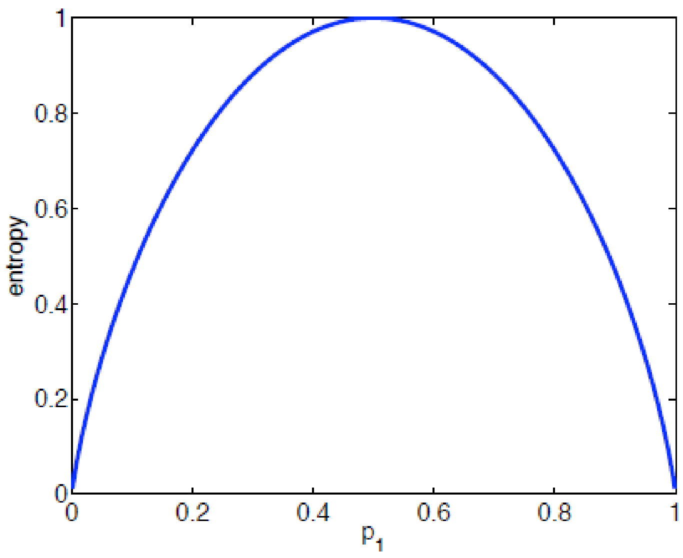
- Overview of common mistakes in midterm
- Boosting

Sanity checks

- Entropy for discrete variables is always non-negative and equals zero only if the variable takes on a single value

$$H(X) = E(I(X)) = \sum_i p(x_i) I(x_i) = -\sum_i p(x_i) \log_2 p(x_i)$$

- Information gain is always non-negative



Sanity checks

In decision trees:

- You cannot obtain a leaf that has no training examples
- If a leaf contains examples from multiple classes, you predict the most common class.
- If there are multiple, you predict any of the most common classes.

Common mistakes

For each of the listed descriptions below, circle whether the experimental set up is *ok* or *problematic*. If you think it is problematic, briefly state **all** the problems with their approach:

4. **[Points: 4 pts]** A project team performed a feature selection procedure on the full data and reduced their large feature set to a smaller set. Then they split the data into test and training portions. They built their model on training data using several different model settings, and report the the best test error they achieved.

   (a) Ok

   (b) Problematic ★

Many people only stated one of either of the problems.

Common mistakes

6.3 Controlling overfitting

Increase the number of training examples in logistic regression, the bias remains unchanged. MLE is an approximately unbiased estimator.

11 Bayesian networks’

$$H \rightarrow U \leftarrow P \leftarrow W$$

[Points: 4 pts] *True* or *false*: Given the above network structure, it is possible that $H \perp U \mid P$. Explain briefly.

Many people forgot about the possibility of accidental independences.

12 Graphical model inference

Entries in potential tables aren't probabilities

Boosting

- As opposed to bagging and random forest learn many **big trees**
- Learn many **small trees (weak classifiers)**

Commonly used terms:

Learner = Hypothesis = Classifier

Boosting

- Given weak learner that can consistently classify the examples with error $\leq 1/2-\gamma$
- A boosting algorithm can provably construct single classifier with error $\leq \varepsilon$

where ε and γ are small.

AdaBoost

In the first round all examples are equally weighted

$D_t(i)=1/N$

At each run:

Concentrate on the hardest ones:

The examples that are misclassified in the previous run are weighted more so that the new learner focuses on them.

At the end:

Take a weighted majority vote.

Formal description

- given training set $(x_1, y_1), \ldots, (x_m, y_m)$
- $y_i \in \{-1, +1\}$ correct label of instance $x_i \in X$
- for $t = 1, \ldots, T$:
  - construct distribution D_t on $\{1, \ldots, m\}$
  - find weak classifier ("rule of thumb")

$$h_t : X \to \{-1, +1\}$$

  with small error ϵ_t on D_t:

$$\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$$

- output final classifier H_{final}

this is a distribution over examples

this is the classifier or hypothesis

weighted error mistake on an example with high weight costs much.

weighted majority vote

Updating the distribution

- constructing D_t:
  - $D_1(i) = 1/m$
  - given D_t and h_t:

Correctly predicted this example
decrease the weight of the example

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$

Mistaken.
Increase the weight of the example

Updating D_t

- constructing D_t:
  - $D_1(i) = 1/m$
  - given D_t and h_t:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$
$$= \frac{D_t(i)}{Z_t} \exp(-\alpha_t \, y_i \, h_t(x_i))$$

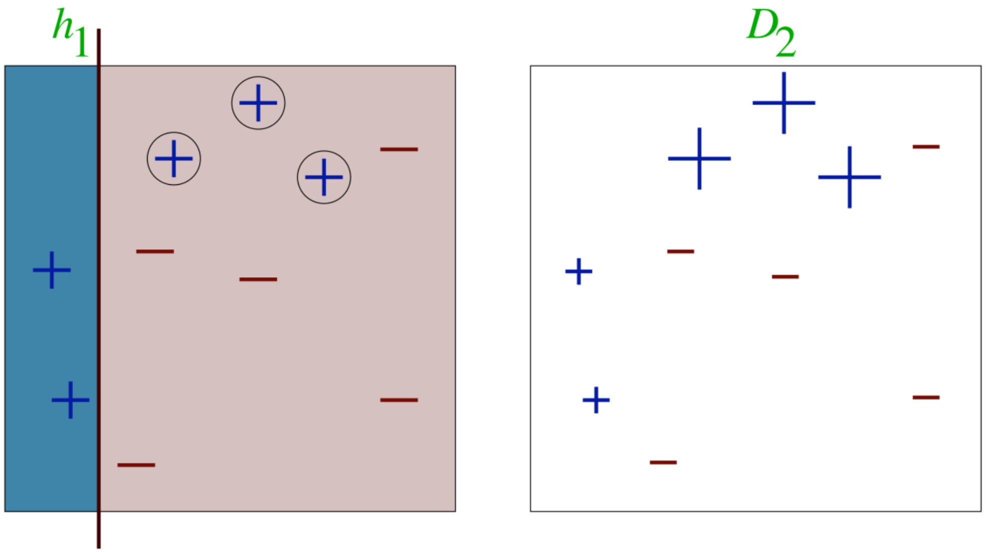
$$y_i \in \{-1, +1\}$$
$$h_t \in \{-1, +1\}$$

where Z_t = normalization constant

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) > 0$$

weighted error of the classifier

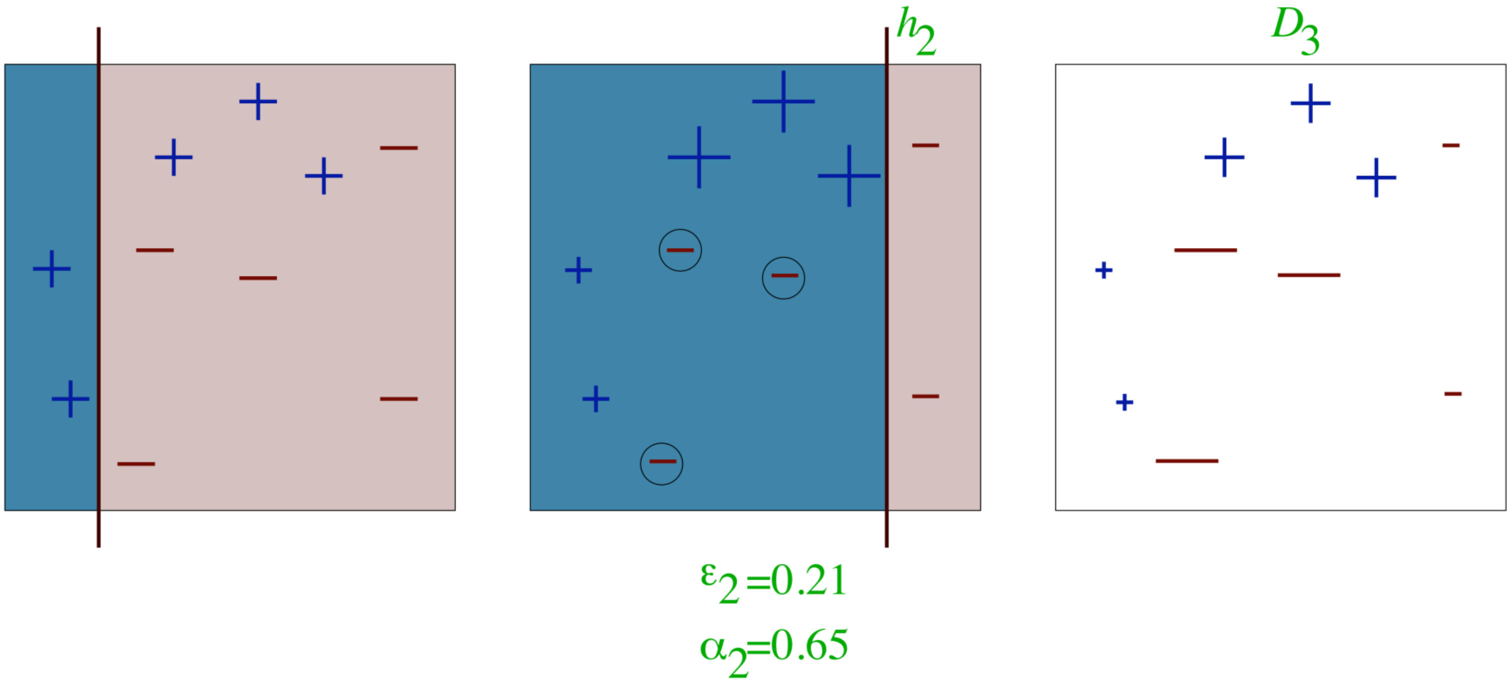
Round 1



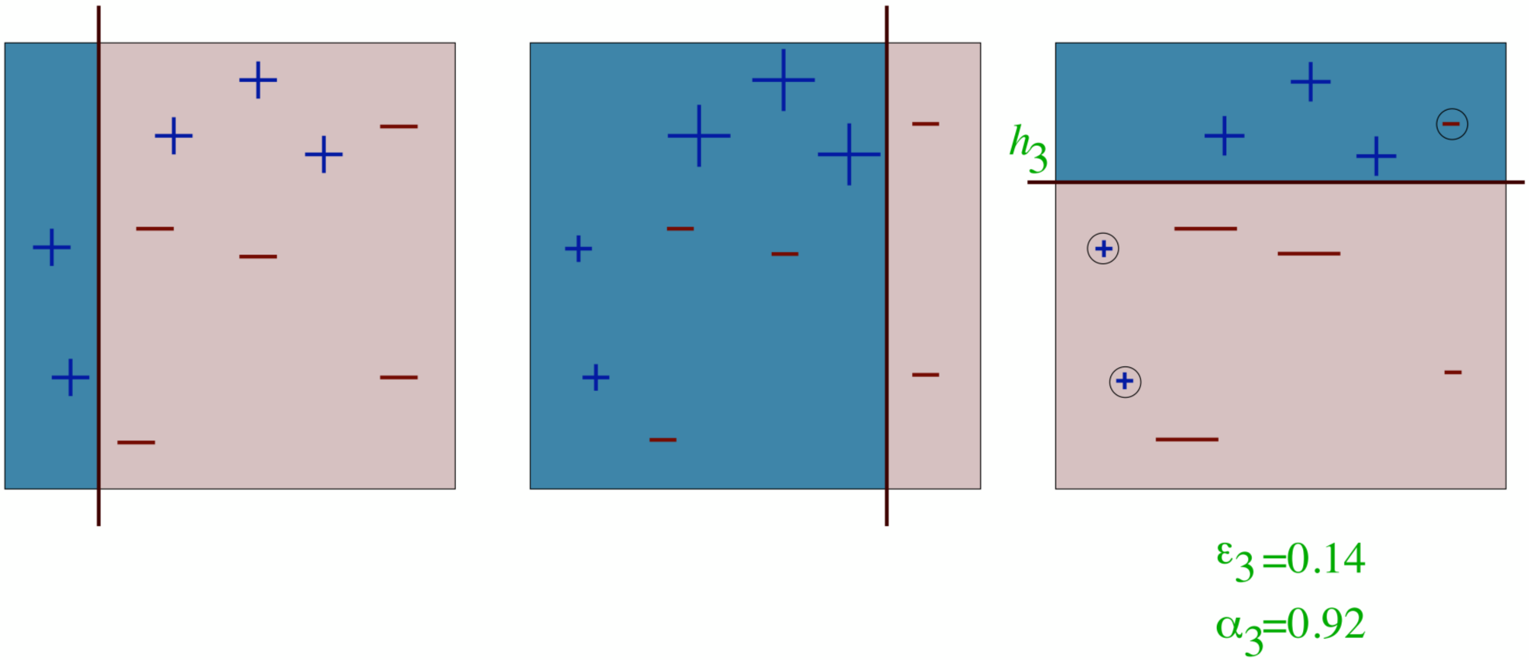
$\varepsilon_1 = 0.30$

$\alpha_1 = 0.42$ → 1/2 ln((1-0.3)/0.3)

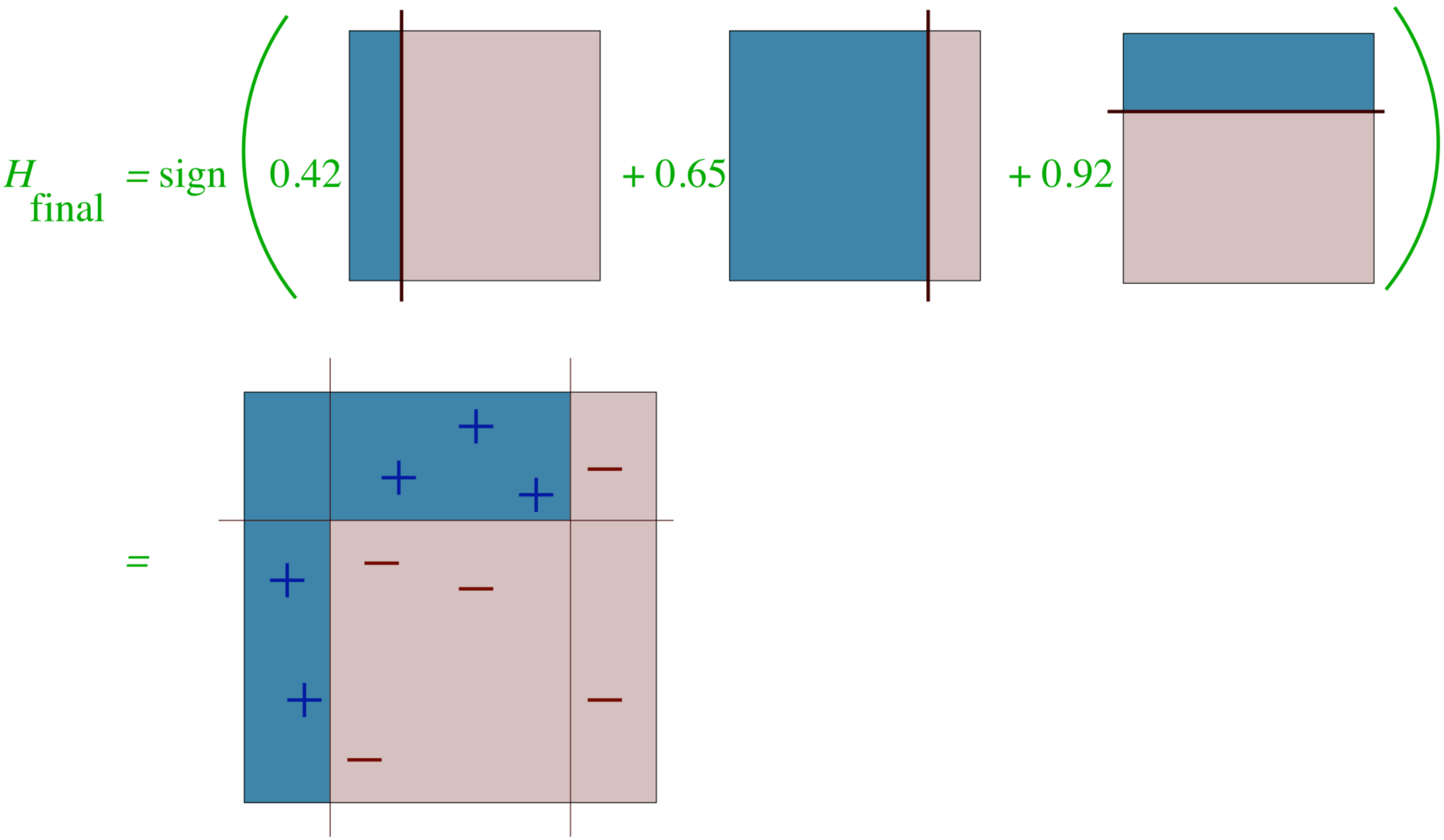
Round 2



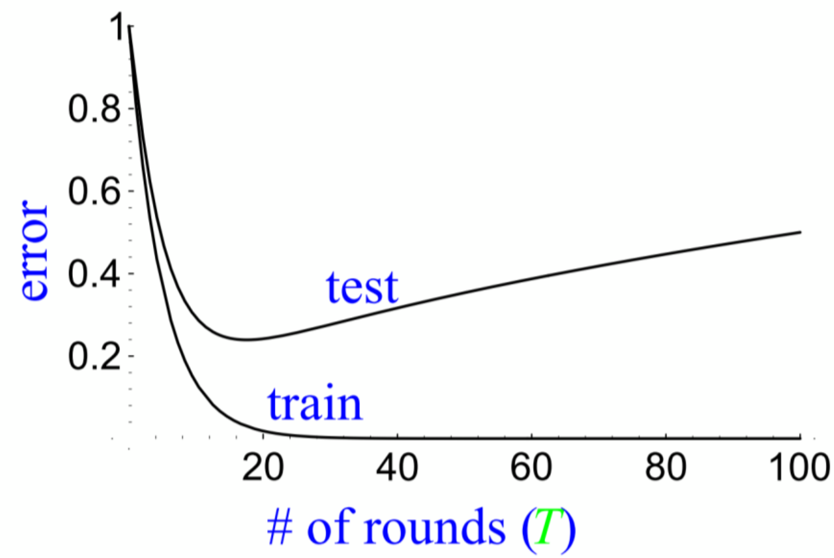
Round 3



Final classifier



When final hypothesis is too complex



expect:

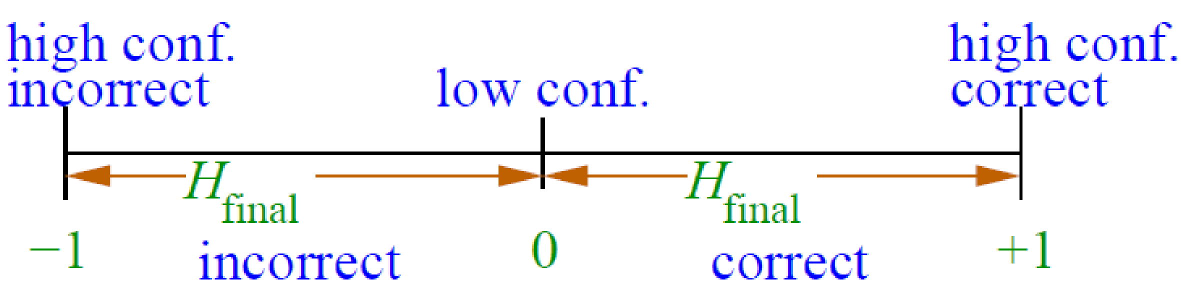
- training error to continue to drop (or reach zero)
- test error to increase when H_{final} becomes "too complex"

Margin of the classifier

- key idea:
  - training error only measures whether classifications are right or wrong
  - should also consider confidence of classifications
- can write: $H_{\text{final}}(x) = \text{sign}(f(x))$

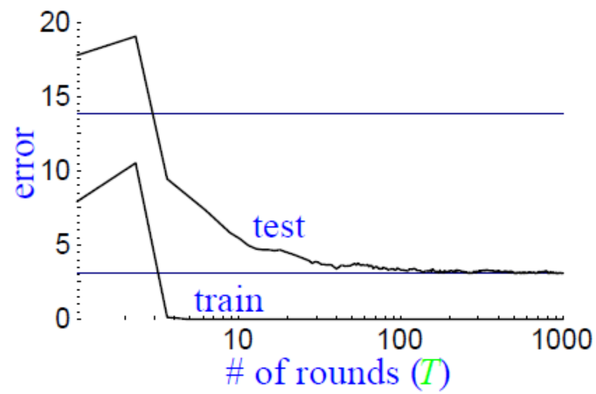
  where $f(x) = \dfrac{\sum_t \alpha_t h_t(x)}{\sum_t \alpha_t} \in [-1, +1]$
- define margin of example (x, y) to be $y\ f(x)$

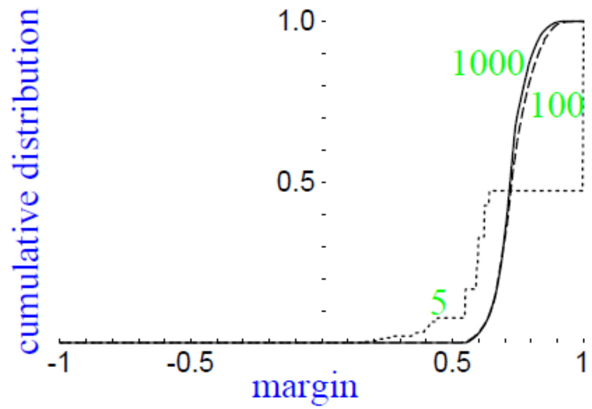
  $=$ measure of confidence of classifications



Cumulative distribution of the margins

- <u>margin distribution</u>
  = cumulative distribution of margins of training examples





	# rounds		
	5	100	1000
train error	0.0	0.0	0.0
test error	8.4	3.3	3.1
% margins ≤ 0.5	7.7	0.0	0.0
minimum margin	0.14	0.52	0.55

Although the final classifier is getting larger, the margins are increasing.

Advantages

- Fast
- Simple and easy to program
- No parameters to tune (except T)
- Provably effective

- Performance depends on the data and the weak learner
- Can fail if the weak learners are too complex (overfitting)
- If the weak classifiers are too simple (underfitting)

References

Miroslav Dudik lecture slides.