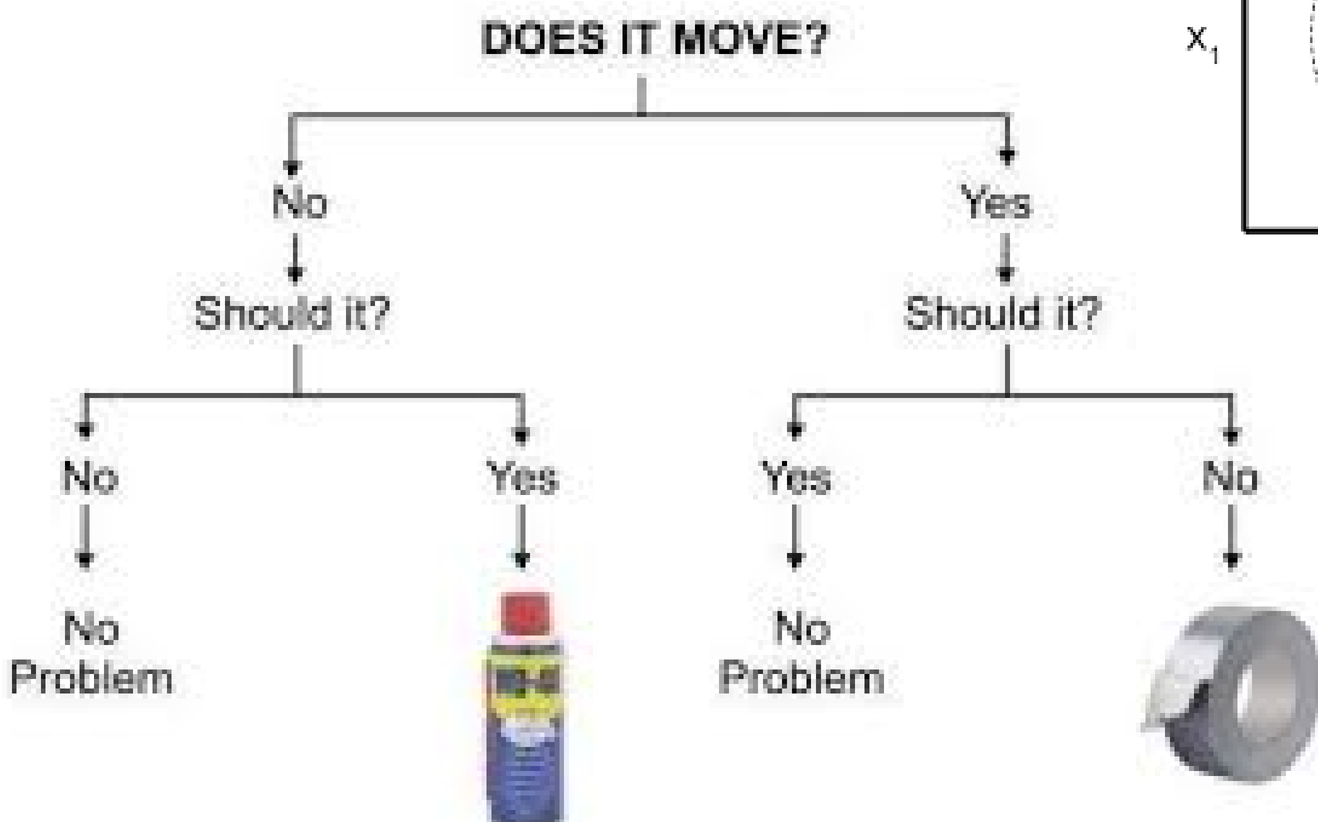
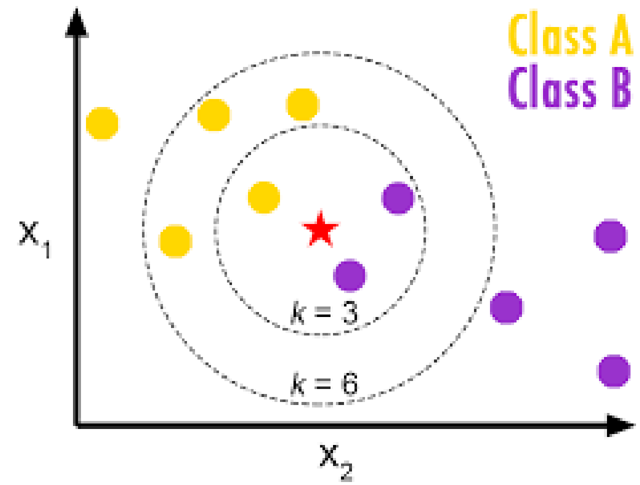
Machine Learning

4771

Instructor: Itsik Pe’er

Reminder: Decision Trees, k-Nearest Neighbors





Today: Ensembles

How to turn weak classifiers into a strong one?

Framework

- Probability distribution P over $\boldsymbol{X} \times \{\pm 1\}$
- Samples $S = \{(x_i, y_i)\}_{i=1}^{N}$ from P
- Goal:

  Learn classifier $f\colon \boldsymbol{X} \rightarrow \{\pm 1\}$ s.t.

$$err(f) = P(f(X) \neq Y) = 0$$

Framework

- Probability distribution P over $\boldsymbol{X} \times \{\pm 1\}$
- Samples $S = \{(x_i, y_i)\}_{i=1}^{N}$ from P
- Goal: w/prob $\geq 1 - \delta$ (over choice of S)

Learn classifier $f: \boldsymbol{X} \rightarrow \{\pm 1\}$ s.t.

$$err(f) = P(f(X) \neq Y) \leq \epsilon$$

Framework

- Probability distribution P over $\boldsymbol{X} \times \{\pm 1\}$
- Samples $S = \{(x_i, y_i)\}_{i=1}^{N}$ from P
- Goal: w/prob $\geq 1 - \delta$ (over choice of S)
  Learn classifier $f: \boldsymbol{X} \rightarrow \{\pm 1\}$ s.t.
  $$err(f) = P(f(X) \neq Y) \leq \epsilon$$
- Easier goal: learn non-trivial classifier (better than chance) $err(f) \leq \frac{1}{2} - \gamma$

Boosting

- For $t = 1,2,..T$:
  - Train "weak learner" f_t on samples S_t: $f_t \leftarrow WL(S_t)$
- Return ensemble_classifier$(f_1, f_2, \ldots, f_T)$

Boosting

- For $t = 1,2,..T$:
  - $S_t \leftarrow$ random subset of samples $S_t \subseteq S$
  - Train "weak learner" f_t on samples S_t:$f_t \leftarrow WL(S_t)$
- Return ensemble_classifier$(f_1, f_2, \dots, f_T)$

Bootstrap Aggregating

- For $t = 1,2,..T$:
  - $S_t \leftarrow$ rand. multiset from S w/ replacement $|S_t| = |S|$
  - Train "weak learner" f_t on samples S_t: $f_t \leftarrow WL(S_t)$

- Return ensemble_classifier$(f_1, f_2, \ldots, f_T)$

History

- 1984 Is boosting possible? [Valiant,Kearns]
- 1989 1st boosting algorithm [Schapire]

History

- 1984 Is boosting possible? [Valiant,Kearns]
- 1989 1st boosting algorithm [Schapire]
- 1990 Boost-by-majority: optimal! [Freund]
- 1995 AdaBoost [Freund & Schapire]

Bootstrap Aggregating

- For $t = 1,2,..T$:
  - $S_t \leftarrow$ rand. multiset from S w/ replacement $|S_t| = |S|$
  - Train "weak learner" f_t on samples S_t: $f_t \leftarrow WL(S_t)$
- Return ensemble_classifier$(f_1, f_2, \dots, f_T)$

Bootstrap Aggregating

$D_t(i) \leftarrow$ #copies of (x_i, y_i) in S_t

- For $t = 1,2,..T$:
  - $S_t \leftarrow$ rand. multiset from S w/ replacement $|S_t| = |S|$
  - Train "weak learner" f_t on samples S_t: $f_t \leftarrow WL(S_t)$
- Return ensemble_classifier$(f_1, f_2, \dots, f_T)$

Bootstrap Aggregating

- For $t = 1,2,..T$:
  - $\forall i$: $D_t(i) \leftarrow$ random integer representing resampling
  - Train "weak learner" f_t on D_t-weighted samples

$$f_t \leftarrow WL(D_t, S)$$

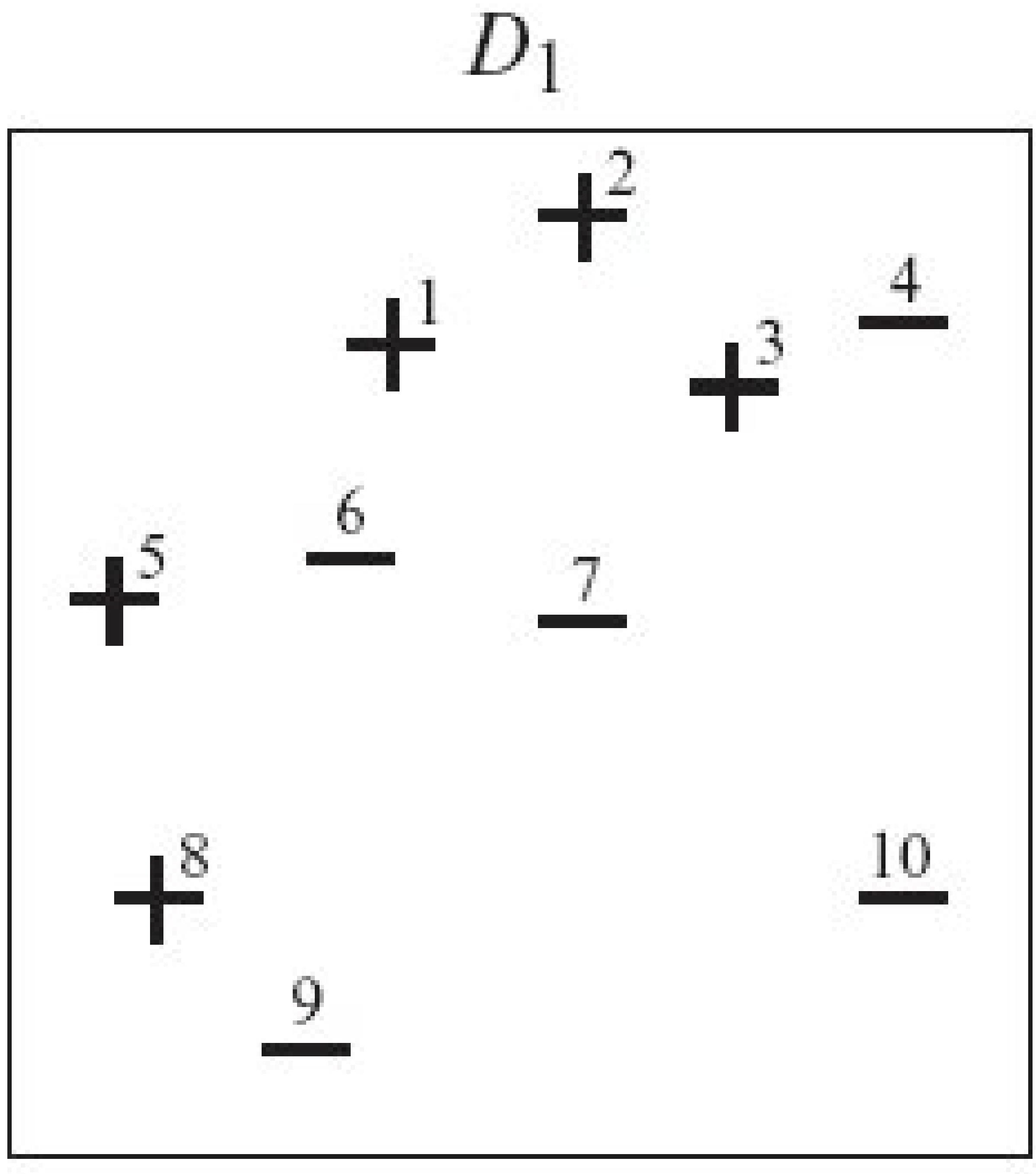
- Return ensemble_classifier$(f_1, f_2, \ldots, f_T)$

Adaptive Boosting

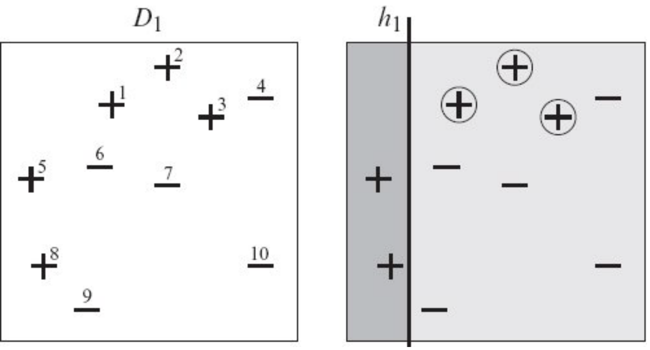
- $\forall i\colon\ D_t(i) \leftarrow \frac{1}{N}$
- For $t = 1,2,..T$:
  - Train f_t on D_t-weighted samples $f_t \leftarrow WL(D_t, S)$
  - Reevaluate weights: compute D_{t+1}
- Return ensemble$(f_1, f_2, \dots, f_T)$

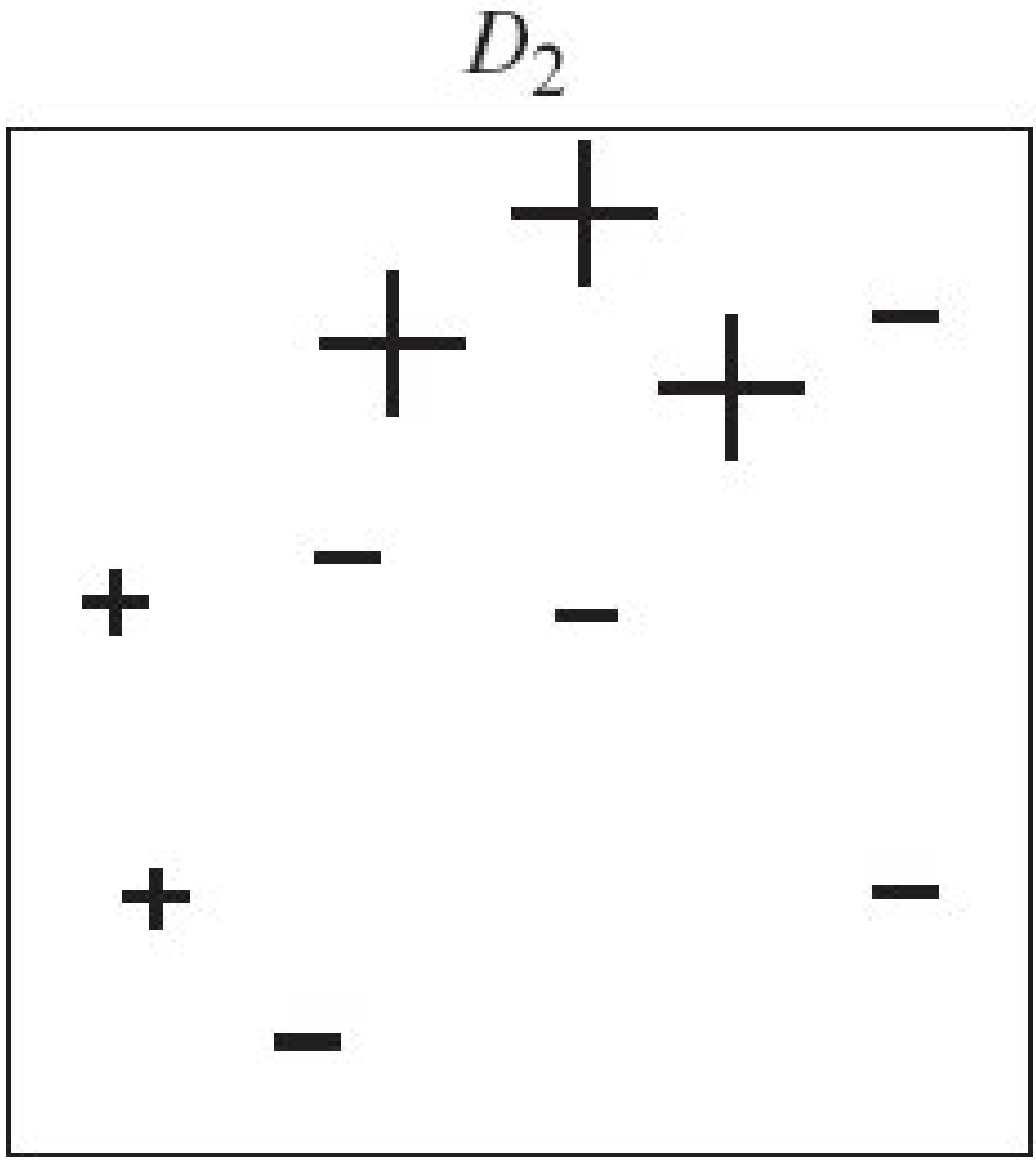
Example

Weak learner: Axis-parallel classifiers (decision stump)

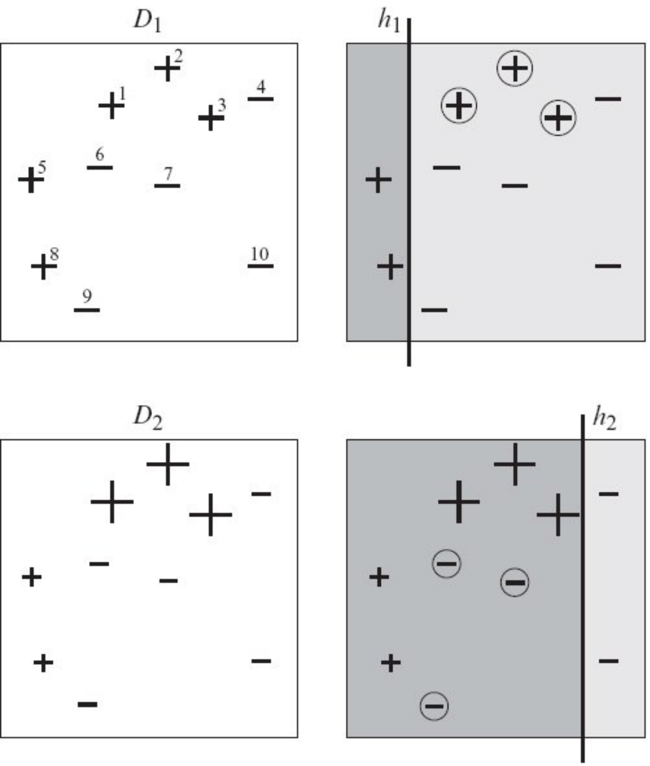


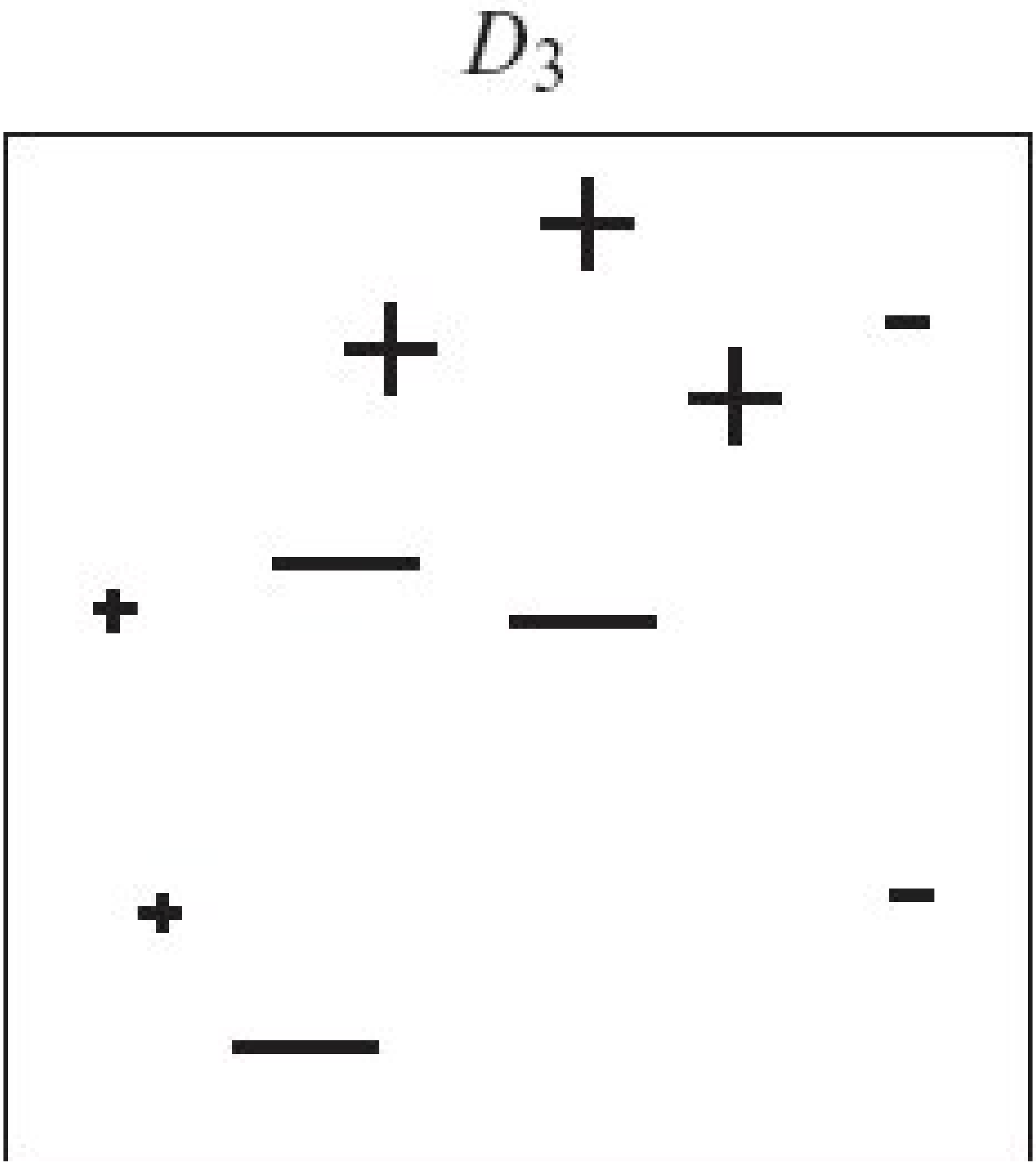
Example





Example





Quality of a weighted classifier

- Suppose $(X,Y)\sim D_t;\ P(f(X)=Y)=\frac{1}{2}+\gamma_t$

Quality of a weighted classifier

- Suppose $(X,Y) \sim D_t$; $P(f(X)=Y) = \frac{1}{2} + \gamma_t$
- $z_t = \sum_{i=1}^{N} D_t(i) y_i f_t(x_i) = 2\gamma_t$
  - $z_t = 0$: random guessing w.r.t. D_t
  - $z_t > 0$: f_t better than random
  - $z_t < 0$: $-f_t$ better than random

AdaBoosting

- $\forall i\colon D_t(i) \leftarrow \frac{1}{N}$
- For $t = 1,2,..T$:
  - Train f_t on D_t-weighted samples $f_t \leftarrow WL(D_t, S)$
  - Reevaluate weights: compute D_{t+1}
    - Quality of f_t : $z_t \leftarrow \sum_{i=1}^{N} D_t(i) y_i f_t(x_i) \in (-1, +1)$
- Return ensemble$(f_1, f_2, \dots, f_T)$

AdaBoosting

- $\forall i$: $D_t(i) \leftarrow \frac{1}{N}$
- For $t = 1,2,..T$:
  - Train f_t on D_t-weighted samples $f_t \leftarrow WL(D_t, S)$
  - Reevaluate weights: compute D_{t+1}
    - Quality of f_t : $z_t \leftarrow \sum_{i=1}^{N} D_t(i) y_i f_t(x_i) \in (-1, +1)$
    - Up-weight wrong samples: $D_t(i) *= \sqrt{\frac{1+z_t}{1-z_t}}$
    - Down-weight right samples: $D_t(i) /= \sqrt{\frac{1+z_t}{1-z_t}}$
- Return ensemble$(f_1, f_2, \ldots, f_T)$

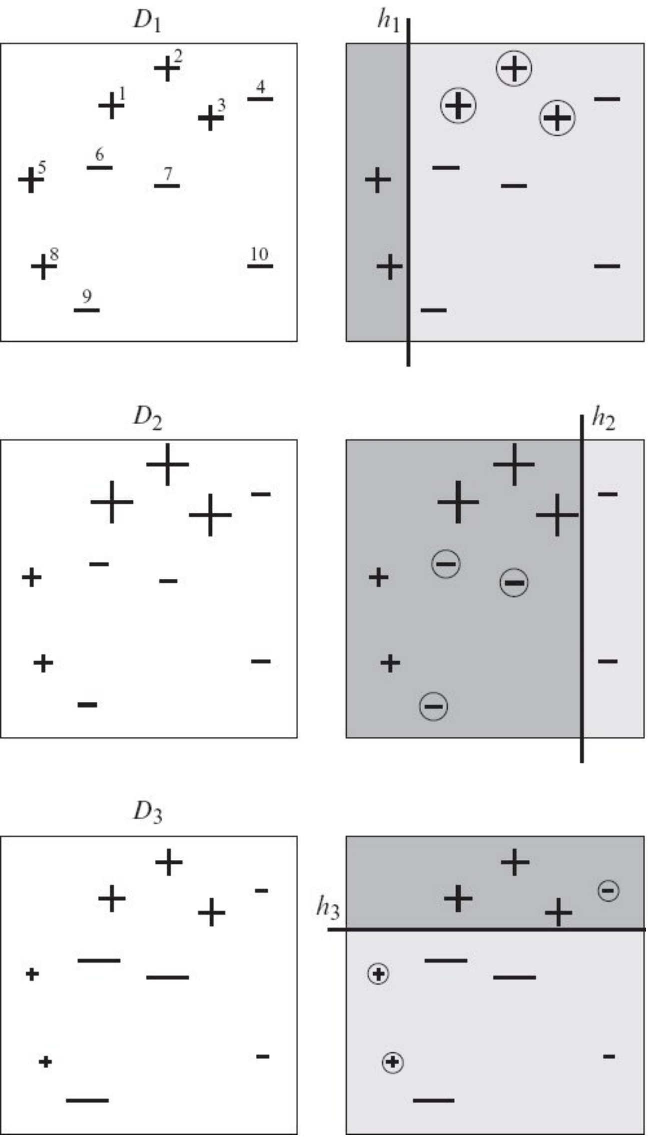
AdaBoosting

- $\forall i$: $D_t(i) \leftarrow \frac{1}{N}$
- For $t = 1,2,..T$:
  - Train f_t on D_t-weighted samples $f_t \leftarrow WL(D_t, S)$
  - Reevaluate weights: compute D_{t+1}
    - Quality of f_t : $z_t \leftarrow \sum_{i=1}^{N} D_t(i) y_i f_t(x_i) \in (-1, +1)$
    - Weight of f_t : $\alpha_t \leftarrow \frac{1}{2} \ln \frac{1+z_t}{1-z_t} \in \boldsymbol{R}$
    - Up-/down-weight samples $D_{t+1}(i) \leftarrow D_t(i) \exp(-\alpha_t y_i f_t(x_i))$
- $g(x) \stackrel{\text{def}}{=} \frac{\sum_{t=1}^{T} \alpha_t f_t(x)}{\sum_{t=1}^{T} |\alpha_t|}$; return $\hat{f}(x) = sign\big(g(x)\big)$

AdaBoost

- $\forall i$: $D_t(i) \leftarrow \frac{1}{N}$
- For $t = 1,2,..T$:
  - Train f_t on D_t-weighted samples $f_t \leftarrow WL(D_t, S)$
  - Reevaluate weights: compute D_{t+1}
    - Quality of f_t : $z_t \leftarrow \sum_{i=1}^{N} D_t(i) y_i f_t(x_i) \in (-1, +1)$
    - Weight of f_t : $\alpha_t \leftarrow \frac{1}{2} \ln \frac{1+z_t}{1-z_t} \in \boldsymbol{R}$
    - Up-/down-weight samples $D_{t+1}(i) \leftarrow D_t(i) \exp(-\alpha_t y_i f_t(x_i))$
    - Normalize:$Z_t = \sum_{i=1}^{N} D_{t+1}(i)$; $D_{t+1}(i) \leftarrow \frac{D_{t+1}(i)}{Z_t}$
- $g(x) \stackrel{\text{def}}{=} \frac{\sum_{t=1}^{T} \alpha_t f_t(x)}{\sum_{t=1}^{T} |\alpha_t|}$; return$\hat{f}(x) = sign\big(g(x)\big)$

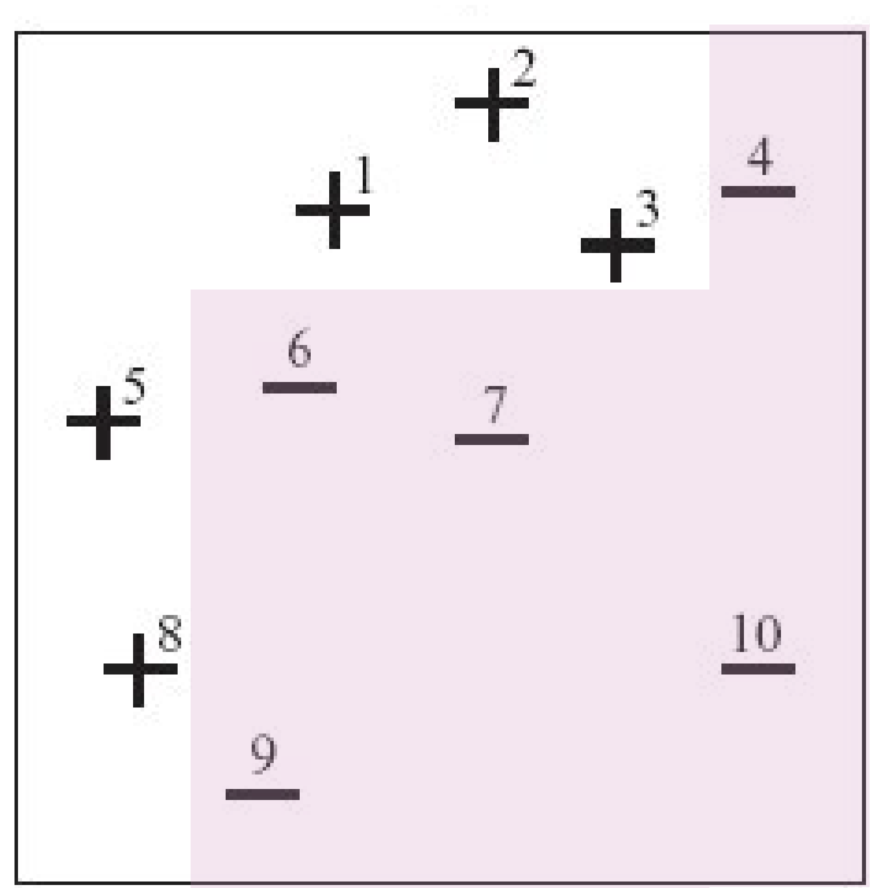
Example



$z_1 = .40$
$\alpha_1 = .42$

$z_2 = .58$
$\alpha_2 = .65$

$z_3 = .72$
$\alpha_3 = .92$



$$\hat{f}(x) = sign(.42 f_1(x) + .65 f_2(x) + .92 f_3(x))$$

Performance of Classifier

- Complexity: $T \times$ Complexity(WL)
- Accuracy (decision stumps in $\boldsymbol{R}^d$)
  - Partially prove at home:

$$err(\hat{f}) \leq \exp(-2\bar{\gamma}^2 T) + O\left(\sqrt{\frac{T \log d}{N}}\right)$$

Boosting Margins

- $margin\big((x,y)\big) = yg(x) \in [-1,+1]$
- Thm (SFBL '98):
  - $\uparrow$margins $\Rightarrow$ $\downarrow$overfitting
  - AdaBoost tends to increase training margins
- Even if train error=0, more iterations help!

AdaBoost as Linear Classifier

- Weak classifiers as features:
  Family of classifiers $\boldsymbol{F} = \{f\}$ implies feature vector

$$\phi\colon \boldsymbol{X} \to \{-1, +1\}^{\boldsymbol{F}}$$

AdaBoost as Linear Classifier

- Weak classifiers as features:
  Family of classifiers $\boldsymbol{F} = \{f\}$ implies feature vector
  $$\phi: \boldsymbol{X} \rightarrow \{-1,+1\}^{\boldsymbol{F}}$$
- $\hat{f} = sign\left(\sum_{t=1}^{T} \frac{\alpha_t}{\sum_{\tau=1}^{T} |\alpha_\tau|} f_t(x)\right) = sign(\langle w, \phi(x) \rangle)$
- $w_f = \begin{cases} \frac{\alpha_t}{\sum_{\tau=1}^{T} |\alpha_\tau|} & f = f_t \\ 0 & otherwise \end{cases}$

AdaBoost as Coordinate Descent

- At each point choose one dimension to improve

- Use exponential loss

$$\min_{w \in \boldsymbol{R}^F} \frac{1}{N} \sum_{i=1}^{N} exp(-y_i \langle w, \phi(x_i) \rangle)$$

Summary

- Ensembles of weak classifiers are strong
- Resampling/reweighting samples:
- AdaBoost as a flexible framework