AdaBoost

Data Visualizations and Performances

Introduction

- What is ML(classification problem)
- What is boosting/AdaBoost
- Intense theoretical studies / empirical testing
- Main idea of AdaBoost

Assumptions

- Binary classification
- Unknown distribution on given training sets
- All the training examples are i.i.d

Pseudocode

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

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

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$.
- Update:

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

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

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

Figure 1: The boosting algorithm AdaBoost.

Original Distribution & Generation of Training Sets

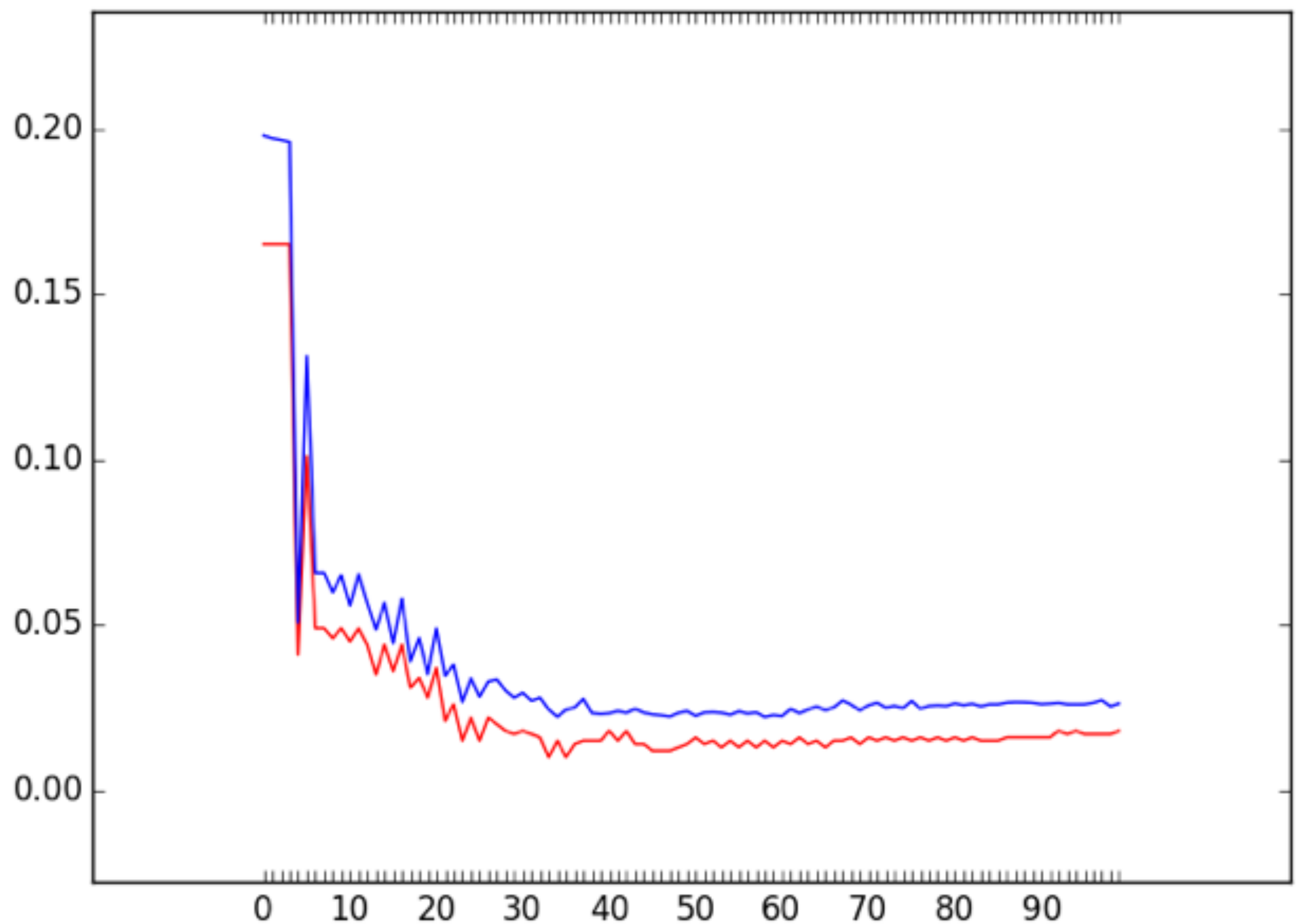
```
#Unknown distribution P on X*Y


def f(x,y):
    if (x)**2+(y-1)**2<0.25 or (x)**2+(y+1)**2<0.25 :
     return 1
    else:
     return -1
```

```
# Generates n training examples using f from function*.
# The points lie in [-1,1]x[-1,1]
import random

from function1 import f

def examples(n):# 1000 points or 10000 points
    r=random.random
    posvals=[]
    for i in range(n):
     xtemp = 2*r()-1
     ytemp = 2*r()-1
     valtemp = f(xtemp,ytemp)
     posvals.append([xtemp,ytemp,valtemp,1/n])
    return posvals
```

T vs Error for 1000 instances

Blue:=Gen/Testing error

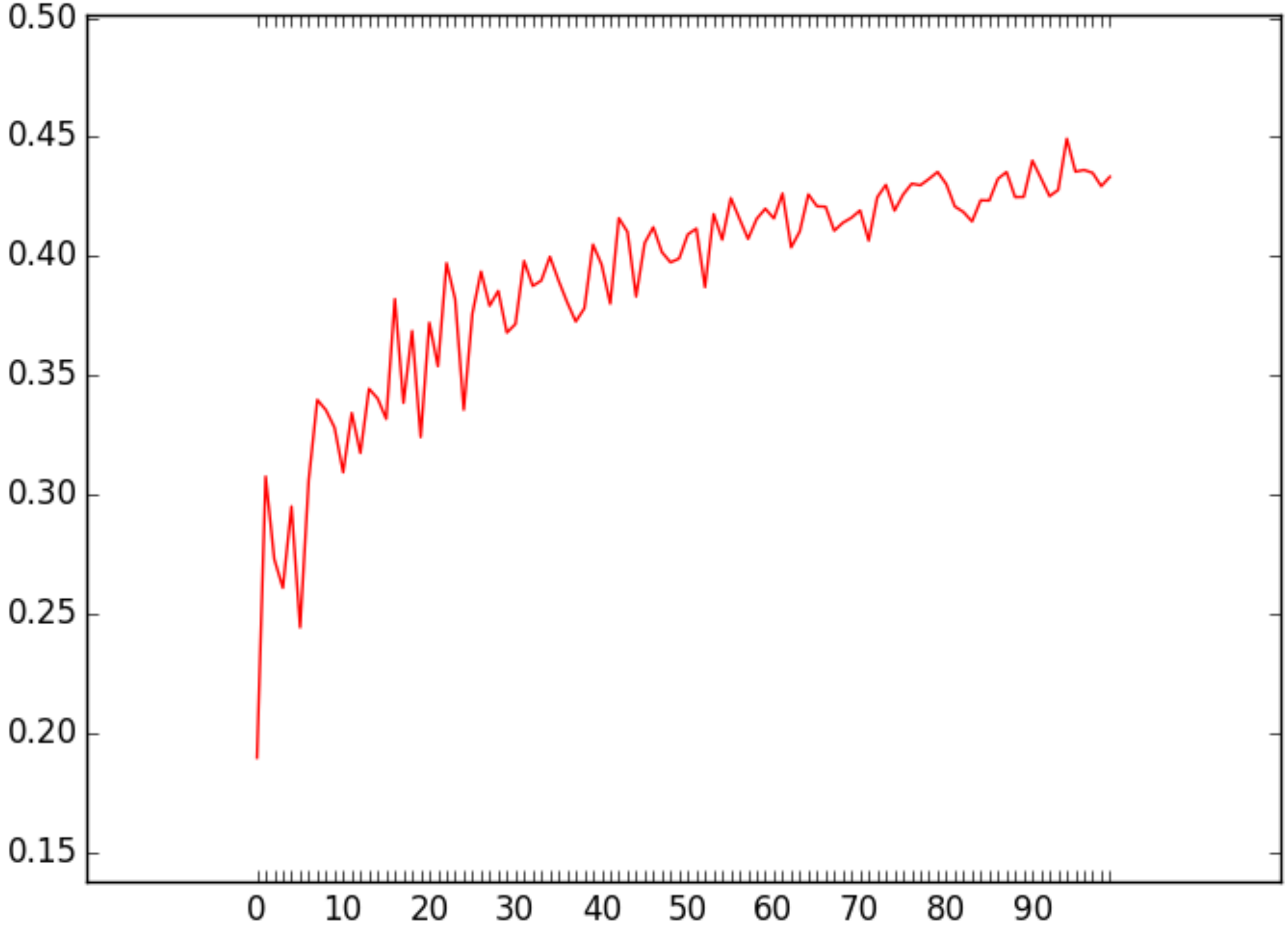
Red:=Training error

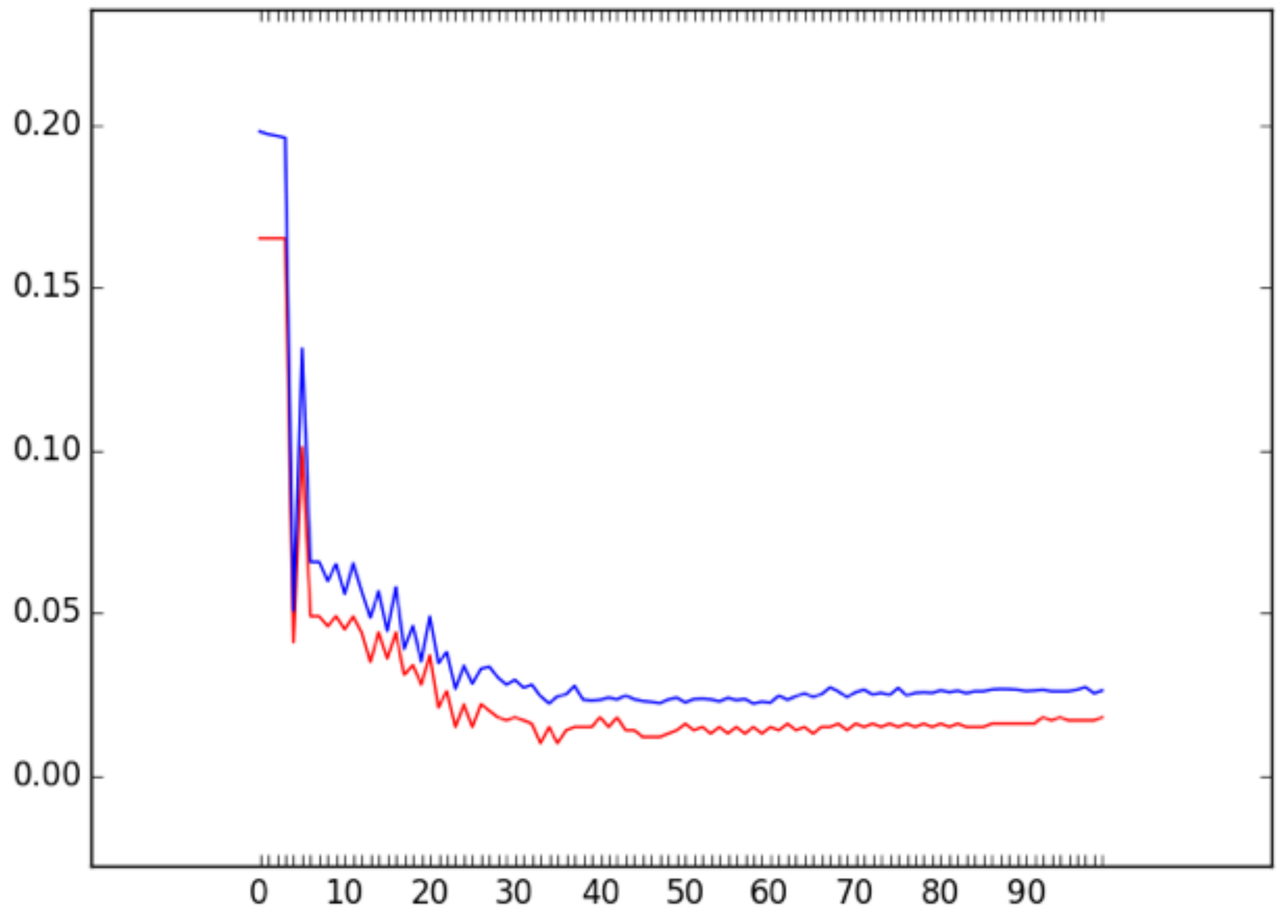


Intuition/ formal argument

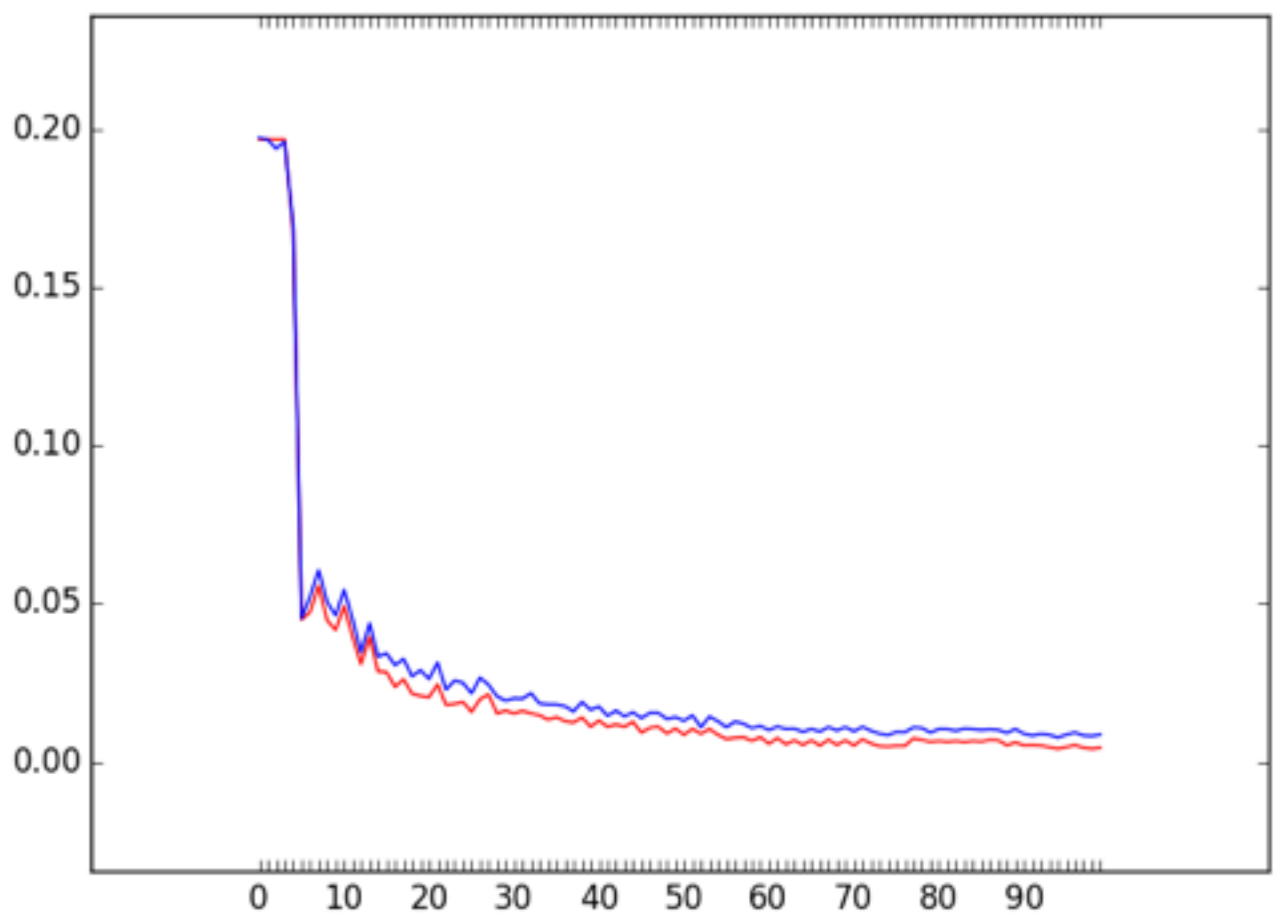
- Every round, Ada increases weights of misclassified examples
- H is a weighted majority vote, if some examples are misclassified by H then it must have been misclassified by most of the weak hypo—>large weights

T vs Epsilon t

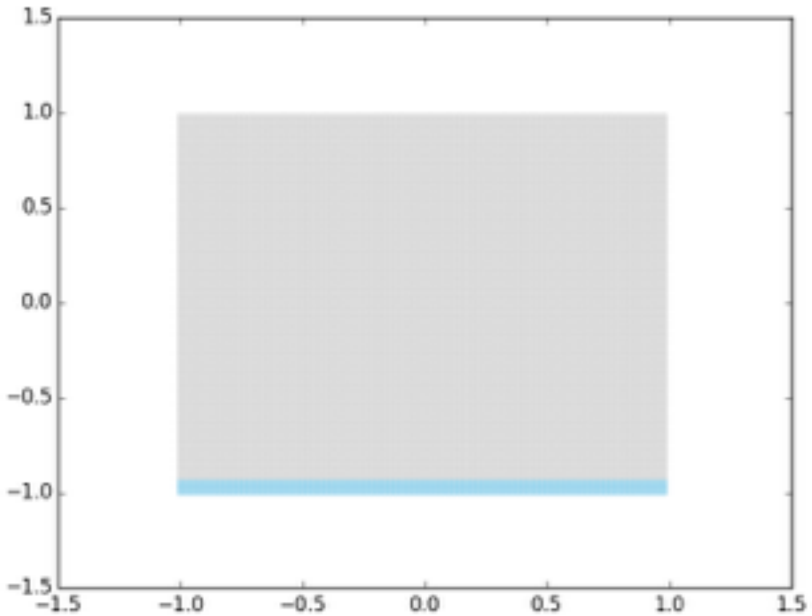


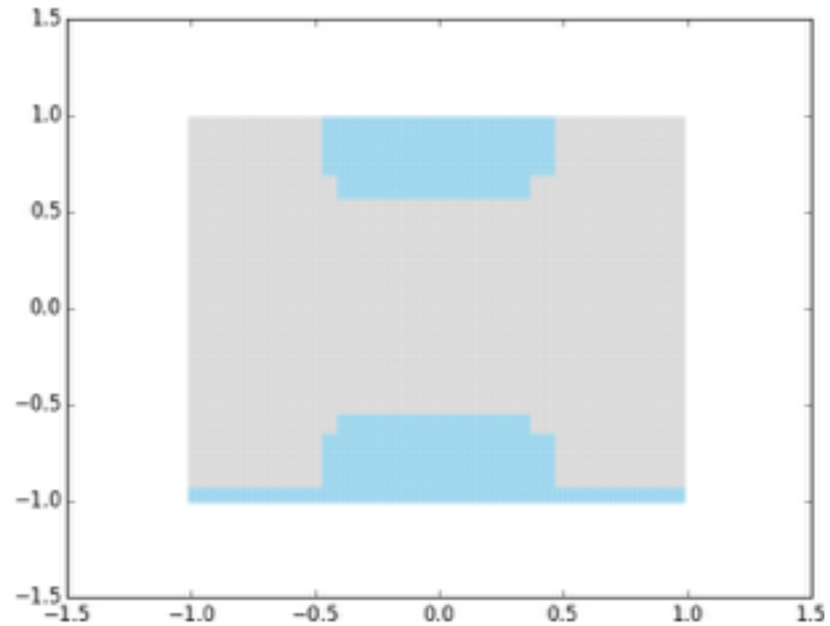


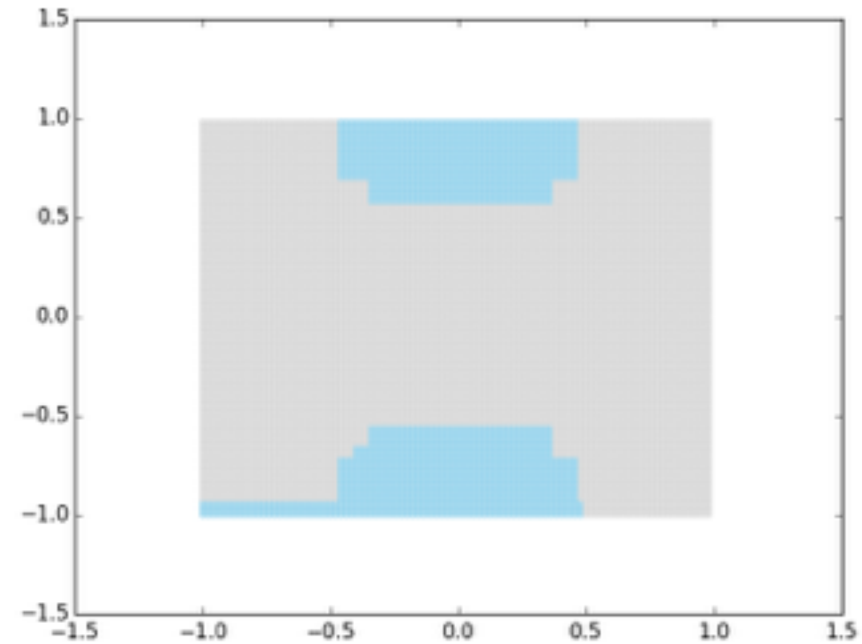
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
70
80
90

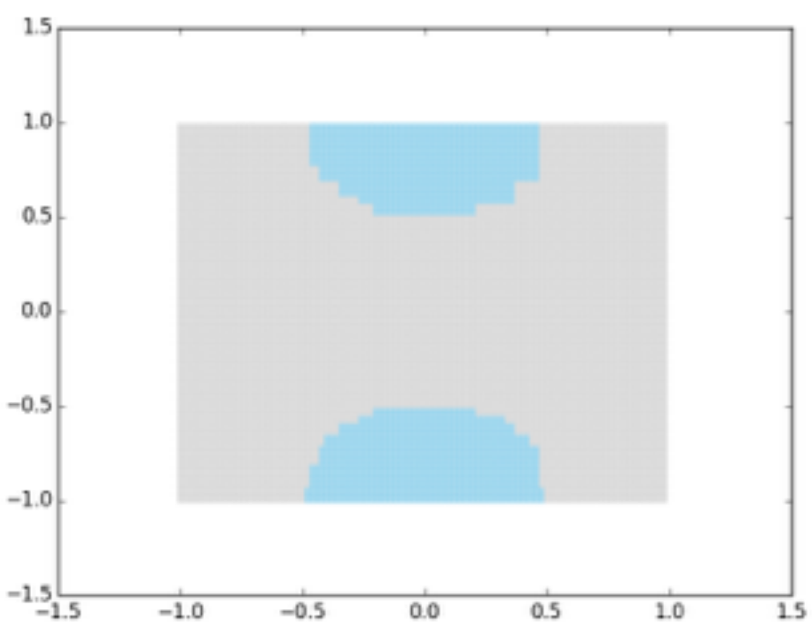


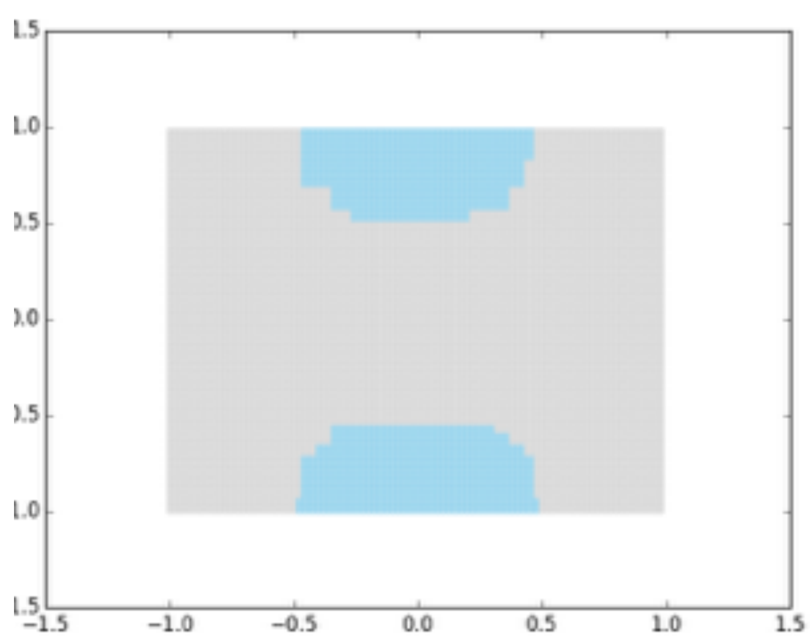
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
70
80
90

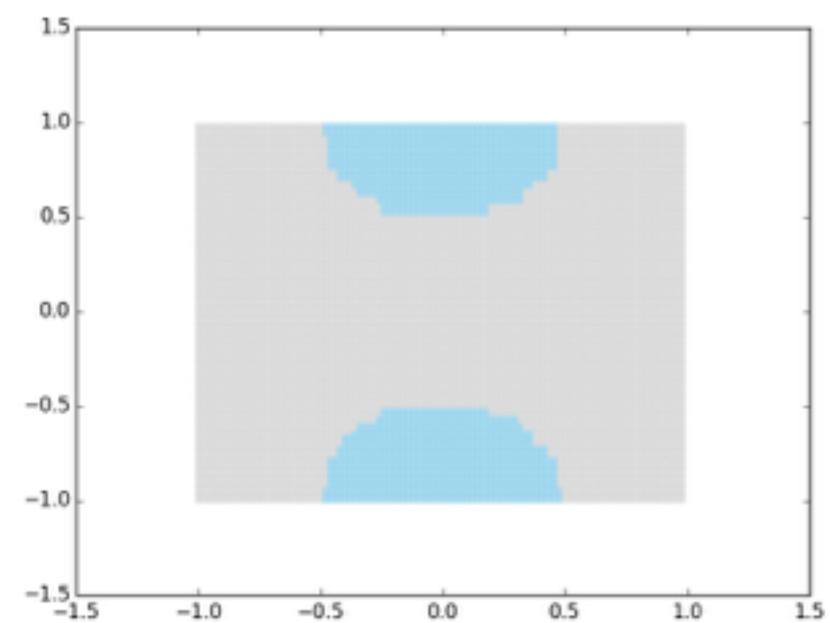

1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5


1.5
1.0
0.5
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5

Why Adaboost is good?

- Bound for training error $$\prod_t \left[2\sqrt{\epsilon_t(1-\epsilon_t)}\right] = \prod_t \sqrt{1-4\gamma_t^2} \le \exp\left(-2\sum_t \gamma_t^2\right).$$
- Bound for generalization error

$$\hat{\Pr}\left[H(x) \neq y\right] + \tilde{O}\left(\sqrt{\frac{Td}{m}}\right)$$

$$\hat{\Pr}\left[\text{margin}(x,y) \le \theta\right] + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right)$$