CIS 519 Recitation 9

Boosting / Generative Models

Question 1

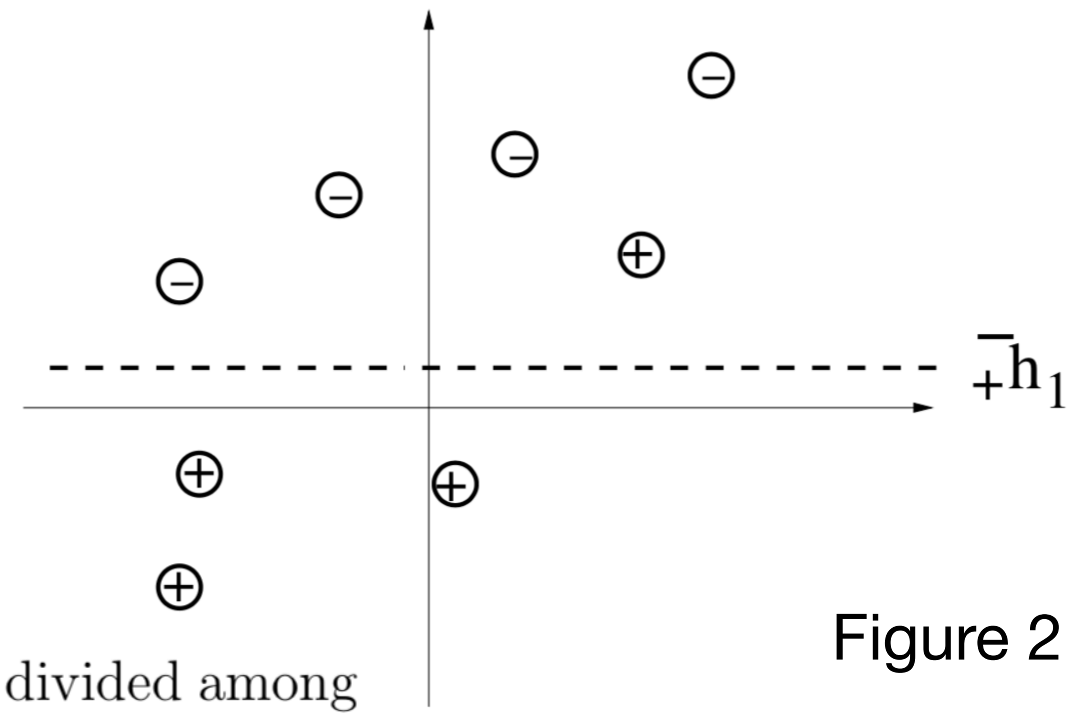


Figure 2

1. **(3 points)** Figure 2 shows a dataset of 8 points, equally divided among the two classes (positive and negative). The figure also shows a particular choice of decision stump h_1 picked by AdaBoost in the first iteration. What is the weight α_1 that will be assigned to h_1 by AdaBoost? (Initial weights of all the data points are equal, or 1/8.)

$\ln(x) \Rightarrow \log_2(x)$
$e^x \Rightarrow 2^x$

2. **(T/F – 2 points)** AdaBoost will eventually reach zero training error, regardless of the type of weak classifier it uses, provided enough weak classifiers have been combined.

3. **(T/F – 2 points)** The votes α_i assigned to the weak classifiers in boosting generally go down as the algorithm proceeds, because the weighted training error of the weak classifiers tends to go up

4. **(T/F – 2 points)** The votes α assigned to the classifiers assembled by AdaBoost are always non-negative

Answers:

1. $\log_2 \sqrt{7}$
2. F
3. T
4. T

Question 2

link: dataset

Recall that Adaboost learns a classifier H using a weighted sum of weak learners h_t as follows

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

In this question we will use decision trees as our weak learners, which classify a point as $\{1, -1\}$ based on a sequence of threshold splits on its features (here x, y).

1. [2 points] Assume that our weak learners are decision trees of depth 1 (i.e. decision stumps), which minimize the weighted training error. Using the dataset below, draw the decision boundary learned by h_1.

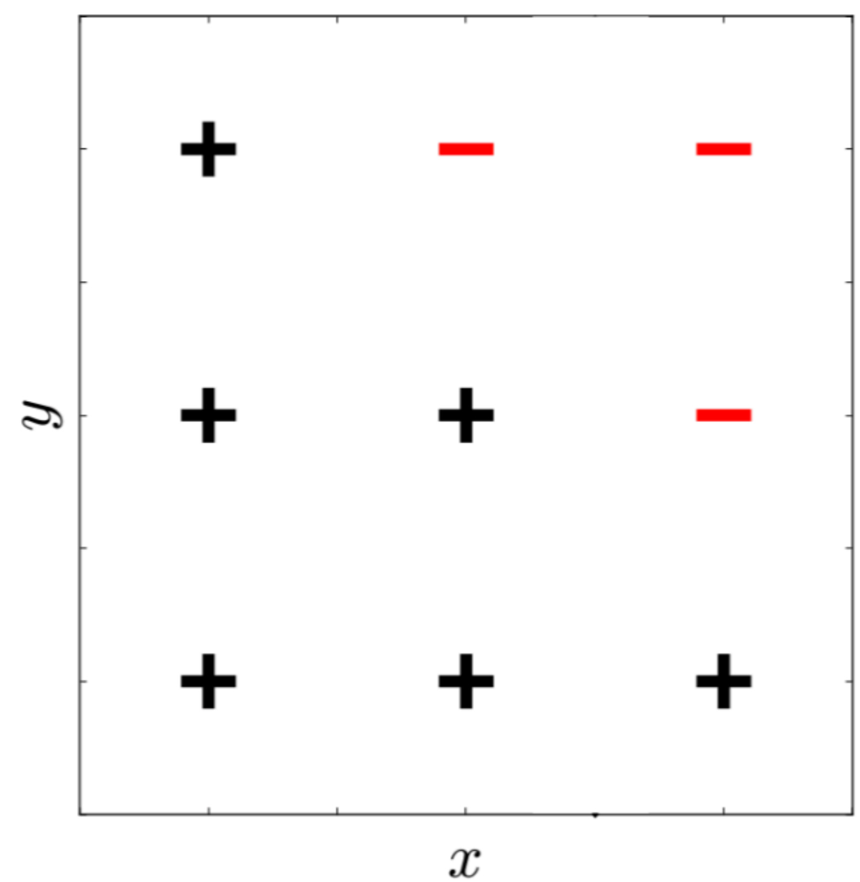
2. [3 points] On the dataset below, circle the point(s) with the highest weights on the second iteration, and draw the decision boundary learned by h_2.

3. [3 points] On the dataset below, draw the decision boundary of $H = sgn\left(\alpha_1 h_1 + \alpha_2 h_2\right)$. (Hint, you do not need to explicitly compute the α's).

4. [2 points] Now assume that our weak learners are decision trees of maximium depth 2, which minimize the weighted training error. Using the dataset below, draw the decision boundary learned by h_1.

5. [3 points] On the dataset below, circle the point(s) with the highest weights on the second iteration, and draw the decision boundary learned by h_2.

6. [3 points] On the dataset below, draw the decision boundary of $H = sgn\left(\alpha_1 h_1 + \alpha_2 h_2\right)$. (Hint, you do not need to explicitly compute the α's).

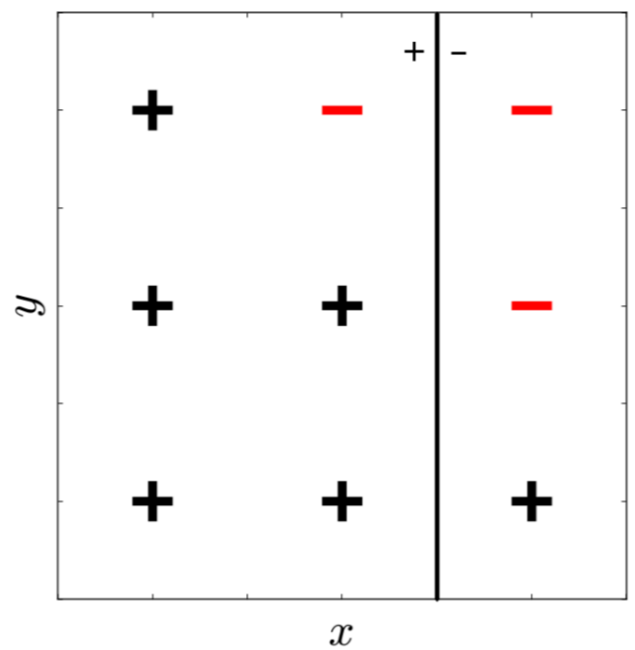


Dataset

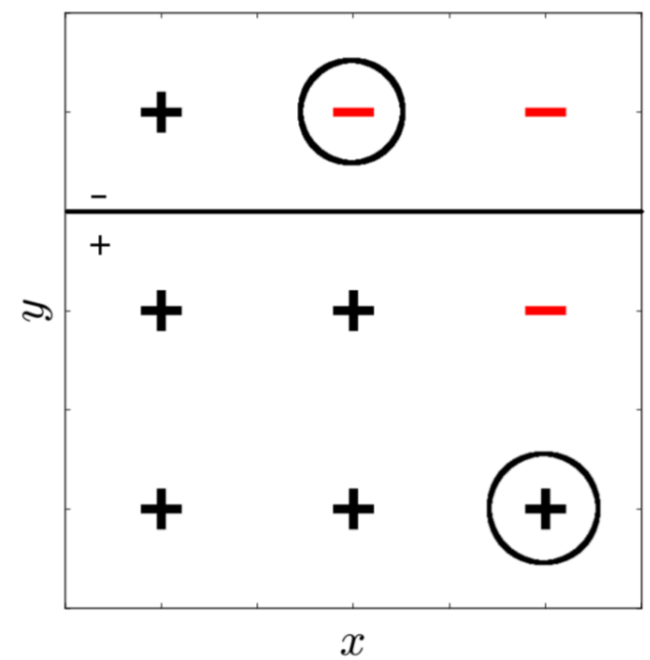
Answers:

Note: The solutions below are one of several possible answers.

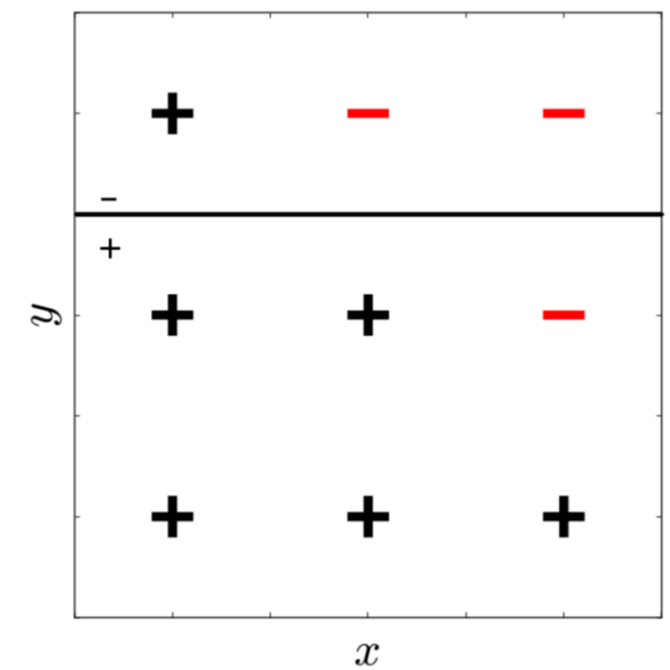
1.



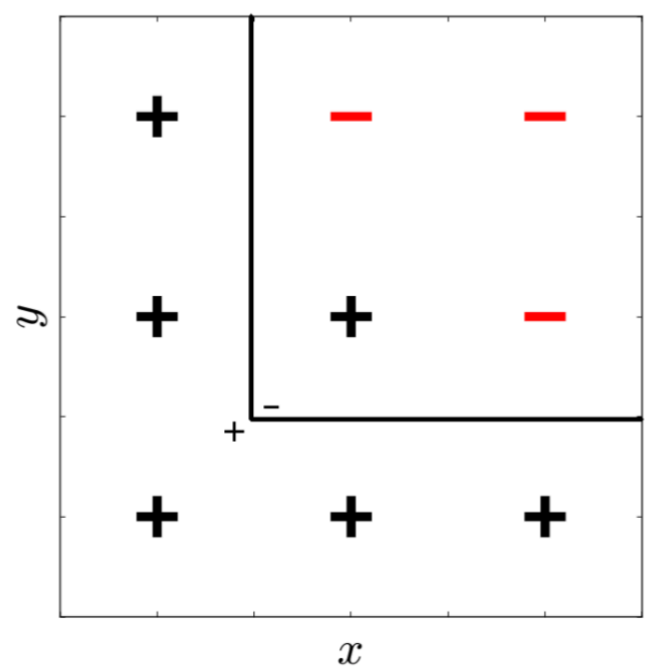
2.



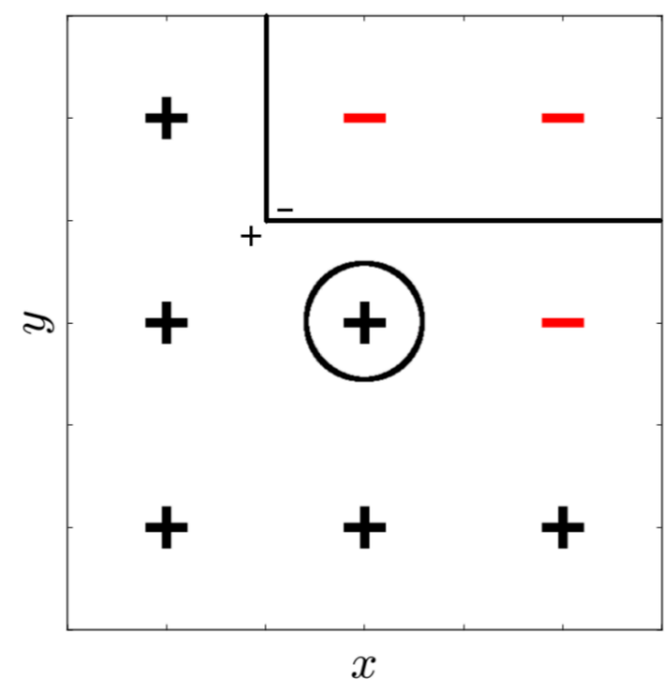
3.



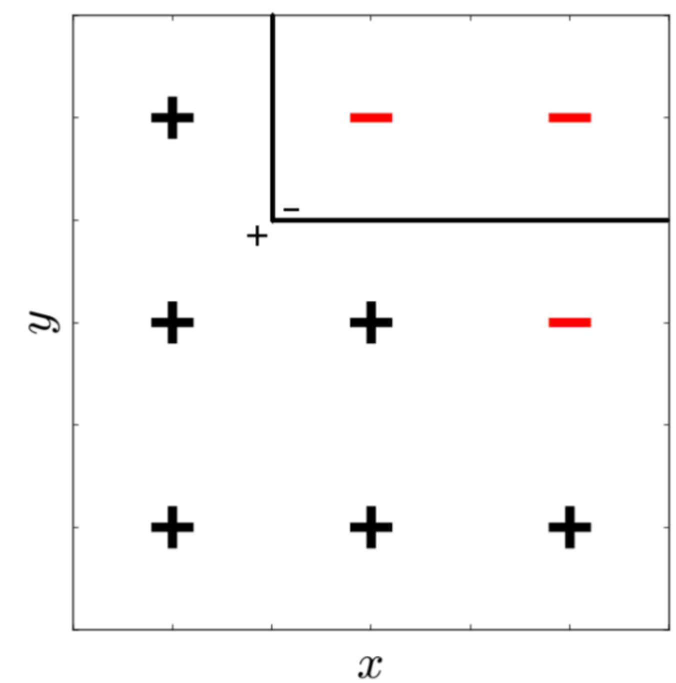
4.



5.



6.



A Multinomial Bag of Words

Our eventual goal will be: Given a document, predict whether it's "good" or "bad"

How do we model it?
What is the learning problem?

- We are given a collection of documents written in a three word language $\{a, b, c\}$. All the documents have exactly n words (each word can be either a, b or c).
- We are given a labeled document collection $\{D_1, D_2 \ \ldots \ , D_m\}$. The label y_i of document D_i is 1 or 0, indicating whether D_i is "good" or "bad".
- Our generative model uses the multinominal distribution. It first decides whether to generate a good or a bad document (with $P(y_i = 1) = \eta$). Then, it places words in the document; let a_i (b_i, c_i, resp.) be the number of times word a (b, c, resp.) appears in document D_i. That is, we have $a_i + b_i + c_i = |D_i| = n$.
- In this generative model, we have:

$$P(D_i|y = 1) = n!/(a_i! \ b_i! \ c_i!) \ \alpha_1^{a_i} \ \beta_1^{b_i} \ \gamma_1^{c_i}$$

where α_1 (β_1, γ_1 resp.) is the probability that a (b , c) appears in a "good" document.

- Similarly, $P(D_i|y = 0) = n!/(a_i! \ b_i! \ c_i!) \ \alpha_0^{a_i} \ \beta_0^{b_i} \ \gamma_0^{c_i}$
- Note that: $\alpha_0 + \beta_0 + \gamma_0 = \alpha_1 + \beta_1 + \gamma_1 = 1$

Unlike the discriminative case, the "game" here is different:
- ❑ We make an assumption on how the data is being generated.
  - ❑ (multinomial, with $\eta, \alpha_i, \beta_i, \gamma_i$)
- ❑ We observe documents, and estimate these parameters (that's the learning problem).
- ❑ Once we have the parameters, we can predict the corresponding label.

A Multinomial Bag of Words (2)

- We are given a collection of documents written in a three word language $\{a, b, c\}$. All the documents have exactly n words (each word can be either a, b or c).
- We are given a labeled document collection $\{D_1, D_2 \ \ldots \ , D_m\}$. The label y_i of document D_i is 1 or 0, indicating whether D_i is "good" or "bad".
- The classification problem: given a document D, determine if it is good or bad; that is, determine $P(y|D)$.
- This can be determined via Bayes rule: $P(y|D) \ = \ P(D|y) \ P(y)/P(D)$
- But, we need to know the parameters of the model to compute that.

A Multinomial Bag of Words (3)

- How do we estimate the parameters?
- We derive the most likely value of the parameters defined above, by maximizing the log likelihood of the observed data.
- $PD = \prod_i P(y_i, Di) = \prod_i P(D_i | y_i) P(y_i) =$
  - We denote by $P(y_i = 1) = \eta$ the probability that an example is "good" ($y_i = 1$; otherwise 0). Then:

$$\prod_i P(y, D_i) = \prod_i [\left(\eta \frac{n!}{a_i!\ b_i!\ c_i!} \alpha_1^{a_i} \beta_1^{b_i} \gamma_1^{c_i}\right)^{y_i} \cdot \left((1-\eta) \frac{n!}{a_i!\ b_i!\ c_i!} \alpha_0^{a_i} \beta_0^{b_i} \gamma_0^{c_i}\right)^{1-y_i}$$

- We want to maximize it with respect to each of the parameters. We first compute $\log(PD)$ and then differentiate:
- $\log(PD) = \sum_i y_i\ [\log(\eta) + C + a_i \log(\alpha_1) + b_i \log(\beta_1) + c_i \log(\gamma_1)] + (1 - y_i) [\log(1-\eta) + C' + a_i \log(\alpha_0) + b_i \log(\beta_0) + c_i \log(\gamma_0)]$
- $\frac{dlogPD}{\eta} = \sum_i [\frac{y_i}{\eta} - \frac{1-y_i}{1-\eta}] = 0 \rightarrow \sum_i (y_i - \eta) = 0 \rightarrow \boxed{\eta = \sum_i \frac{y_i}{m}}$
- The same can be done for the other 6 parameters. However, notice that they are not independent: $\alpha_0 + \beta_0 + \gamma_0 = \alpha_1 + \beta_1 + \gamma_1 = 1$ and also $a_i + b_i + c_i = |D_i| = n$.

Labeled data, assuming that the examples are independent

Notice that this is an important trick to write down the joint probability without knowing what the outcome of the experiment is. The ith expression evaluates to $p(D_i, y_i)$ (Could be written as a sum with multiplicative y_i but less convenient)

Makes sense?