CSCI 567, Summer 2022
Prof. Victor Adamchik

Theory Assignment 3

Due: 11:59 pm, June 20, 2022

Instructions

Submission: Assignment submission will be via courses.uscden.net. By the submission date, there will be a folder named 'Theory Assignment 1' set up in which you can submit your files. Please be sure to follow all directions outlined here.

You can submit multiple times, but only *the last submission* counts. That means if you finish some problems and want to submit something first and update later when you finish, that's fine. In fact you are encouraged to do this: that way, if you forget to finish the homework on time or something happens (remember Murphy's Law), you still get credit for whatever you have turned in.

Problem sets must be typewritten or neatly handwritten when submitted. In both cases, your submission must be a single PDF. It is strongly recommended that you typeset with LaTeX. There are many free integrated LaTeX editors that are convenient to use (e.g Overleaf, ShareLaTeX). Choose the one(s) you like the most. This tutorial Getting to Grips with LaTeX is a good start if you do not know how to use LaTeX yet.

Please also follow the rules below:

- The file should be named as `firstname_lastname_USCID.pdf` e.g., `Don_Quijote_de_la_Mancha_8675309045.pdf`).
- Do not have any spaces in your file name when uploading it.
- Please include your name and USCID in the header of your report as well.

Collaboration: You may discuss with your classmates. However, you need to write your own solutions and submit separately. Also in your report, you need to list with whom you have discussed for each problem. Please consult the syllabus for what is and is not acceptable collaboration. Review the rules on academic conduct in the syllabus: a single instance of plagiarism can adversely affect you significantly more than you could stand to gain.

Notes on notation:

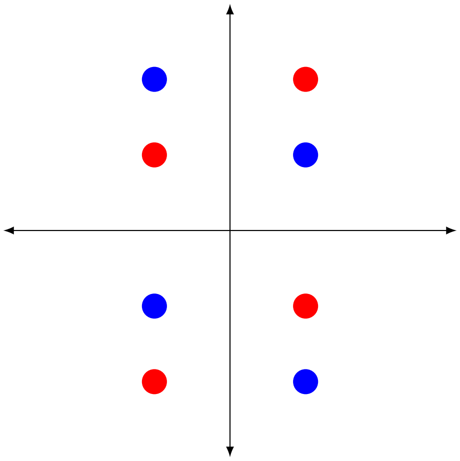
- Unless stated otherwise, scalars are denoted by small letter in normal font, vectors are denoted by small letters in bold font and matrices are denoted by capital letters in bold font.
- $\|.\|$ means L2-norm unless specified otherwise i.e. $\|.\| = \|.\|_2$

Problem 1 Adaboost (36 points)

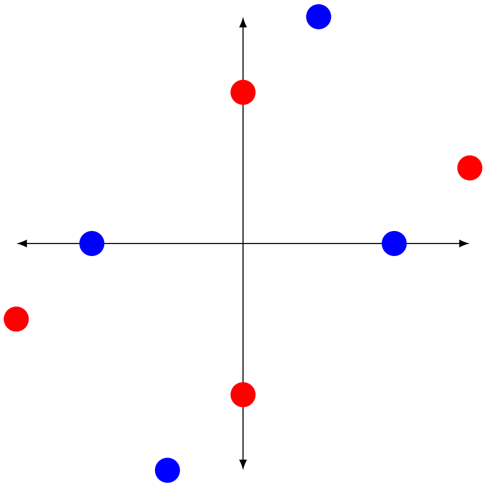
In the lecture, we learnt that we can use boosting to learn a good classifier from an ensemble of weak classifier. In particular, Adaboost algorithm (see algorithm 1), does this by iteratively reweighing the samples and fitting a weak classifier to the new data. The final classifier is weighted ensemble of all the weak classifiers.

Algorithm 1 AdaBoost Algorithm

```
Given: $\mathcal{H}$: A set of functions, where $h \in \mathcal{H}$ takes a D-dimensional vector as input and outputs $+1$ or $-1$
Given: A training set $\{(x_n \in^D, y_n \in \{+1, -1\})\}_{n=1}^N$
Goal:Learn $F(x) = sgn(\sum_{t=1}^T \beta_t f_t(x))$, where $f_t \in \mathcal{H}, \beta_t \in, sgn(a) = \begin{cases} +1, & \text{if } a \geq 0 \\ -1, & \text{otherwise} \end{cases}$
Initialization: $w_1(n) = 1/N$                                ▷ Start with equal weights
for $t = 1 \ldots N$ do
    $f_t = \arg\min_{h \in \mathcal{H}} \sum_n w_t(n) \mathbb{I}[y_n \neq h(x_n)]$          ▷ Fit a weak classifier
    $\epsilon_t = \sum_n w_t(n) \mathbb{I}[y_n \neq h(x_n)]$                          ▷ Compute the error
    $\beta_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$
    $w_{t+1}(n) = \begin{cases} w_t(n) \exp(-\beta_t) & \text{if } y_n = f_t(x_n) \\ w_t(n) \exp(\beta_t) & \text{if } y_n \neq f_t(x_n) \end{cases}$   ▷ Update weights
    $w_{t+1}(n) \leftarrow \frac{w_{t+1}(n)}{\sum_{n'} w_{t+1}(n')}$                     ▷ Normalization
return $F(x) = sgn(\sum_{t=1}^T \beta_t f_t(x))$
```



Data for problem 1.1, 1.2



Data for problem 1.3, 1.4, 1.5

In this problem, we consider weak classifier of following type:

$$h_{s,b,d} = \begin{cases} s & \text{if } x_d > b \\ -s & \text{otherwise} \end{cases}$$

where $s \in \{-1, 1\}, b \in \mathbf{R}, d \in \{1 \ldots D\}$. Such weak classifiers are called decision stumps as they can also be seen as one-level decision tree. Note that for this problem, if you have two classifiers achieving the same error, pick either one.

We are given the following data:

$$\mathcal{D} = \{(x_1, y_1) = ([-1,-2], -1), (x_2, y_2) = ([-1,-1], 1), (x_3, y_3) = ([-1,1], -1), (x_4, y_4) = ([-1,2], 1),$$
$$(x_5, y_5) = ([1,-2], 1), (x_6, y_6) = ([1,-1], -1), (x_7, y_7) = ([1,1], 1), (x_8, y_8) = ([1,2], -1)\}$$

We want to run adaboost upto $T = 3$ iterations.

1.1 Compute first iteration of adaboost algorithm. Clearly write down f_1, β_1, ϵ_1 and w_2. **(8 points)**

1.2 Compute second iteration of adaboost algorithm. Clearly write down f_2, β_2, ϵ_2 and w_3. Can you tell the outcome of this adaboost algorithm without doing the third step? **(8 points)**

For the next problems, we linearly transform the dataset by multiplying with the matrix $\mathbf{W} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$ Under this transformation, the data now becomes:

$$\text{D} = \{(\text{x}_1, y_1) = ([-3,-1], -1), (x_2, y_2) = ([-2,0], 1), (x_3, y_3) = ([0,2], -1), (x_4, y_4) = ([1,3], 1),$$
$$(x_5, y_5) = ([-1,-3], 1), (x_6, y_6) = ([0,-2], -1), (x_7, y_7) = ([2,0], 1), (x_8, y_8) = ([3,1], -1)\}$$

We will run first two iterations of adaboost algorithm on the transformed data.

1.3 Compute first iteration of adaboost algorithm. Clearly write down f_1, β_1, ϵ_1 and w_2. **(8 points)**

1.4 Compute second iteration of adaboost algorithm. Clearly write down f_2, β_2, ϵ_2 and w_3. (round up to 3 decimal places, e.g. 0.001) **(8 points)**

1.5 Write down $F(x)$ after two iterations. **(4 points)**

Problem 2 PCA (32 points)

Consider the following design matrix, representing four sample points $X_i \in R^2$.

$$X = \begin{bmatrix} 4 & 1 \\ 2 & 3 \\ 5 & 4 \\ 1 & 0 \end{bmatrix}$$

We want to represent the data in only one dimension, so we turn to principal components analysis (PCA).

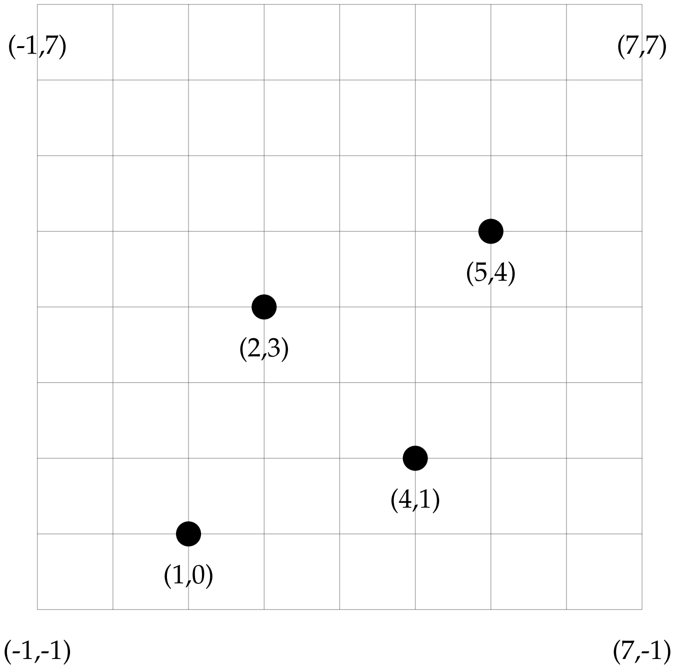
2.1 Which of the following are true about principal components analysis (PCA)? Assume that no two eigenvectors of the sample covariance matrix have the same eigenvalue.

Choose all of the right choices **(3 points)**

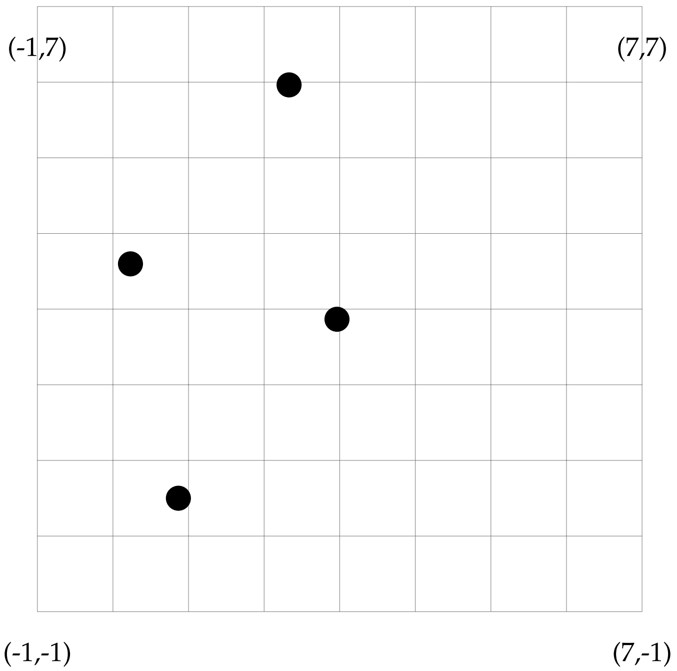
1. A: Appending a 1 to the end of every sample point doesn't change the results of performing PCA (except that the useful principal component vectors have an extra 0 at the end, and there's one extra useless component with eigenvalue zero).
2. B: If you perform an arbitrary rigid rotation of the sample points as a group in feature space before performing PCA, the largest eigenvalue of the sample covariance matrix does not change.
3. C: If you use PCA to project d-dimensional points down to j principal coordinates, and then you run PCA again to project those j-dimensional coordinates down to k principal coordinates, with $d > j > k$, you always get the same result as if you had just used PCA to project the d-dimensional points directly down to k principle coordinates.
4. D: If you perform an arbitrary rigid rotation of the sample points as a group in feature space before performing PCA, the principal component directions do not change.

2.2 Compute the unit-length principal component directions of X, and state which one the PCA algorithm would choose if you request just one principal component. Please provide an exact answer, without approximation. (You will need to use the square root symbol.) Show your work here. **(9 points)**

2.3 The plot below depicts the sample points from X. We want a one-dimensional representation of the data, so draw the principal component direction (as a line) and the projections of all four sample points onto the principal direction. Label each projected point with its principal coordinate value (where the origin's principal coordinate is zero). Give the principal coordinate values exactly. **(10 points)**



2.4 The plot below depicts the sample points from X rotated 30 degrees counterclockwise about the origin. As in part (b), identify the principal component direction that the PCA algorithm would choose and draw it (as a line) on the plot. Also draw the projections of the rotated points onto the principal direction. Label each projected point with the exact value of its principal coordinate. **(10 points)**


(-1,7)
(7,7)
(-1,-1)
(7,-1)

Problem 3 Naive Bayes (32 points)

In this problem we try to predict whether it is suitable for playing tennis or not based on the weather condition, the emotion and the amount of homework, using Naive Bayes Classifier. You can think ”Play Tennis” is a label and ‘PlayTennis = Yes‘ means it is suitable for playing tennis. We assume the probability P(Weather, Emotion, Homework | PlayTennis) can be factorized into the product form such that

$$P(Weather, Emotion, Homework|PlayTennis) =$$

$$P(Weather|PlayTennis)xP(Emotion|PlayTennis)xP(Homework|PlayTennis)$$

The training data is as following. Each data point has three attributes (*Weather, Emotion, Homework*) , where Weather ∈ (*Sunny , Cloudy*), Emotion ∈ (*Happy, Normal, Unhappy*), Homework ∈ (*Much, Little*).

Weather	Emotion	Homework	PlayTennis
Sunny	Happy	Little	Yes
Sunny	Normal	Little	Yes
Cloudy	Happy	Much	Yes
Cloudy	Unhappy	Little	Yes
Sunny	Unhappy	Little	No
Cloudy	Normal	Much	No

1. What are the probabilities of P(PlayTennis = *Yes*) and P(PlayTennis = *No*)? Each of your answer should be an irreducible fraction. **(8 points)**

2. Write down the following conditional probabilities. Each of your answer should be an irreducible fraction. **(8 points)**

   (a) P(Weather = *Sunny* | PlayTennis = *Yes*) =?

   (b) P(Emotion = *Normal* | PlayTennis = *Yes*) =?

   (c) P(Homework = *Much* | PlayTennis = *Yes*) =?

3. Given the new data instance x = (Weather = *Sunny*, Emotion = *Normal*, Homework = *Much*), which of the following has larger value: P(PlayTennis = *Yes* | x) or P(PlayTennis = *No* | x)? Each of your answer should be an irreducible fraction. **(16 points)**