# HW4 – Gal Kaptsenel 209404409

## Q1

1. Yes.

Let’s split the proof according to the value of

* ,

And indeed,

* + If

1. has a maximum value in respect to at point , and therefore the value of bounded by a maximum of .

Therefore

* + Otherwise, ,



Therefore

* ,

And indeed,

* + If

Therefore

* + If

Therefore

Therefore, at call cases we conclude that

Yes, the algorithm will converge to a minimum at with value .

Lets prove that is a series of points which obeys

Proof by induction over the iteration number,

Iteration 0

At iteration , , and therefore , and then we will get that,

Therefore,

And therefore, the minimized function will converge to a value of .

Indeed is a non-negative function, which gets a value of 0 at point , and therefore the gradient decent algorithm indeed converges to the minimum.

|  |  |  |  |
| --- | --- | --- | --- |
| I |  |  |  |
| 0 |  |  |  |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
|  |  |  |  |

No,

The series of will be , that is, the algorithm will alternate between -1 and 1, and will never converge to the minimum which, as stated at 1.3 above, is at .

Lets prove it by showing that each iteration of the algorithm, or .

Proof by induction over the iteration number,

Iteration 0

and therefore the statement holds.

* If

, and therefore,

* If

, and therefore,

Therefore, it holds that at all cases, equals to 1 or -1, and therefore the statement holds.

And indeed, as can be seen from the first three iterations,

|  |  |  |  |
| --- | --- | --- | --- |
| I |  |  |  |
| 0 |  |  |  |
| 1 |  |  |  |
| 2 |  |  |  |
|  |  |  |  |

The algorithm will alternate between and , and will not converge.

## Q2

Denote a random variable,

And it holds that given and , . Note that,

1. define
2. it is given that

Therefore,

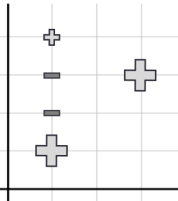
Therefore,

1. conditional probability + definition of
2. from above, , and because the samples are independent of the chosen vector of weights
3. and therefore it does not affect the which maximizes the expression, and therefore we can omit it from the expression.
4. , constant value, could be extracted from the multiplication operator + exponent rules + the constant value could be extracted from the summation operator .
5. is a monophonic ascending function, and therefore it doesn’t affect the which maximizes the expression.
6. is a constant value, therefore it doesn’t affect the which maximizes the expression.
7. maximizing the expression is the same as minimizing the expression
8. Multiplying the expression by , which is a constant value, doesn’t affect the which Minimizes .

## Q3

### Figure **(a)**.

Figure **(a)** could be accomplished using the following weak classifier,



This weak classifier, which separates using the y-axis value, could be chosen because it succeeds in classifying 3 out of 5 samples. There is no classifier which separates using only a single y-value or x-value, which succeeds in separating 4 or more samples, and therefore this classifier could be chosen.

Any classifier which separates using only a single x-value, will yield the same classification for all the left samples, and therefore will be mistaken for at least 2 samples (and therefore correct for at most 3 samples).

Any classifier which separates only using a single y-value, will be mistaken on one of the ‘+’ and ‘–‘ samples with the same y-value, and in addition, because there is two ‘-‘ samples in between two ‘+’ samples (in respect to the y-values), any y-value weak classifier will be mistaken over at least (another) one sample. Therefore, any y-value weak classifier will be mistaken over at least two samples (correct for at most 3 samples).

Any of the other figures could not be the result of AdaBoost with a weak classifier,

* **(b)** – as described above, there is no weak classifier for the given samples and features, which will accomplish less then 2 incorrect classifications, but according to figure **(b)**, the classifier obtained at the first iteration only classifies incorrectly a single sample, therefore it is impossible to achieve this figure after a single iteration.
* **(c)** –The classifier must successfully separate most of the left samples, and because all of them got the same x-value, the classifier must separate them using the y-value.

The classifier must chose a y value that causes the two middle left “-“ samples to be classified differently, therefore the chosen y value must be in between them, and indicate that all points beneath it are “+”.

On the other hand, the upper left “+” is classified currently, and also the “-“ beneath it, and therefor the chosen y value should be in between those two samples, and indicate that all points above it are “+”.

Therefore, The chosen classifier must indicate that all points above and beneath it are “+”, which is impossible according to the stump classifiers class.

* **(d)** – the weak classifier that is chosen will be mistaken over 3 out of the 5 samples, but there exists a classifier which successfully classifies 3 out of the 5 samples correctly, for example the one we described above, or for example a classifier which returns ‘+’ for any positive y-value. Therefore, the AdaBoost algorithm, which is a greedy algorithm, will not chose the classifier which resulted in figure **(d)**.