Intro To ML – HW2

students:

Dor Bank - 301380416 - dorbank@gmail.com, Avi Caciularu - 203056585 - avi.c33@gmail.com

**Theory Questions**

**Question 1:**

1. Consider the following problems:

We will show that It doesn’t matter whether the following constrain is included or not:

.

\*\*\*\*\*addition???

for that, all we need to show is that even without the constraint, for the optimal solution, which is the global minimum, so let's assume by contradiction there the global minimum GM contains a which is negative. then for that we get . meaning also satisfies the constrain. because we conclude that .

therefore, we get:

GM =

in contradiction for GM being the global minimum.

\*\*\*\*\*end addition

We will show that problem (1)’s parameters are equivalent to problem (2)’s:

* : Proof by contradiction: Let be the parameters that minimize the function in problem (1), and let be the parameters that minimize the function in problem (2), so that (2)’s target function gives a smaller solution than (1)’s. We get that are the parameters that minimize the function in problem (1) and also satisfy the constraint

, but contradicting that (2) gives the smallest solution.

* : Proof by contradiction: Let be the parameters that minimize the function in problem (2), and let be the parameters that minimize the function in problem (1), so that (1)’s target function gives a smaller solution than (2)’s. It’s obvious that (1)’s parameters satisfy the (2)’s constrains, so again, we get the same minimal value for both (1) and (2) target functions, in contradiction.



By comparing (1) to 0 we get the relation: .

By comparing (2) to 0 we get the new constrain: .

By comparing (3) to 0 we get the relation: or .

why not deriving by alpha also??

So now our Lagrangian is:

=

1. Our dual problem is:

.

**Question 2:**

a. using the \* notations, no need to write the constrains because they are already satisfied. so we get:

notice that on the right side we didn't use to ease notation.

b.

maximizing will occur by minimizing . by constraints we'll set .

maximizing the right side we set that

with adding to both sides of algebra we get

taking square root from both sides we get

**Question 3:**

a. not submitting full answer. just a reminder for us that we are optimizing:

b. for we use derivation w.r.t it and compare to 0. we get the following problem:

c.

d. for a single iteration, assuming all are pre-estimated, the time and space complexity are just O(1).

**Question 4:**

a. false. take Of course they are both PDK. but for we get

and then for any given we get

b. true. = , where the final size is positive because both and are PDK which means that = is positive.

c. true. = , where the final size is positive because is a PDK and a is positive.

d. false. take and a=1. we get that is the same as in section a which is not a PDK.

e. true. Assign is clearly a PDK. notice that . from lecture 7 (slide 28) we get: where the final size is positive because is a PDK.

f. true. from the scribes and lecture, we know that if and are PDK than is also a PDK. from section c we know that the PDK remains a PDK after multiplying with a positive coefficiet.

is composed of additive degrees of with a positive coefficient. with the stated above, all that remains is to prove that the PDK are closed under "power" and we are done. but that comes directly from:

**Question 5:**

a. proof by induction on the T iteration:

basis: for we had no iterations so the claim is clearly satisfied.

step: assume the claim is true for . we shall split the next step to separate cases:

1. prediction correct:

2. prediction mistake: **.** by the induction assumption we directly get our proof by construction.

b. As before, . Therefore, which is the kernel version of the perceptron.

Algorithm:

1. start with the all-zeros weight vector , and initialize *t* to 1.

2. Given a sample predict positive iff

3. on correct prediction, set .

4. On mistake, set .

for demonstration, let's say that *t*=4 and the previous predictions were:

in that case, we get:

|  |  |  |
| --- | --- | --- |
| t |  |  |
|  | 1 | -1 |
|  | 0 | 1 |
|  | 1 | 1 |

So given the sample the algorithm will predict positive iff

and act accordingly. it is easy to see from this example that the samples who were correctly predicted are not taken into account and the others influence the weights similarly as the original.