Intro To ML – HW2

students:

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

**Theory Questions**

Note:  
We denote in this exercise vectors in **bold** font, e.g. and

**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 .

So now our Lagrangian is:

=

1. Our dual problem is:

.

**Question 2:**

1. By sing the star notation (\*) for the optimal solution, we don’t need to write the constrains, because they are already satisfied. We get:

On the right member, we didn't use for ease notation.



So as required.

**Question 3:**

1. Not the full answer, just a reminder that we try to optimize:

1. Firstly, we evaluate the expression:

For not fixed, and fixed we will use the partial derivative compared to 0. First, the problem in section a, for un-fixed is equivalent to:

We can see the objective function is concave.

* If or so any , fits.
* If and then

1. The expression under the takes and we it . In total, time complexity is . We should keep in memory all our samples, s and labels, so space complexity is

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

**Question 4:**

1. This statement is false. Counter-example:

Given and , and are valid positive definite kernels. So, which is negative definite, because

.

1. This statement is correct. Proof:

Assumption- and are valid positive definite kernels.

Positive definite:

. is positive from the definition of the norm of un-zeroed vector, and gives a positive value because of the assumption.

In total, we get that hence .

Symmetry:

.

1. This statement is correct. Proof:

Assumption- is a valid positive definite kernel and .

Positive definite:

. gives a positive value, and because of the assumption. In total, we get that hence.

Symmetry:

.

1. This statement is false. Counter-example:

Given and , is a valid positive definite kernel. So, which is negative definite, because .

1. This statement is false. Counter-example:

Given

For and the positive definition condition is not satisfied:

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.

1. This statement is correct. Proof:

Assumption- is a valid positive definite kernel and is a polynomial with positive coefficients- .

Positive definite:

First, we will prove by induction that for any valid positive definite kernel , is valid positive definite kernel too:

Base:

For : is valid positive definite kernel.

For : is a valid positive definite kernel from the multiplication property proved in class.

Step:

Assuming is a valid positive definite kernel, is a valid positive definite kernel as well from the multiplication property proved in class: .

We proved in class the addition property.

We proved in section the multiplication by a positive factor property.

In total, we get that is a valid positive definite kernel.

Symmetry:

**Question 5:**

1. Proof by induction on the Tth iteration:

Base:

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

Step:

Assume the claim is true for , and we will prove for :

* Prediction is correct:
* Prediction is mistake: **.** By the induction assumption we directly get our proof by construction.

1. the weight vectorcan be expressed by where is the number of times that was misclassified. Therefore, which would be the kernel version of the perceptron.

Algorithm:

1. start with the all-zeros weight vector , and initialize t: .
2. Given a sample , predict positive iff
   1. If the prediction was correct, set .
   2. If the prediction was mistaken, set .
3. Go over 1.

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 |

Given the sample the algorithm will predict positive iff

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