Intro To ML – HW3

students:

Dor Bank - 301380416 - [dorbank@gmail.com](mailto:dorbank@gmail.com),

Avi Caciularu - 203056585 - avi.c33@gmail.com

**Theory Questions**

**Question 1:**

We will show a lower bound: . Let be a unit vector from the standard basis of size , where '' appears in the place.

First, we will show that the set can be shattered by . To show that, we should prove that every label assignment for is consistent with some hypothesis in . Let be the assignment for the vectors . Define hypothesis as:

.

Furthermore, for the computation of the result at the output node we will set:

We get

Note that here we assume that . the permutation for is trivial.

**Question 2:**

In this questions we will use the following lemmas, proved in recitation 7:

Lemma 1:

Lemma 2:

Lemma 3:

1. Denote the function family of each node *j* in layer *i* as . In our case, since the node separators are independent of each other, we get that for each layer *i*, . Because there are d nodes at each layer, we can see that all are equal to our wanted . From Radon's theorem, we get that for each , . from Lemma 2 we get that
2. is simply composed from concatenating the family times. Explicitly:

. From Lemma 3 we get that

1. For every node, we have parameters for the weight vector w, and more for the bias . In total, we have parameters, layers with nodes, and layer (the final one) with one node. Combining all together we get

.

1. Assume . First notice that . otherwise we get which is surely not true for a large enough .

Now, by taking the log of each size we get:

, where the second transition is true as stated above.

1. first notice that . Now, let there be a set of size that is shattered. Then . combining this with the inequality mentioned we get . from section d we get that . Thus, we conclude that .

**Question 3:**

1. Consider the new primal problem of SVM:

The regular SGD step was:

Denote so

We will show that:

And then the new SGD with the “Subgradient Projection Method” step would be:

The definition of the projection of on ( ) is .

If then the trivial solution is , so we will assume that :

* If so and then .
* If so and we get that the normalized vector and then , because is the optimal solution for .

1. Let be a convex set, and .

Because is a convex set, there is so that .

Let . Particularly, this equality is correct for : Hence . From the Pythagoras’s theorem, we get that meaning that . Hence, we get by the triangle inequality.

1. We have the same requirements of the original proof. Hence the expression in the equation after (12) fulfills:

.

After that, we continue the analysis and get:

and after that (only after equation (13)), the rest of the proof remains the same, and Theorem 1.1 still holds for stochastic subgradient projection Method.

**Question 4:**

1. By substituting in the given problem, we get the following loss function:

Where .

Now, map the set into . So now and meaning , and we get ( instead of (.

In the binary case, recall that the best weight solutions , must be in exact opposite direction on the plain, meaning =-1. Substitute:

and we get the decision rule, detailed:

* + - if , then
    - if , then

By assigning and adding to both members we get:

* + - if , then
    - if , then

Thus, it means that which is exactly the optimal binary classifier with weight vector .

1. First, we compute the derivative of with respect to :

Denote .

Hence the gradient’s output given :

.

In total, we get: .

Using the computed gradient for our new step in our SGD algorithm, we derive the algorithm:

1. For finding the kernel version of our algorithm, we could solve the dual problem and derive the optimal , despite this, we will try to retrieve the kernel version as we did in the kernel perceptron algorithm.

As seen in section b, a weight vector can be written as:

.

We can notice that can be expressed as the following linear combination:

where the coefficient for class and time .

Combining the last section, we derive the following algorithm:

Therefore, we must save all which costs , and for each new sample , we can use the classifier:

As seen by section b, the weight vector's matrix can be expressed by where the dependency on relies on the regularization gradient (can be proved by easy induction the same way as the perceptron proof from last exrecise). Therefore, classification becomes =  **=**  which would be the kernel version of our SVM.

Notice that each is a K size vector, and is equal to one of the options stated in the following algorithm.

We get the algorithm:

1. Initialize the weight vector's matrix with , and initialize t: .
2. Given a sample , classify by
   1. if classified correct, set
   2. if classified incorrect, set
3. Go over 1.

{comment: and are the classification estimate and the true label of }

**Question 5:**

Proof by induction on :

base: For there can’t be any samples, so only 1 node is needed to return always the required answer ( or ).

Step: Assume the claim is true for a classifier with domain dimensions up to a specific . That means that there is a decision tree of height at most as stated in the question. For a classifier with dimensions, we use the first node as .

we get sub trees with dimension (which are relevant), and by the induction assumption, those sub trees are of height at most . Therefore, the new tree is of height at most as needed.

Now, notice that each binary tree of height has leaf nodes (and therefore different paths). Moreover, in domain of there can be maximum different sample vectors.

For a lower bound, we take the sample group of size which contains different vectors. Every node in the height binary tree will map to a specific sample. Each node will return the label of the corresponding label, and the correctness is obvious.

For an upper bound we tale a sample group of size . Based on what said before, contains samples such that and . Both samples will be mapped by the tree model to the same leaf node. Hence, by labeling those samples differently, we get that is not shattered.

Thus, we conclude that the VC-dimension of the class of decision trees over the domain of is .