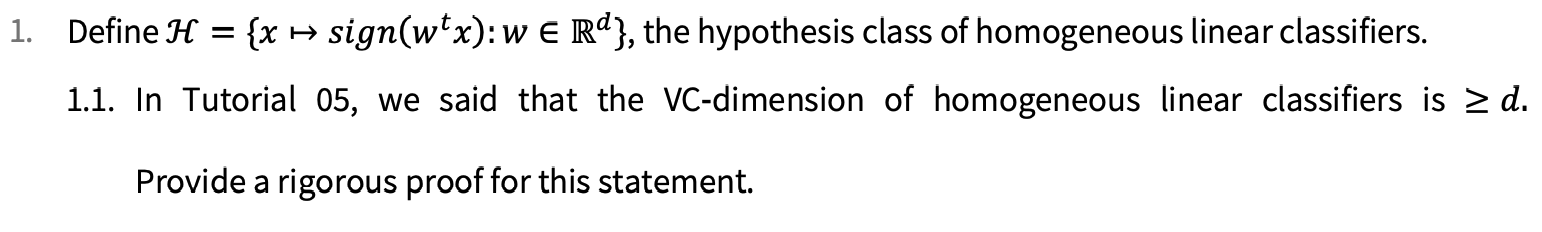
Short HW3 – SVM, Optimization, and PAC learning

Eva Poluliakhov 321882649



**Answer 1.1:**

We want to show that **there exists a set of d points** that is shattered by H.

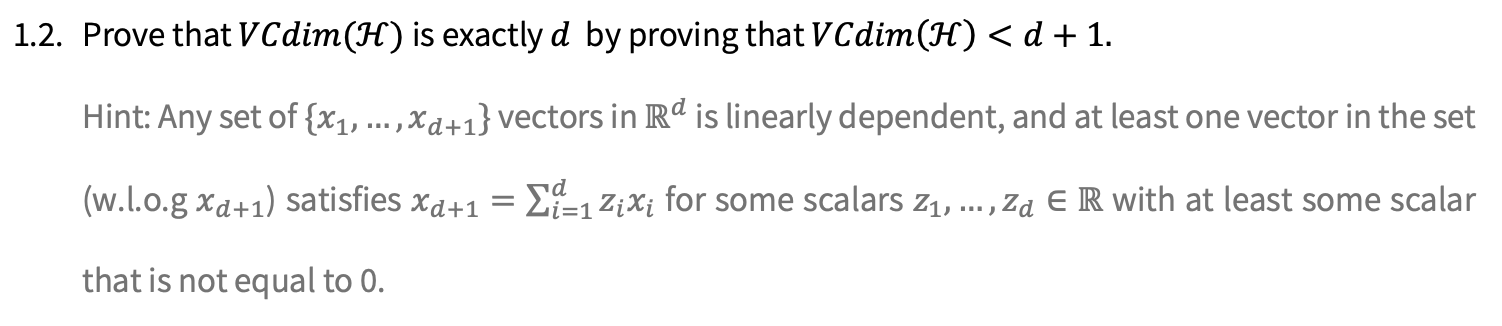
Formally:  
Given

Denote

Denote

Given

Thus



**Answer 1.2:**

We want to show that **any set of d+1 points** cannot be shattered by H.

Given any set is linearly dependent. Hence there is a non-trivial linear combination that satisfies

For the set C denote the following labels:

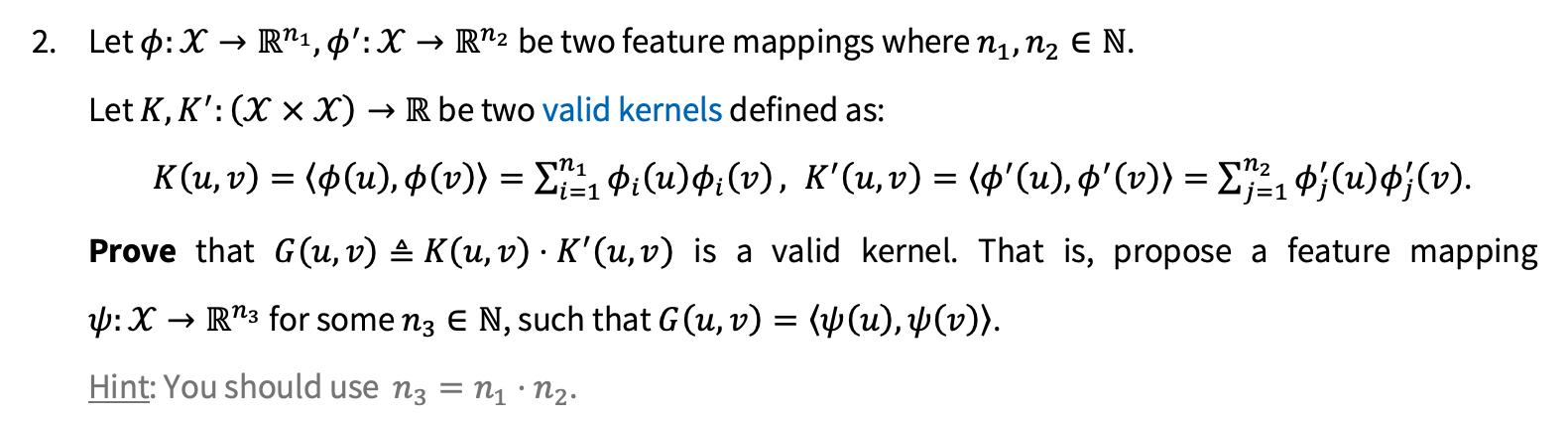
Assume that where exists that defines that shatters C.

Then the prediction for would be:

in contrediction.

Thus, do not shatters C of size d+1 so

Together with 1.1 we have that .



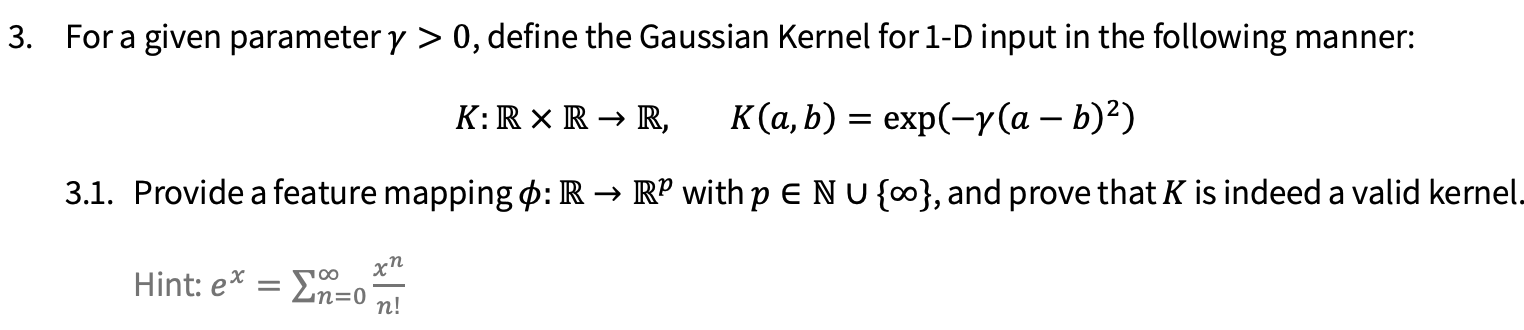
**Answer 2:**

We want to show that

Let ,

Define .

Consider the following:

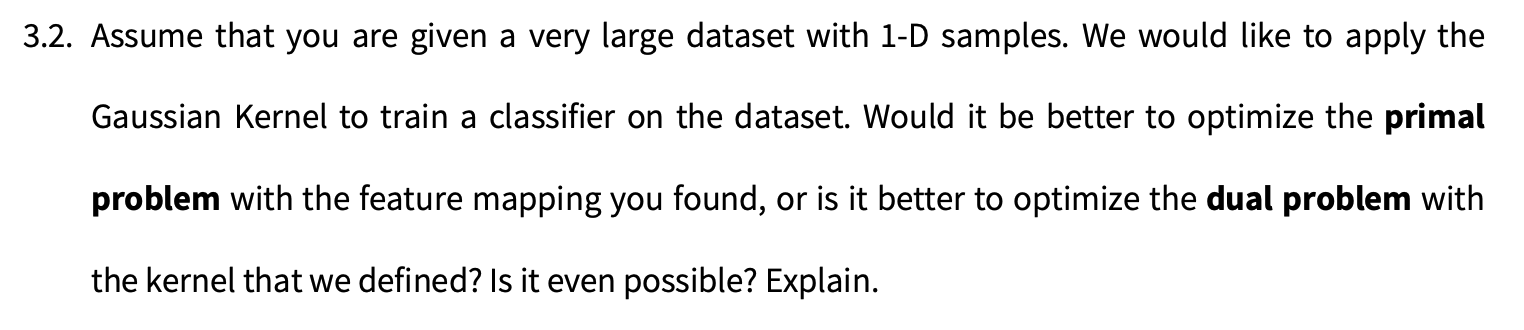


**Answer 3.1:**

We want to find a feature mapping .

Define

Given all together we have:

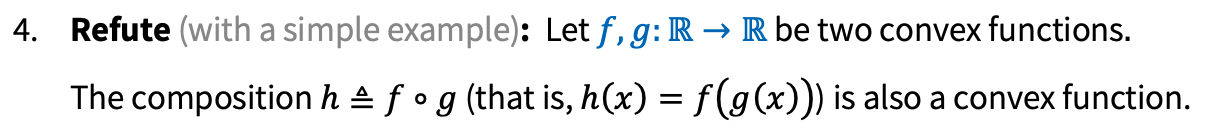


**Answer 3.2:**

For a very large database with 1-D samples, I would prefer to use the dual problem with the Gaussian kernel to train the classifier, mostly for its computational performance.

In the primal problem, after choosing we will apply for all in the dataset and only then use SVM algorithm to learn on the modified data . For any new sample x we will classifie it by applying . Hence, we will need to calculate for any coordinate and store the result for later predictions. If p is large the computation will be long, multiplied by the size of the database will be even longer and storing the massive data could be challenging. If p is infinite this solution might be not possible, we cannot store infinite vector and the runtime will be infinite as well.

On the other hand, in the dual problem, on training time we will go over all pairs in the dataset and store the K(a,b) result in the kernel matrix. In this case each pair calculation is O(1) since we have a close formula for K(a,b) and the memory cost is the size of the dataset squared. Then we could apply Dual SVM and get w which contains that indicated the support vectors. Then for prediction we don’t need to implicitly do inner product with w and sample x rather then we could calculate . But since dual svm optimizes directly over , we expect to have many thus the prediction is computationally faster, each element is O(1) and the sum is at most O(|database|) but we expect it so be faster.



**Answer 4:**

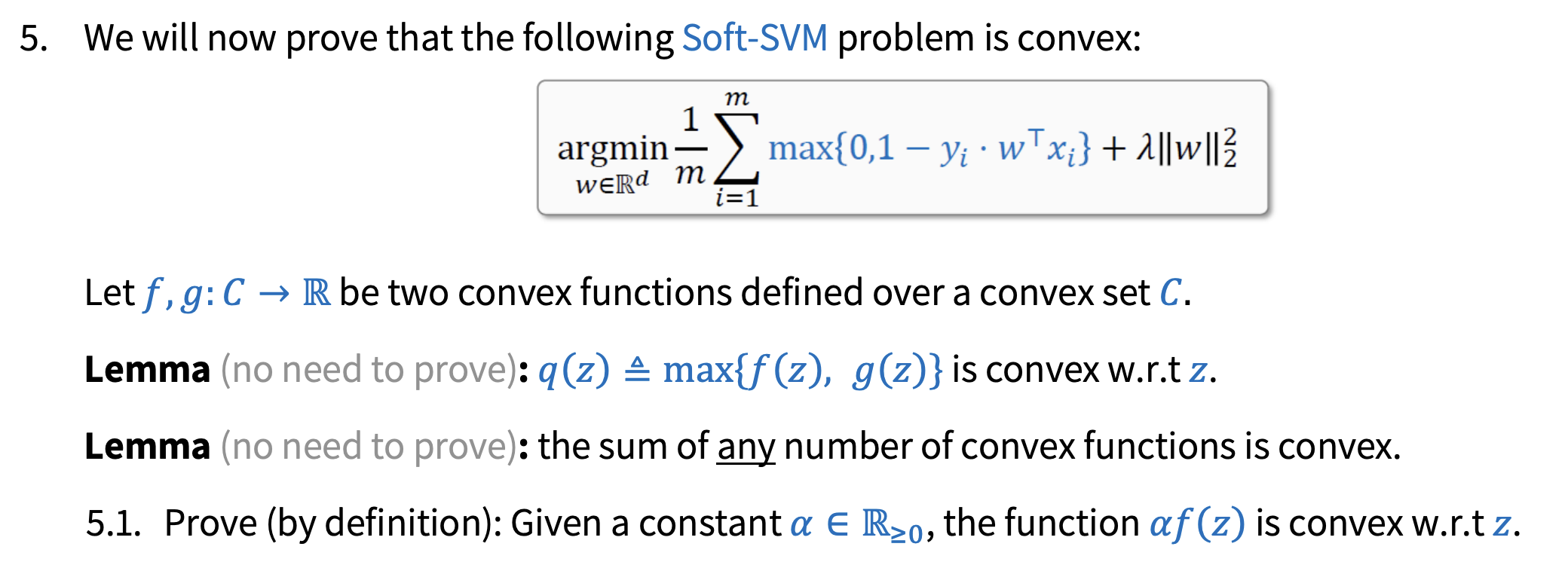
Denote

First, we will show that f(x) and g(x) are convex. Notice that

1. => f(x) is convex
2. => g(x) is convex

Now we will show that h(x) is not convex!

=> h(x) is not convex!

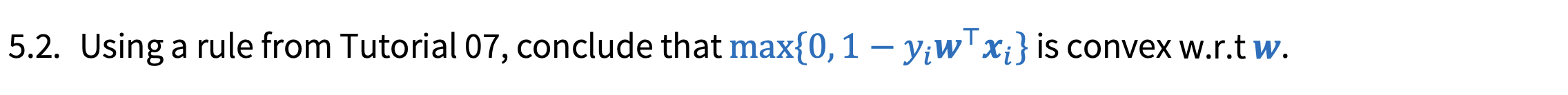


**Answer 5.1:**

Given f(x) is convex by definition:

Then simply by multiplying the inequality by a non-negative scaler we have:

hence, we showed is convex by def.



**Answer 5.2:**

In tutorial 07 we proved that any linear function is convex over a convex set.

First of all, notice that is a convex set (using vectors linearity).

Now, denote .

is trivially convex over a convex set.

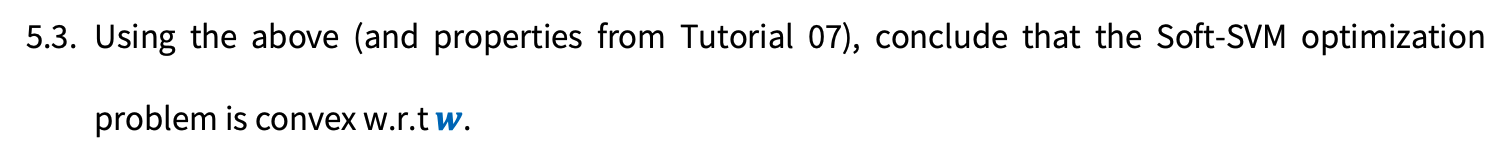
Let , we will show is convex with respect to w.

Thus, for all is convex over a convex set.

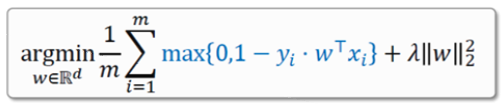
Denote

Using the given first lemma we have that is the maximum of two convex functions, hence is convex itself, over the same convex set.

**MORE ANSWERS ON THE NEXT PAGE**



**Answer 5.3:**



In Tutorial 07, using the 2nd derivate test we showed that is convex w.r.t.w.

Since using answer 5.1 we have is also convex w.r.t.w.

Using the 2nd lemma provided, is also convex as a sum of convex functions, also w.r.t.w.

Since using answer 5.1 we have is also convex w.r.t.w.

Lastly, in Tutorial 07 we showed that argmin of a convex function over a convex set , is convex, w.r.t.w.

It is given from the definition of argmin:



Since is convex over a convex set, it achieves its global minimum. Meaning the set

and of course, if w is in the set, z(w) is smaller\equal to any other z(x) since w is the minimum. We also saw in the tutorial that if a function is convex over a convex set, if it has more then one minimum, then they’re function values are all equal. Any local minimum is also global in that case. Thus, the convexity inequality still holds.

All together we have that Soft-SVM problem is convex w.r.t.w.