Foundation of Data Science and Machine Learning
Tutorial 2

ANISH-111801003
ALISETTI SAI VAMSI - 111801002
AHMED - 111701002
DEVANSH - 111701011
NARTHU - 142002016

1)

Algorithm: Input Transformation (input_transform)

Input: Two linearly separable sets G, B $\subseteq R^n$
Output: Transformed sets G', B' $\subseteq R^{n+1}$

G' $\leftarrow \{(x_1, x_2, ..., x_n, 1) : (x_1, x_2, .., x_n) \in G\}$
B' $\leftarrow \{(x_1, x_2, ..., x_n, 1) : (x_1, x_2, .., x_n) \in B\}$

Algorithm: Output Reconstruction (output_reconstruction)

Input: Normal vector a' of the separable hyperplane in R^{n+1} which passes through the origin
Output: Normal vector a $\in R^n$ and bias b $\in R$ of the separable hyperplane of sets of points G and B in R^n

$a \leftarrow (a'_1, .., a'_n)$
$b \leftarrow -a'_{n+1}$

Algorithm: Perceptron Learning Algorithm (pla)

Input: Two sets G, B which can be separated by a hyperplane passing through the origin
Output: Normal vector a of a hyperplane which separates the two sets, the bias of this hyperplane is 0

Initialize a = 0

Repeat Until no updates to a:

for $x_j \in$ G:

if $<a,x_j> \leq 0$:
a = a + x
for $x_j \in$ B:

if $<a,x_j> \geq 0$:
a = a - x

Algorithm: Modified Perceptron Learning Algorithm (mod_pla)

Input: Two linearly seperable sets G, B $\subseteq R^n$ (which may not be linearly separable from a hyperplane which passes through the origin)
Output: Normal Vector a and bias term b of a hyperplane which passes through the origin

G', B' $\leftarrow$ input_transformation(G, B)
a' $\leftarrow$ pla(G', B')
a, b $\leftarrow$ output_reconstruction(a')

Analysis of Running time of the modified algorithm

Since we are applying the perceptron algorithm to the modified input (each vector appended by the value 1), the running time parameter would become R' and δ', and the perceptron part (pla method call) of the modified algorithm would have number of updates upper bounded by,

$$\#updates \le \frac{4R'^2}{\delta'^2}$$

Now, it's only remaining to show the relation between R', δ' and R, δ.

$$R'^2 = max\,\{|(x_1, x_2, .., x_n, 1)|^2 : (x_1, ..., x_n) \in G \cup B\}$$
$$= max\,\{x_1{}^2 + x_2{}^2 + ... + x_n{}^2 + 1 : (x_1, ..., x_n) \in G \cup B\}$$
$$= max\,\{x_1{}^2 + x_2{}^2 + ... + x_n{}^2 : (x_1, ..., x_n) \in G \cup B\} + 1$$
$$= R^2 + 1$$

We know that δ is the maximum margin of a hyperplane over all separating hyperplanes of G and B.
And δ' is the maximum margin of a hyperplane over all separating hyperplanes of G' and B'.

Let the maximum margin hyperplane for G and B be <p*, x> = b
Now, consider the separating hyperplane for G' and B',
< $(p*_1, ..., p*_n, -b)$, x'> = 0. Clearly this has margin δ.

Hence, $\delta' \ge \delta$. Hence, our running time is bounded by,

$$\#updates \le \frac{4R'^2}{\delta'^2} = \frac{4(R^2 + 1)}{\delta'^2} \le \frac{4(R^2 + 1)}{\delta^2}$$

Hence, we have found an upper bound on the number of updates for the perceptron part of our modified algorithm

2)

1. Show that any constant function c and <x,y> are kernel functions.

   claim1: K(x, y) = <x, y> is a kernel
   proof:

   Define: $\Phi(u) = u$

   Now, clearly we can see that

   $<\Phi(x), \Phi(y)> = <x, y> = K(x, y)$

   Hence $K(x, y) = <x, y>$ is a kernel

   claim1: K(x, y) = c is a kernel (where c > 0)
   proof:

   Define: $\Phi(u) = \sqrt{c}$, this is well define since c is positive

   Now, clearly, we can see that

   $<\Phi(x), \Phi(y)> = <\sqrt{c}, \sqrt{c}> = c = K(x, y)$

   Hence, K(x, y) = c is a kernel function

2. Show that for any kernels K1,K2, K1+K2 is also a kernel and K1K2 is also a kernel.

   claim: If K1(x, y) and K2(x, y) are kernel functions, then K1(x, y) + K2(x, y) is also a kernel function

Proof:
Define: phi(x) = (phi1(x) , phi2(x)), i.e. phi1(x) concatenated with phi2(x)

K1(x,y) = <phi1(x), phi1(y)>
K2(x,y) = <phi2(x), phi2(y)>

K1(x, y) + K2(x, y) = <phi1(x), phi1(y)> + <phi2(x), phi2(y)>
K(x,y) = <phi(x), phi(y)> = <(phi1(x), phi2(x)), (phi1(y), phi2(y))>
= <phi1(x), phi1(y)> + <phi2(x), phi2(y)> = K1(x, y) + K2(x, y)

Hence proved

claim: If K1(x, y) and K2(x, y) are kernel functions, then K1(x, y) * K2(x, y) is also a kernel function

proof:
K(x,y) = <phi(x), phi(y)>
(K1(x, y))(K2(x, y)) = (<phi1(x), phi1(y)>)(<phi2(x), phi2(y)>)

3. Show that the given Kernel K can be generated from 1 and <x, y> using additions and multiplications.

1 is a kernel, <x,y> is a kernel, then using the claim of point 2 (i.e sum of two kernels is also a kernel), we get 1 + <x,y> which is also a kernel.

Since 1+<x,y> is a kernel then again using the same claim as above (i.e product of two kernels is also a kernel), the product is also a kernel, i.e (1 + <x,y>) * (1 + <x,y>) … is also a kernel.

Hence for any degree d, $(1+<x,y>)^d$ is a kernel.