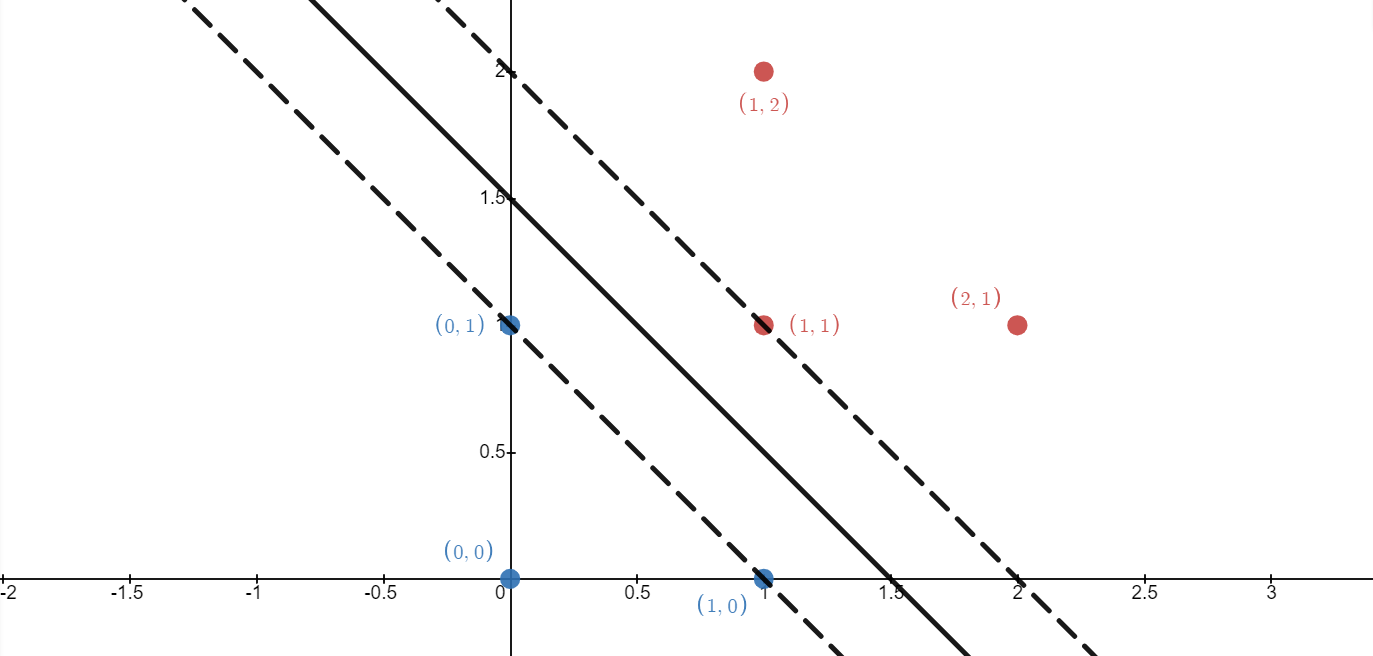
Q1



Margin Boundaries: X1 + X2 – 1 = 1

X1 + X2 – 2 = -1

Optimal Separating Line: X1 + X2 – 1.5 = 0

Support Vectors:

Margin (one side):

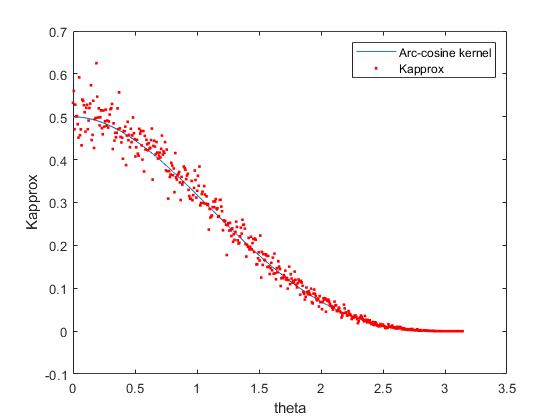
Q2

Q3

The feature space would have 10 dimensions.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | c-a | c-t | a-t | b-a | b-t | c-r | a-r | b-r | r-a | r-t |
| (cat) |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| (car) |  | 0 | 0 | 0 | 0 |  |  | 0 | 0 | 0 |
| (bat) | 0 | 0 |  |  |  | 0 | 0 | 0 | 0 | 0 |
| (bar) | 0 | 0 | 0 |  | 0 | 0 |  |  | 0 | 0 |
| (rat) | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 |  |  |

Q4



I used n = 500 to create this plot.

hw8 = csvread("hw8.csv");

X1 = hw8(:, 1:500);

X2 = hw8(:, 501:1000);

theta = hw8(:, 1001);

Kac = (1/(2\*pi))\*(sin(theta)+(pi-theta).\*cos(theta));

plot(theta, Kac, '-');

hold on;

W = randn(500, 500);

KapproxVector = zeros(500, 1);

for i = 1:500

x1 = X1(i, :)';

x2 = X2(i, :)';

sum = 0;

for j = 1:500

w = W(j, :);

sum = sum + max(0, dot(w, x1)) \* max(0, dot(w, x2));

end

Kapprox = sum / 500;

KapproxVector(i, 1) = Kapprox;

end

scatter(theta, KapproxVector, '.', 'r');

hold off;

xlabel('theta');

ylabel('Kapprox');

legend('Arc-cosine kernel', 'Kapprox');

Q5

This equation describes neural networks, each hidden layer has many neurons, Relu activation function is used within the model, this function helps to make the model parameters sparse.