

Übungsblatt 6

„Mustererkennung“

J. Cavojska, N. Lehmann, R. Toudic

16.06.2015

Inhaltsverzeichnis

1	Perzeptron Lernalgorithmus	2
2	Schwellwerte	8
2.1	Aufgabe 2A	8
2.2	Aufgabe 2B	10

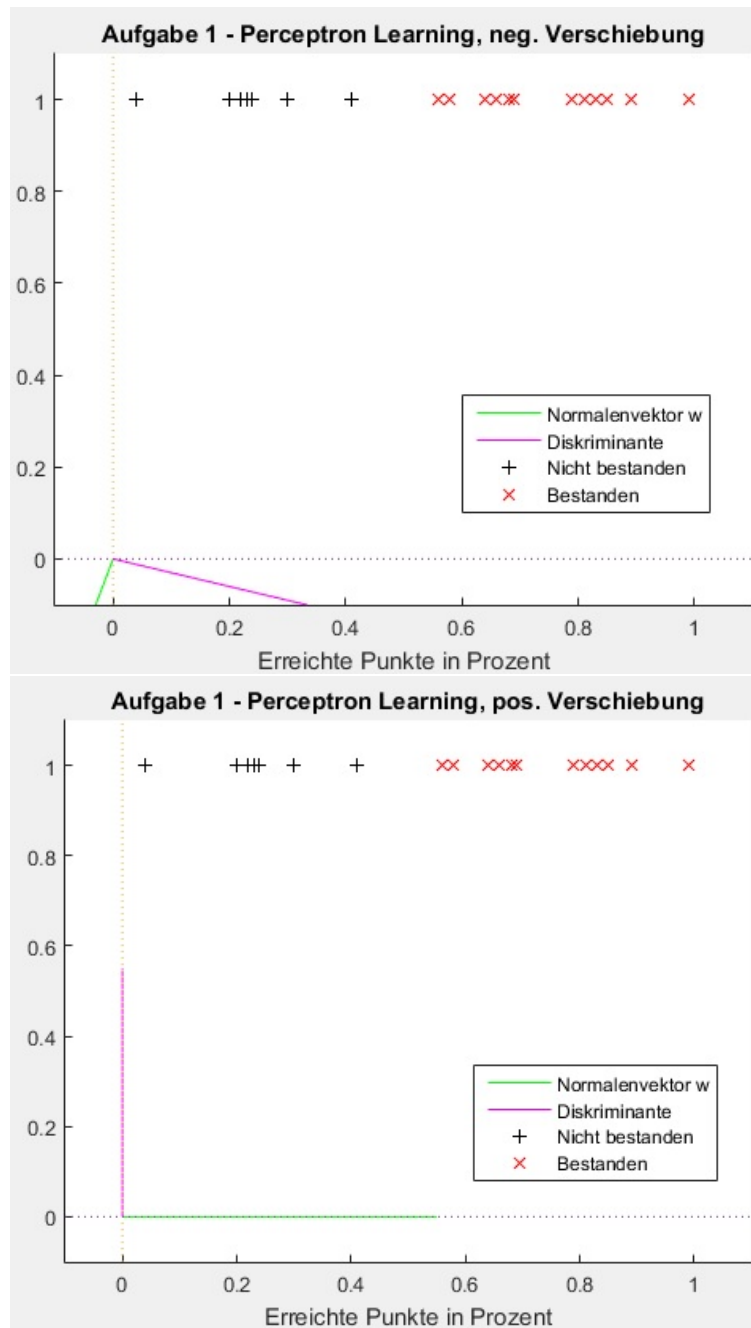
1 Perzeptron Lernalgorithmus

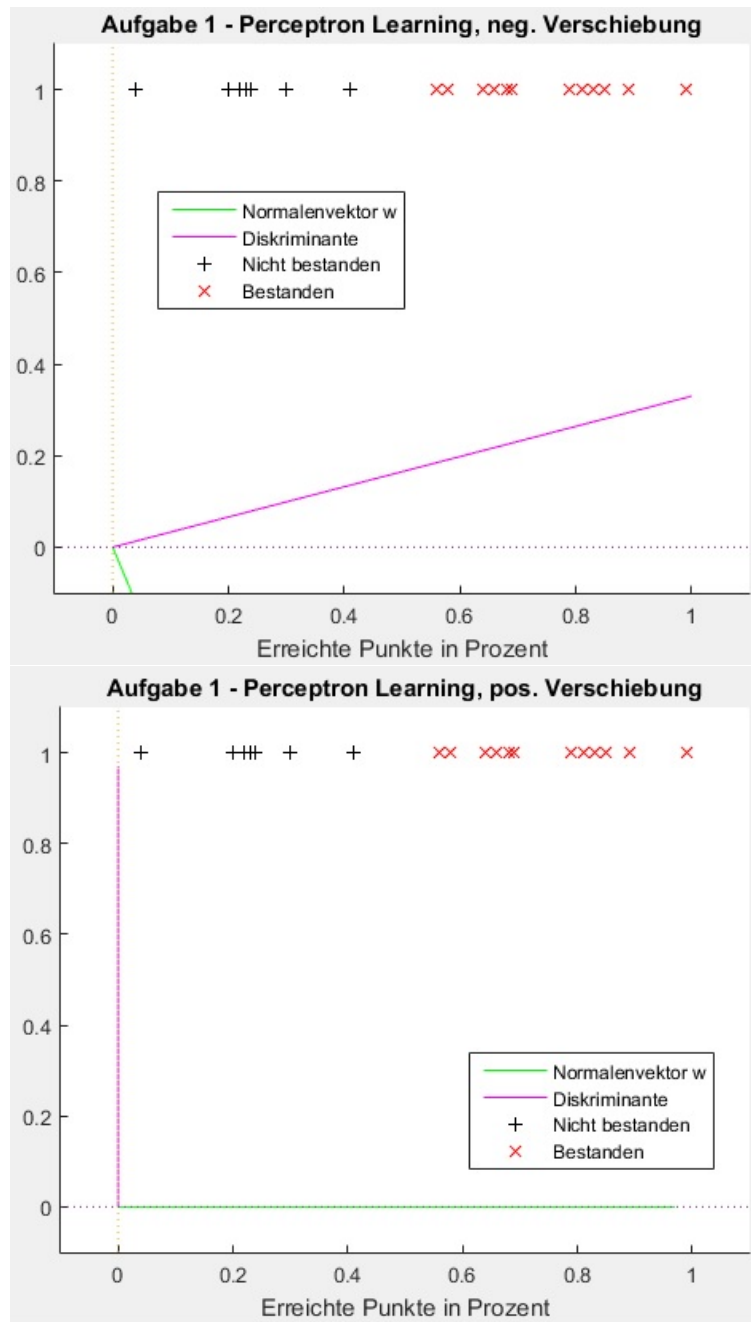
```
1 % Clean up
2 clear all
3 close all
4 clc
5
6 % Datenaufbereitung
7 Data = load('klausur.txt');
8 Punkte = Data(:,1);
9 Features = horzcat(Punkte, ones(size(Punkte,1), 1));
10 Noten = Data(:,2);
11 Punkte0 = Data((Data(:,2)==0),:);
12 Punkte1 = Data((Data(:,2)==1),:);
13 x1 = linspace(0,1);
14 x2 = linspace(-5,5);
15
16
17 w = [0 0]; % random choosen vector w
18 limit = size(Data, 1); % number of iterations | Abbruchkriterium
19 for i = 1:limit
20     if w(1) == 0 && w(2) == 0
21         w_norm = [0 0];
22     else
23         w_norm = w / norm(w);
24     end
25     lineNum = mod(i, size(Features,1))+1;
26     proj = Features(lineNum, :) * w_norm'; % scalar projection
27
28     if Noten(lineNum) == 1
29         if proj < 0 % wrong classification
30
31             Features(lineNum, :);
32             w = w + Features(lineNum, :);
33             w_norm = w / norm(w);
34             w_x = w(1) * x1;
35
36             if w(2) == 0
37                 w_y = w_x * 0;
38             else
39                 coeff_w = w(2) / w(1)
40                 w_y = w_x * coeff_w;
41             end
42
43             diskriminante = [-w(2) w(1)];
44             diskriminante_x = diskriminante(1) * x1;
45
46             if diskriminante(1) == 0
47                 diskriminante_y = linspace(0, diskriminante(2));
48             else
49                 coeff_d = diskriminante(2) / diskriminante(1);
50                 diskriminante_y = diskriminante_x * coeff_d;
51             end
52
```

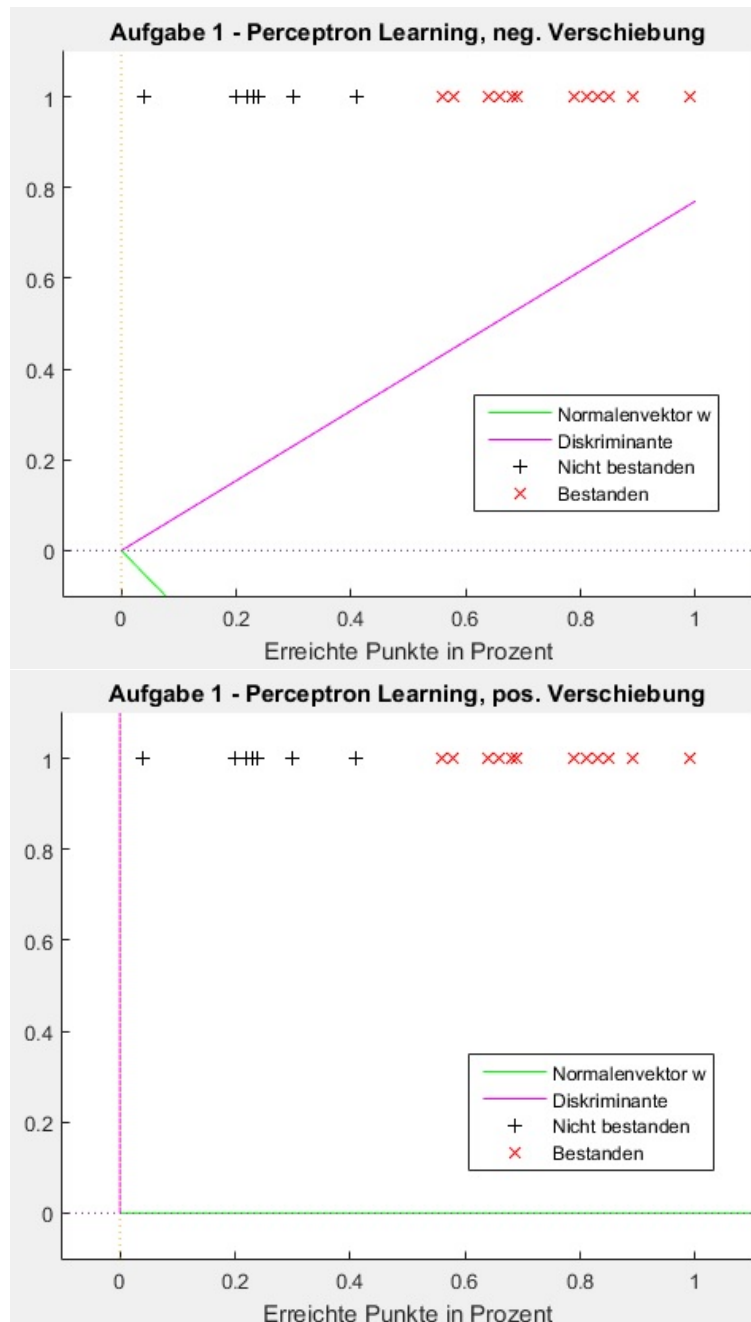
```

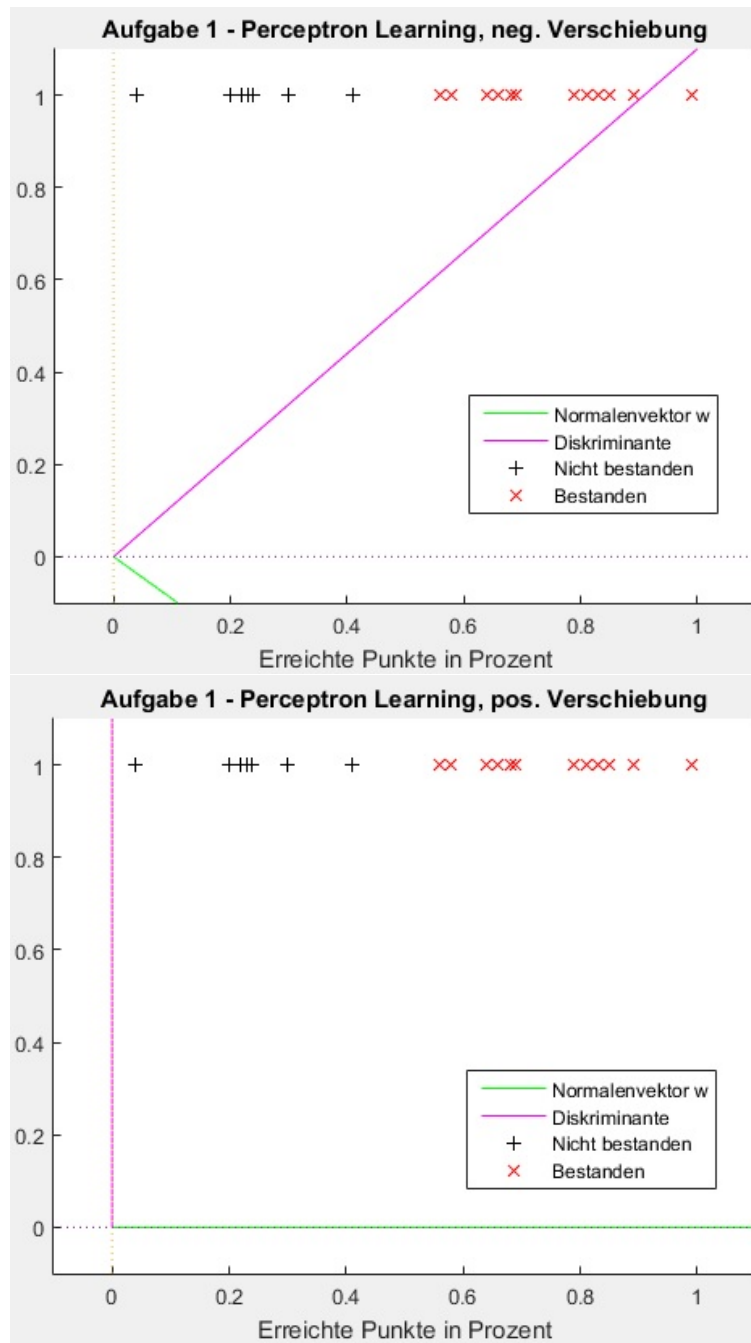
53     % plot
54     figure('NumberTitle','off','Name','Aufgabe 1 – Perceptron ←
55         Learning');
56     hold on
57
58     plot(w_x, w_y, 'g');
59     plot(diskriminante_x, diskriminante_y, 'm');
60     gscatter(Punkte, ones(size(Punkte,1), 1), Noten, 'krb', '+x', [], '←
61         off');
62
63     title('Aufgabe 1 – Perceptron Learning, pos. Verschiebung');
64     xlabel('Erreichte Punkte in Prozent');
65     axis([-0.1 1.1 -0.1 1.1]);
66     legend('Normalenvektor w', 'Diskriminante', 'Nicht bestanden', '←
67         Bestanden');
68     xL = xlim;
69     yL = ylim;
70     plot([0 0], yL, ':');
71     plot(xL, [0 0], ':');
72 end
73
74 if Noten(lineNum) == 0
75     if proj >= 0 % wrong classification
76
77         w = w - Features(lineNum,:);
78         w_norm = w / norm(w);
79         coeff_w = w(2) / w(1);
80         w_x = w(1) * x1;
81         w_y = w_x * coeff_w;
82         diskriminante = [-w(2) w(1)];
83         coeff_d = diskriminante(2) / diskriminante(1);
84         diskriminante_x = diskriminante(1) * x1;
85         diskriminante_y = diskriminante_x * coeff_d;
86
87     % plot
88     figure('NumberTitle','off','Name','Aufgabe 1 – Perceptron ←
89         Learning');
90     hold on
91
92     plot(w_x, w_y, 'g');
93     plot(diskriminante_x, diskriminante_y, 'm');
94     gscatter(Punkte, ones(size(Punkte,1), 1), Noten, 'krb', '+x', [], '←
95         off');
96
97     title('Aufgabe 1 – Perceptron Learning, neg. Verschiebung');
98     xlabel('Erreichte Punkte in Prozent');
99     axis([-0.1 1.1 -0.1 1.1]);
100    legend('Normalenvektor w', 'Diskriminante', 'Nicht bestanden', '←
101        Bestanden');
102    xL = xlim;
103    yL = ylim;
104    plot([0 0], yL, ':');
105    plot(xL, [0 0], ':');
106 end
107 end
108 end

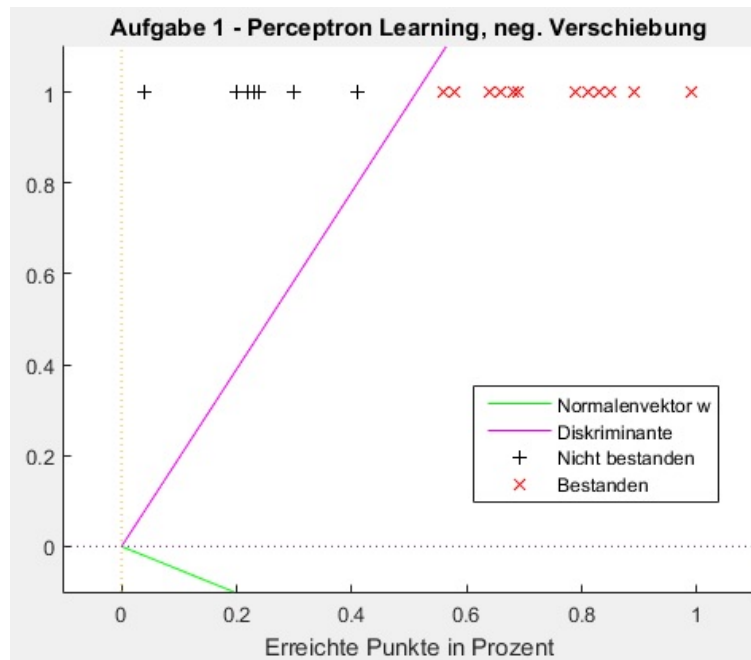
```











2 Schwellwerte

2.1 Aufgabe 2A

```

1 schwellwerte = [];
2 for iter = 1:100
3     randOrder = randperm(size(Features, 1));
4     randFeatures = Features(randOrder');
5     w = [max(Punkte) max(Noten)]; % random chosen vector w
6     t = 0;
7     limit = size(Data, 1); % number of iterations
8     for i = 1:limit
9         w_norm = w / norm(w);
10        lineNum = mod(i, size(randFeatures,1))+1;
11        proj = randFeatures(lineNum, :) * w_norm'; % scalar projection
12        if Noten(lineNum) == 1
13            if proj < 0
14                t = t + 1;
15                w = w + randFeatures(lineNum, :);
16                diskriminante = [-w(2) w(1)];
17            end
18        end
19        if Noten(lineNum) == 0
20            if proj >= 0 % wrong classification
21                t = t + 1;

```



```

22         w = w - randFeatures(lineNum, :);
23         diskriminante = [-w(2) w(1)];
24     end
25 end
26 end
27     schwellwerte = vertcat(schwellwerte, w);
28 end
29
30 % output
31 schwellwerte
32 mean_schwellwert = mean(schwellwerte)
33 % mean_schwellwert = [-0.1980 -0.1880]

```

2.2 Aufgabe 2B

```
1 figure('NumberTitle','off','Name','Aufgabe 2 – Lin. Regression');
2
3 % calculate
4 onesVector = ones(size(Data,1), 1);
5 X = horzcat(onesVector, Punkte);
6 beta = inv(X'*X) * X' * Noten;
7 fx = beta(1) + beta(2)*x2;
8 pkt = (0.5-beta(1))/beta(2);
9 % pkt = 0.4804
10
11 % plot
12 hold on
13 scatter(Punkte, Noten, 'x', 'b')
14 plot(x2,fx, 'g')
15 scatter(pkt,0.5, 'o', 'r')
16
17 axis([-0.1 1.1 -0.1 1.1]);
18 legend('Noten', 'Diskriminante', 'Schwellenwert');
```

