CS 559: Homework Set 3

Alana Laryssa S A Santos

11/5/2015

1 First problem

$$Y = \begin{bmatrix} 1 & -2 & 1 \\ 1 & -5 & -4 \\ 1 & -3 & 1 \\ 1 & 0 & -3 \\ 1 & -8 & -1 \\ -1 & 2 & 5 \\ -1 & 1 & 0 \\ -1 & 5 & -1 \\ -1 & -1 & -3 \\ -1 & 6 & 1 \end{bmatrix}$$

2 Second problem

The margin is given by $m = 2/||w||$ and we want the distance to the closest samples to be large. In order to do that, we have to minimize $||w||$. Not mathematically, no. But we can minimize $\frac{1}{2}||w||^2$ instead.

3 First problem

3.1

$$mean(dat) = \begin{bmatrix} -0.0012 & -0.0010 & -0.0021 \end{bmatrix}^T$$
$$mean(dat1) = \begin{bmatrix} 0.9988 & 1.9990 & 2.9979 \end{bmatrix}^T$$
$$mean(dat2) = \begin{bmatrix} 9.9877 & 5.9969 & 2.9979 \end{bmatrix}^T$$
$$mean(dat3) = \begin{bmatrix} 11.3290 & 3.5794 & 4.0828 \end{bmatrix}^T$$

$$cov(dat) = \begin{bmatrix} 0.9978 & -0.0003 & 0.0016 \\ -0.0003 & 1.0016 & 0.0014 \\ 0.0016 & 0.0014 & 0.9994 \end{bmatrix}$$
$$cov(dat1) = \begin{bmatrix} 0.9978 & -0.0003 & 0.0016 \\ -0.0003 & 1.0016 & 0.0014 \\ 0.0016 & 0.0014 & 0.9994 \end{bmatrix}$$
$$cov(dat2) = \begin{bmatrix} 99.7804 & -0.0079 & 0.0159 \\ -0.0079 & 9.0148 & 0.0041 \\ 0.0159 & 0.0041 & 0.9994 \end{bmatrix}$$

$$cov(dat3) = \begin{bmatrix} 49.1116 & 44.5992 & 7.0450 \\ 44.5992 & 58.6218 & 7.6648 \\ 7.0450 & 7.6648 & 2.0661 \end{bmatrix}$$

3.2

The direction with maximum variance for $dat2$ is $a = \begin{bmatrix} 1.000 & -0.0001 & 0.0002 \end{bmatrix}^T$ while for $dat3$ is $b = \begin{bmatrix} 0.6649 & 0.7394 & 0.1059 \end{bmatrix}^T$. For obtaining $dat3$, we multiplied $dat2$ by R. So, as expected, the direction with maximum variance for $dat3$, b, is close to $R * a$.

3.3

As I said above, yes. I expected the first principal component of $dat3$ to be approximate to the first principal component of $dat2$ multiplied by R.

3.4

The expected mean for $dat1$ is $mean(dat) + \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T$. And the expected mean for $dat2$ is $diag(\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}) * mean(dat1)$. For $dat3$, the mean is supposed to be equals to R multiplied by $mean(dat2)$.

As for the covariances matrices for dat and $dat1$, they are expected to be equal, since a translation in the data does not change how much two variables relate.

$Dat2$ was generated by scaling $dat1$ by $s = \begin{bmatrix} 10 & 3 & 1 \end{bmatrix}^T$. So, the covariance matrix for $dat2$ should be 3x3 with elements $b_{ij} = s_i * s_j * a_{ij}$, where s_i is the $i-th$ element of vector s and a_{ij} are the elements of the covariance matrix of $dat1$. Then, for $dat3 = R * dat2$, we have $cov(dat3) = R * cov(dat2) * R^T$.

The first principal component of $dat3$ is close to R multiplied by the one of $dat2$. So, the more samples we have, the less is the error between this estimation and the real value. For example:

- for 100 samples, the first component for $dat3$ is $v3 = \begin{bmatrix} 0.6510 & 0.7498 & 0.1183 \end{bmatrix}^T$ while $v2 = \begin{bmatrix} 0.6529 & 0.7520 & 0.1162 \end{bmatrix}^T$ is R multiplied by the first component for $dat2$. Using SAD, the distance between these vectors is 0.0062.

- for 1000 samples, $v3 = \begin{bmatrix} 0.6716 & 0.7323 & 0.1125 \end{bmatrix}^T$, $v2 = \begin{bmatrix} 0.6727 & 0.7336 & 0.1107 \end{bmatrix}^T$ and the error is 0.0041.

- for 10000*100 samples, $v3 = \begin{bmatrix} 0.6644 & 0.7398 & 0.1058 \end{bmatrix}^T$, $v2 = \begin{bmatrix} 0.6645 & 0.7400 & 0.1038 \end{bmatrix}^T$ and the error is 0.0023.

So, the more samples we have to observe, the closer is the real value to our estimation.

4 Second problem

For 15 runs, the mean of accuracy was 74,5%.

Selected principal components for one run: $princomp = \begin{bmatrix} -0.0013 & 0.0201 & 0.0244 \\ 0.0857 & 0.9650 & -0.1814 \\ 0.0153 & 0.1836 & 0.8590 \\ 0.0525 & -0.0481 & 0.4153 \\ 0.9948 & -0.0840 & -0.0212 \\ 0.0115 & 0.0493 & 0.1491 \\ 0.0003 & 0.0012 & 0.0006 \\ -0.0001 & 0.1515 & 0.1828 \end{bmatrix}$

Where the colunms of the matrix are the vectors of the new basis.

5 Third problem

```
function [y1,y2] = hw3_fisher(c1,c2)
    % compute the mean for each class
    mu1 = mean(c1);
    mu2 = mean(c2);

    % compute scatter matrices S1 and S2 for each class
    d1 = c1 - repmat(mu1,length(c1),1);
    d2 = c2 - repmat(mu2,length(c2),1);
    s1 = d1'*d1;
    s2 = d2'*d2;

    % within class scatter Sw
    sw = s1+s2;

    % get the optimal line direction
    v = inv(sw)*(mu1-mu2)';

    y1 = v'*c1';
    y2 = v'*c2';
end
```

All points were classified correctly, except for the point $\begin{bmatrix} -1 & -3 \end{bmatrix}$ in the second class. The optimal line direction is $\begin{bmatrix} -0.0862 & -0.0129 \end{bmatrix}^T$.

6 Fourth problem

The average accuracy in 15 runs was 76.77%. And one optimal projection was $\begin{bmatrix} -0.0004 & -0.0001 & 0.0000 & -0.0000 & 0.0000 & -0.0002 & -0.0018 & -0.0000 \end{bmatrix}^T$.

7 Fifth problem

7.1 Part a

Before normalization:

$$\begin{cases} a^T \begin{bmatrix} 0 & 0 & 1 & 2 & 0 \\ 4 & 0 & 1 & 2 & 1 \\ -1 & 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & 1 & 0 \end{bmatrix} > 0 \\ a^T \begin{bmatrix} 1 & 1 & -1 & 0 & 2 \\ -1 & -1 & 1 & 1 & 0 \\ -1 & 1 & 1 & 2 & 1 \end{bmatrix} < 0 \end{cases}$$

After normalization:

$$a^T \begin{bmatrix} -1 & -1 & -1 & 1 & 0 & -2 \\ 1 & 0 & 0 & 1 & 2 & 0 \\ -1 & 1 & 1 & -1 & -1 & 0 \\ 1 & 4 & 0 & 1 & 2 & 1 \\ 1 & -1 & 1 & 1 & 1 & 0 \\ 1 & -1 & -1 & -1 & 1 & 0 \\ -1 & 1 & -1 & -1 & -2 & -1 \end{bmatrix} > 0$$

7.2 Part b

We have the normalized input to perceptron: $data = \begin{bmatrix} -1 & -1 & -1 & 1 & 0 & -2 \\ 1 & 0 & 0 & 1 & 2 & 0 \\ -1 & 1 & 1 & -1 & -1 & 0 \\ 1 & 4 & 0 & 1 & 2 & 1 \\ 1 & -1 & 1 & 1 & 1 & 0 \\ 1 & -1 & -1 & -1 & 1 & 0 \\ -1 & 1 & -1 & -1 & -2 & -1 \end{bmatrix}$
and initial weight vector $a_0 = \begin{bmatrix} 3 & 1 & 1 & -1 & 2 & -7 \end{bmatrix}$.

$a_0 * data_0 = 3*(-1)+1*(-1)+1*(-1)+(-1)*1+2*0+(-7)*(-2) = 8$
Since we don't have misclassification, we move on
$a_0 * data_1 = 3*1+1*0+1*0+(-1)*1+2*2+(-7)*0 = 6$
$a_0 * data_2 = 3*(-1)+1*1+1*1+(-1)*(-1)+2*(-1)+(-7)*0 = -2$
Now we have misclassification, let's update a.
$a_1 = a_0 + data_2 = \begin{bmatrix} 3 & 1 & 1 & -1 & 2 & -7 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 1 & -1 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 & -2 & 1 & -7 \end{bmatrix}$
We're ready to move on again
$a_1 * data_3 = 2*1+2*4+2*0+(-2)*1+1*2+(-7)*1 = 3$ $a_1 * data_4 = 2*1+2*(-1)+2*1+(-2)*1+1*1+(-7)*0 = 1$ $a_1 * data_5 = 2*1+2*(-1)+2*(-1)+(-2)*1+1*1+(-7)*0 = 1$ $a_1 * data_6 = 2*(-1)+2*1+2*(-1)+(-2)*(-1)+1*(-2)+(-7)*(-1) = 5$ Now we visit the data again with the current weight vector
$a_1 * data_0 = 6$

$a_1 * data_1 = 2$
$a_1 * data_2 = 3$
$a_1 * data_3 = 3$
$a_1 * data_0 = 1$
$a_1 * data_5 = 1$
$a_1 * data_6 = 5$
We don't have anymore misclassification, so our final weight vector is $a1 = \begin{bmatrix} 2 & 2 & 2 & -2 & 1 & -7 \end{bmatrix}$. Converting back to the original features before normalizing, the discriminant function is: $g(x) = 2{*}x_1 + 2{*}x_2 - 2{*}x_3 + x_4 - 7{*}x_5 + 2$

7.3 Matlab code

```
data = [1 1 -1 0 2;
        0 0 1 2 0;
        -1 -1 1 1 0;
        4 0 1 2 1;
        -1 1 1 1 0;
        -1 -1 -1 1 0;
        -1 1 1 2 1];
label = [2 1 2 1 1 1 2]';

data = [ones(length(data),1) data];

lbl = logical(label - 1);
data(lbl, :) = - data(lbl,:);

a = [3 1 1 -1 2 -7];
miss = Inf;
while miss > 0
    miss = 0;
    for i = 1:size(data,1)
        y = data(i,:)*a'
        if y < 0
            a = a + data(i,:)
            miss = miss + 1;
        end
    end
end
```