Image Analysis Assignment 2

Ruqayyah Nabage

September 2023

1 Filtering

- Image A was filtered using f_2. This is because when a filter is multiplied by a fraction the effect is a blurred image.
- Image B was filtered using f_1. This is because when f_1 is slid across an image it gives high values for edges horizontally and vertically.
- Image C was filtered using f_3. This is because of the high value (5) at the center of f_3, this gives a sharpening effect to the image.
- Image D was filtered using f_5, because a filter like f_5 would give the edges horizontally (across the rows) and that is why it fainter than image B.
- Image E was filtered using f_4 because the filter returned the image as it is without any changes.

2 Interpolation

a) Linear interpolation involves filling in the gaps of an image by assuming a straight line relationship between different points in the image and making connections based on this linear relationship.

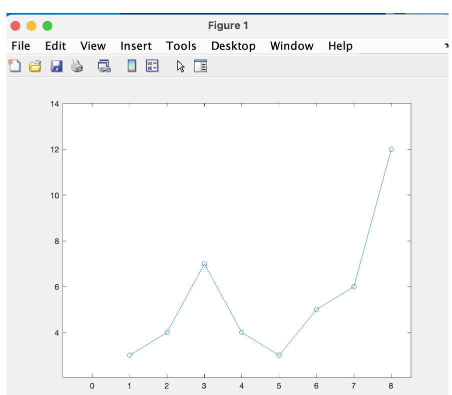


Figure1: Linear interpolation of $F_{lin}(x)$

$F_{lin}(x)$ is continuous. It is differentiable at some points.

b)

$$g(x) = \begin{cases} 1 - x & \text{if } |x| > 0 \\ 0 & \text{otherwise} \end{cases}$$

c)

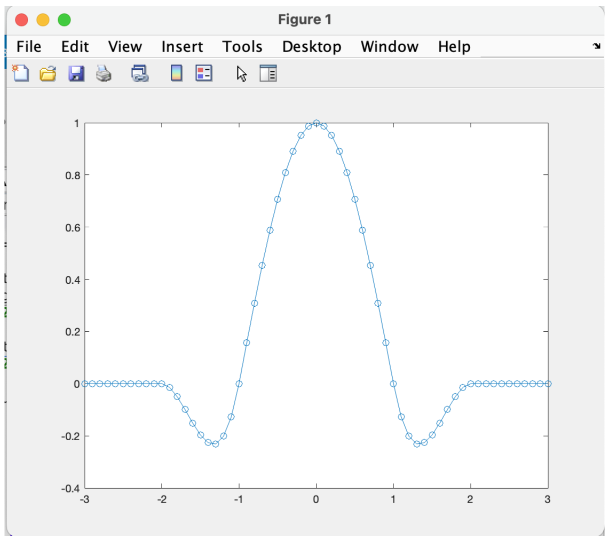


Figure 2: $g(x), -3 \leq x \leq 3$

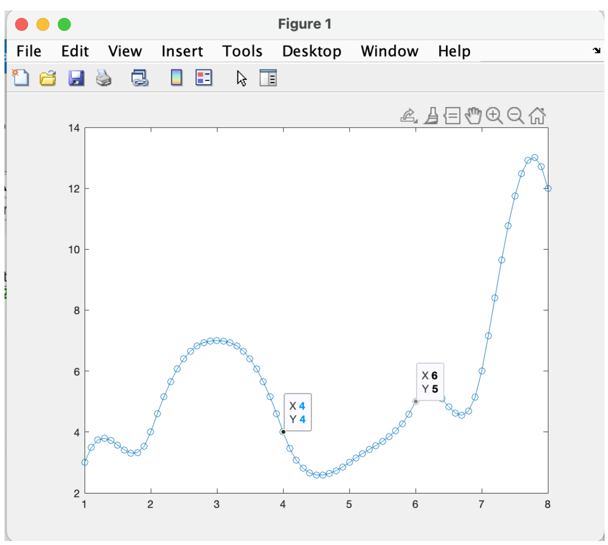


Figure 3: $F_g(x)$, using g(x) for $1 \leq x \leq 8$

Yes, $F_g(x)$ is continuous and it is differentiable. It is differentiable because its properties are gotten from the function $g(x)$ which is fully differentiable. Also, another requirement for differentiability is that the function is continuous at the point where the transition between the pieces occur (i.e at the points that are the values of i, there are no sharp breaks) as indicated by the two example points in the picture above.

3 Classification using Nearest Neighbour and Bayes theorem

3.1 Nearest Neighbours

For the nearest neighbour classification, 8 measurements were classified correctly and only one measurement was misclassified. '0.4243' belongs in Class 3 and it was misclassified to Class 1. For my solution; I stored the first four measurements of every class in an array for training data and I stored the test values in a different array. I performed the calculation with a for loop; finding the distance between the test value and every value in the training set$min|x_{test} - x_{training}|$. The class of the measurement which gives the lowest distance is set as the class of the test value. For example: For test measurement 0.4010:

$$|0.4010-0.4003| = 0.0007 \quad |0.4010-0.3988| = 0.0022 \quad |0.4010-0.3998| = 0.0012$$

$$|0.4010-0.3997| = 0.0013 \quad |0.4010-0.2554| = 0.1456 \quad |0.4010-0.3139| = 0.0871$$

$$|0.4010-0.2627| = 0.1383 \quad |0.4010-0.3802| = 0.0208 \quad |0.4010-0.5632| = 0.1622$$

$$|0.4010-0.7687| = 0.3677 \quad |0.4010-0.0524| = 0.3486 \quad |0.4010-0.7586| = 0.3576$$

$$min|x_{test} - x_{training}| \text{ is } |0.4010 - 0.4003| = 0.0007.$$

Therefore, 0.4003 belongs in Class 1, so test measurement 0.4010 is classified to be in Class 1.

3.2 Gaussian Distributions

For the Gaussian distribution, in all the 7 samples of the 3 classes (a total of 21 measurements), 19 measurements were classified correctly and two measurements were misclassified. '0.3802' belongs in Class 2 and it was classified to Class 1. '0.4243' belongs in Class 3 and it was classified to Class 1. For my solution I used Bayes Theorem.

$$P(y = j|x) = \frac{P(x|y = j)P(y = j)}{P(x)}$$

Using the given means and standard deviations I calculated the likelihood of the three classes $P(x|y = j)$ where j = 1, 2 and 3 using the normpdf function.

And since all three classes are likely to occur we have a prior of $\frac{1}{3}$ for $P(y=j)$ where j = 1, 2 and 3 and

$$P(x) = P(x|y=1)P(y=1) + P(x|y=2)P(y=2) + P(x|y=3)P(y=3)$$

. Each measurement was then classified into the class for which it had the highest a posteriori. The maximum a posteriori classification of a measured x was 1. For example: For test measurement 0.6769:

$$P(y=3|x) = \frac{P(x|y=3)P(y=3)}{P(x)}$$

$$= \frac{normpdf(0.6769, 0.55, 0.2) \times \frac{1}{3}}{P(x|y=1)P(y=1) + P(x|y=2)P(y=2) + P(x|y=3)P(y=3)}$$

$$= \frac{1.6310 \times \frac{1}{3}}{0.5437} = 1$$

4 Image Classification

Considering the x image $\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}$. Our possible cases are

$$y_1 = \begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix} \text{with a priori probability } P(y_1) = \frac{1}{4} = 0.25$$

$$y_2 = \begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix} \text{with a priori probability } P(y_2) = \frac{1}{2} = 0.5$$

$$y_3 = \begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix} \text{with a priori probability } P(y_3) = \frac{1}{4} = 0.25$$

. To calculate the likelihood that the picture x distorted by noise is either y_1 , y_2 or y_3 we compare the pixels of x with the pixels of y_1, y_2 or y_3 and for each calculate $P(x|y_1)$, $P(x|y_2)$ and $P(x|y_3)$. To get this the product of the probabilities that each pixel was observed correctly or not is obtained. For $\epsilon = 10\%$, the probability that a pixel was not observed correctly is 0.1 and the probability that a pixel was observed correctly is 1 - 0.1 = 0.9. and to get $P(x|y_1)$ we compare the pixels in x with the pixels in y_1 and find the product of the probabilities of whether they were observed correctly or not.

$$P(x|y_1) = 0.9 \times 0.9 \times 0.9 \times 0.1 = 0.0729$$

$$P(x|y_2) = (0.1)^3 \times 0.9 = 9 \times 10^{-4} \qquad P(x|y_3) = (0.9)^2 \times (0.1)^2 = 0.0729$$

Using Bayes' theorem, the a posteriori is given by $P(y_j|x) = \frac{P(x|y_j)P(y_j)}{P(x)}$

$$P(x) = P(x|y_1)P(y_1) + P(x|y_2)P(y_2) + P(x|y_3)P(y_3)$$

$$= (0.0729 \times 0.25) + (0.0009 \times 0.5) + (0.0081 \times 0.25) = 0.0207$$

$$P(y_1|x) = \frac{P(x|y_1)P(y_1)}{P(x)} = \frac{0.0729 \times 0.25}{0.0207} = 0.8804 = 88.04\%$$

$$P(y_2|x) = \frac{P(x|y_2)P(y_2)}{P(x)} = \frac{0.0009 \times 0.5}{0.0207} = 0.0217 = 2.17\%$$

$$P(y_3|x) = \frac{P(x|y_3)P(y_3)}{P(x)} = \frac{0.0081 \times 0.25}{0.0207} = 0.0978 = 9.78\%$$

The maximum a posteriori is $P(y_1|x)$ with 88.04% probability.

When ϵ = 40%, the probability that a pixel was not observed correctly is 0.4 and the probability that a pixel was observed correctly is 1 - 0.4 = 0.6. The likelihoods are calculated in the same way as above when = 10%.

$$P(x|y_1) = 0.6 \times 0.6 \times 0.6 \times 0.4 = 0.0864$$

$$P(x|y_2) = (0.4)^3 \times 0.6 = 0.0384 \qquad P(x|y_3) = (0.6)^2 \times (0.4)^2 = 0.0576$$

$$P(x) = P(x|y_1)P(y_1) + P(x|y_2)P(y_2) + P(x|y_3)P(y_3)$$

$$= (0.0864 \times 0.25) + (0.0384 \times 0.5) + (0.0.0576 \times 0.25) = 0.0552$$

$$P(y_1|x) = \frac{P(x|y_1)P(y_1)}{P(x)} = \frac{0.0864 \times 0.25}{0.0552} = 0.3913 = 39.13\%$$

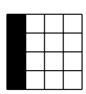
$$P(y_2|x) = \frac{P(x|y_2)P(y_2)}{P(x)} = \frac{0.0009 \times 0.5}{0.0552} = 0.3478 = 34.78\%$$

$$P(y_3|x) = \frac{P(x|y_3)P(y_3)}{P(x)} = \frac{0.0081 \times 0.25}{0.0552} = 0.2609 = 26.09\%$$

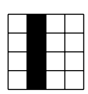
The maximum a posteriori is $P(y_1|x)$ with 39.13% probability.

5 Line Classification

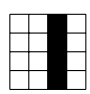
The a priori of the four images are as follows.



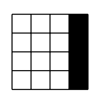
(a) P(L = 1)



(b) P(L = 2)



(c) P(L = 3)



(d) P(L = 4)

The probability that the line is in column 1 is $P(L = 1) = 0.3$.
The probability that the line is in column 2 is $P(L = 2) = 0.2$.
The probability that the line is in column 3 is $P(L = 3) = 0.2$.
The probability that the line is in column 1 is $P(L = 4) = 0.3$.

Considering our given image as x, and $\epsilon = 0.2$. The probability of a pixel that is not observed correctly is 0.2 and the probability of a pixel that is observed correctly is $1 - 0.2 = 0.8$.

Calculating the likelihood for $P(x|L=1)$; there is 1 black pixel in the correct place, 3 black pixels in the wrong place, 9 white pixels in the correct place and 3 white pixels in the wrong place. A total of 6 pixels in the wrong place and 10 pixels in the right place.

$$P(x|L=1) = (0.2)^6(0.8)^{10} = 6.8719 \times 10^{-6}$$

Calculating the likelihood for $P(x|L=2)$; there are 2 black pixels in the correct place, 2 black pixels in the wrong place, 10 white pixels in the correct place and 2 white pixels in the wrong place. A total of 4 pixels in the wrong place and 12 pixels in the right place.

$$P(x|L=2) = (0.2)^4(0.8)^{12} = 1.0995 \times 10^{-4}$$

Calculating the likelihood for $P(x|L=3)$; Since errors in different pixels are independent, and they have the same number of pixels in the right and wrong places, the likelihood of $P(x|L=1) = P(x|L=3) = 6.8719 \times 10^{-6}$

Calculating the likelihood for $P(x|L=4)$; there are no pixels in the right place.

$$P(x|L=4) = (0.2)^{16} = 6.5536 \times 10^{-12}$$

$$P(x) = P(x|L=1)P(L=1)+P(x|L=2)P(L=2)+P(x|L=3)P(L=3)+P(x|L=4)P(L=4)$$

$$P(L=1|x) = \frac{P(x|L=1)P(L=1)}{P(x)} = 8.11\% \quad P(L=2|x) = \frac{P(x|L=2)P(L=2)}{P(x)} = 86.49\%$$

$$P(L=3|x) = \frac{P(x|L=3)P(L=3)}{P(x)} = 5.41\% \quad P(L=4|x) = \frac{P(x|L=4)P(L=4)}{P(x)} = 7.73{\times}10^{-6}\%$$

The maximum a posteriori is $P(L=2|x)$ with 86.49% probability, so the image with the line in Column 2 is the most probable.

6 Character Classification

The a priori for the images $P(\omega_1) = 0.35$, $P(\omega_2) = 0.4$, $P(\omega_3) = 0.25$. Since the probability of error is bigger for white pixels, we have to calculate our likelihoods in a different way. A truly white pixel that is wrongly measured as black in x, translates to a wrong black = 0.3. The opposite of that is a correct white, which is 1 - 0.3 = 0.7. A truly black pixel that is wrongly measured as white in x translates, to a wrong white = 0.2. The opposite of that is a correct black, which is 1 - 0.2 = 0.8.

correct black = 0.8, wrong black = 0.3, correct white = 0.7, wrong white =0.2

To calculate the likelihood that x is ω_i we compare x with each ω_i using the above probabilities.

Calculating the likelihood for $P(x|\omega_1)$; there are 5 black pixels in the correct place, no black pixels in the wrong place, 5 white pixels in the correct place and 5 white pixels in the wrong place.

$$P(x|\omega_1) = (0.8)^5(0.7)^5(0.2)^5 = 1.7623 \times 10^{-5}$$

Calculating the likelihood for $P(x|\omega_2)$; there are 3 black pixels in the correct place, 2 black pixels in the wrong place, 5 white pixels in the correct place and 5 white pixels in the wrong place.

$$P(x|\omega_1) = (0.8)^3(0.3)^2(0.7)^5(0.2)^5 = 2.4783 \times 10^{-6}$$

Calculating the likelihood for $P(x|\omega_3)$; there are 4 black pixels in the correct place, 1 black pixel in the wrong place, 7 white pixels in the correct place and 3 white pixels in the wrong place.

$$P(x|\omega_3) = (0.8)^4(0.3)(0.7)^7(0.2)^3 = 8.0958 \times 10^{-5}$$

$$P(x) = P(x|\omega_1)P(\omega_1) + P(x|\omega_2)P(\omega_2) + P(x|\omega_3)P(\omega_3) + P(x|\omega_4)P(\omega_4)$$

The a posteriori probabilities are:

$$P(\omega_1|x) = \frac{P(x|\omega_1)P(\omega_1)}{P(x)} = 22.51\% \quad P(\omega_2|x) = \frac{P(x|\omega_2)P(\omega_2)}{P(x)} = 3.62\%$$

$$P(\omega_3|x) = \frac{P(x|\omega_3)P(\omega_3)}{P(x)} = 73.87\%$$

The maximum a posteriori is $P(\omega_2|x)$ with 73.87% probability, so the '0' is the most probable image.

7 The OCR system - part 2 - Feature extraction

Before trying to extract features from the image I started by applying a median filter to get rid of any noise in the image. The next preprocessing step I took was to translate the digit to the center of the image to avoid getting features based on the position of the number. The features I used are as follows:

1. I convolved the image with a Sobel G_x that gives the horizontal edges of an image; this feature should highlight parts of numbers that have horizontal strokes and digits like 4,5 and 7 should have high values for it. I took the absolute value of the convolution and then took the mean to represent it as a single number so it can be passed to the feature vector.

2. I convolved the image with the G_y the vertical counterpart of the feature in 1, and it should have high values for digits with vertical strokes like 1 and 7. The next feature $feature2_1$ was a combination of the horizontal and vertical Sobel filters.

3. On this feature, I detected the Harris features in each image, and took the length of the result. The idea is that similar numbers will have the same number of corners.

4. This feature is represented by 3 numbers in the feature vector. The idea was to take the sum of pixels along the center columns of each image (since each image was centered). I found out that all the images had their pixels between column 61 and 80. I divided the columns into two parts, $f5_1$ and $f5_2$ holding the parts respectively. The idea was to have a column histogram to provide the number's profile. $f5_3$ holds the mean of how the pixels are distributed in the columns. The idea is that similar numbers will have similar profiles.

5. This feature is the same as the one above but along the rows instead. With $f5_1$ $f5_2$ and $f5_3$ holding the two bins and the mean respectively.

6. This feature also follows the same logic as the one before it however it looks at another statistical variable; the variance along the rows.

7. This feature also looks at the variance but this time along the columns.

8. This feature looks at the variance of the entire image. The idea being that the pixels in the same numbers will vary in a similar way.

9. Feature 9 looks at the area of the on pixels in the image. Also, the rationale was that similar digits will have similar areas.

10. This feature looks at the perimeter of the on pixels; this is complementary to the feature before it.

11. The features that follow are the width and height of the bounding box of the main region of the image (the digit of interest).

12. The equivalent diameter, convex area, maximum Feret diameter and minumum Feret diameter were chosen as features in an attempt to have features that describe the roundness and curvy nature of some numbers like 2,3,5,6,8,9 and 0.

Note: Almost all features were divided by the highest value of that feature to normalize the data.

```
function features = segment2features(I)

%first step; filter the image to remove noise; I'll use a median
    filter
If= medfilt2(I);
```

```
%second step; centering the image
% Find centroid.
props = regionprops(If, 'Centroid');
% Translate the image.
xt = props.Centroid(1);
yt = props.Centroid(2);
% Get center of image
[rows, columns] = size(If);
xc = columns/2;
yc = rows/2;
deltax = xc - xt;
deltay = yc - yt;
Ic = imtranslate(If,[deltax deltay],'FillValues', 0);

%first feature the horizontal edges of the image
gx= [1 0 -1; 2 0 -2; 1 0 -1];
f1 = abs(conv2(gx,Ic));
feature1 = mean2(f1);

%second feature the vertical edges of the image
gy= [-1 -2 -1; 0 0 0; 1 2 1];
f2 = abs(conv2(gy,Ic));
feature2 = mean2(f2)*10;

%horizontal and vertical edges of the image
gxy = [1 1 0;1 0 -1;0 -1 -1];
f2_2 = abs(conv2(gxy,Ic));
feature2_1 = mean2(f2_2);

%third feature gotten from harris corners
corners = detectHarrisFeatures(Ic);
feature3 = length(corners)/14;

%fourth feature sum along columns
sum_along_1= sum(Ic,1);
sum_1 = sparse(sum_along_1);
f4_1 = sum(sum_1(61:70))/121;
f4_2 = sum(sum_1(71:80))/108;
f4_3 = mean(sum_along_1,"all")/1.6;

%fifth feature sum along rows
sum_along_2= sum(Ic,2);
sum_2 = sparse(sum_along_2);
f5_1 = sum(sum_2(1:14))/123;
f5_2 = sum(sum_2(15:28))/106;
f5_3 = mean(sum_along_2, "all")/8;

%sixth feature variance of number of black pixels in rows
variance_row = var(Ic,1,2);
f6 = var(variance_row, 1, "all")*1000;

%seventh feature variance of number of black pixels in columns
variance_col = var(Ic,1,1);
f7 = var(variance_col, 1, "all")*100;

%eight feature variance of entire image at the same time
```

```
f8 = var(Ic,1,'all')*10;

f9 = bwarea(Ic)/289;

perimeter = bwperim(Ic);
f10 = (sum(perimeter,"all"))/90;

props = regionprops(Ic,'all');
width = props.BoundingBox(3)/16;
height = props.BoundingBox(4)/22;
convex_area = props.ConvexArea/278;
diameter = props.EquivDiameter/14;
feret1 = props.MaxFeretDiameter/24;
feret2 = props.MinFeretDiameter/14;

features = [feature1 feature2 feature2_1 feature3 f4_1 f4_2 f4_3
    f5_1 f5_2 f5_3 f6 f7 f8 f9 f10 width height diameter feret1
    feret2 convex_area];
```

Listing 1: Code

```
Studying the character 1
There are 4 examples in the database.
The feature vectors for these are:

ans =

    0.5122    0.7122    0.4496    0.6919
    0.4395    0.4122    0.2147    0.4293
    0.4324    0.5172    0.2767    0.4946
    0.2143    0.4286    0.1429    0.4286
    0.2782    0.4493    0.2696    0.4325
    0.2346    0.2930    0.1887    0.2654
    0.2634    0.3839    0.2366    0.3616
    0.2772    0.3849    0.2310    0.3738
    0.2349    0.3646    0.2319    0.3304
    0.2634    0.3839    0.2366    0.3616
    0.1785    0.2097    0.2179    0.2152
    0.1049    0.1905    0.1529    0.1763
    0.1208    0.1912    0.1243    0.1832
    0.3131    0.4157    0.2517    0.4014
    0.3778    0.5000    0.3333    0.5111
    0.6875    0.5625    0.2500    0.5625
    0.5455    0.8182    0.5000    0.7727
    0.4559    0.6031    0.4836    0.5699
    0.6783    0.8385    0.4751    0.8015
    0.3293    0.4000    0.2857    0.4518
    0.1547    0.2770    0.1403    0.3022
```

Listing 2: Text Results of Benchmark Script