Doodleverse Lab
Brandon Quach

Warmup:

For a simple image, say a triangle with a black 1 pixel wide edge on a white background, suggest a reasonable edge detector to find the contour of the image. Why might a simple edge detector be "good enough" for this application?

- We can use the sobel operator to detect the change in x and y coordinates of the triangle. A simple edge detector is good enough because there is no other edges in the image that could interfere with the triange

Given an arbitrary two-dimensional shape, determine some candidate feature points that will always exist.

- Where the edge changes directions, i.e when the magnitude of the change in x or y changes

With the star dataset being used in this application taken into consideration, why is it so important to represent the shape in as few feature points as possible?

- In regards to the stars, the stars are not always forming the proper shape nor is there enough stars to be able to find a certain shape. It also makes the shapes easier to find.

Use this script to convert the 79 stars found in stars.mat to (x,y) coordinates centered at ($_c$; $_c$) = (293 ; -2.8). (we have converted hours to degrees for you, but be sure to convert degrees to radians for the function defined above)

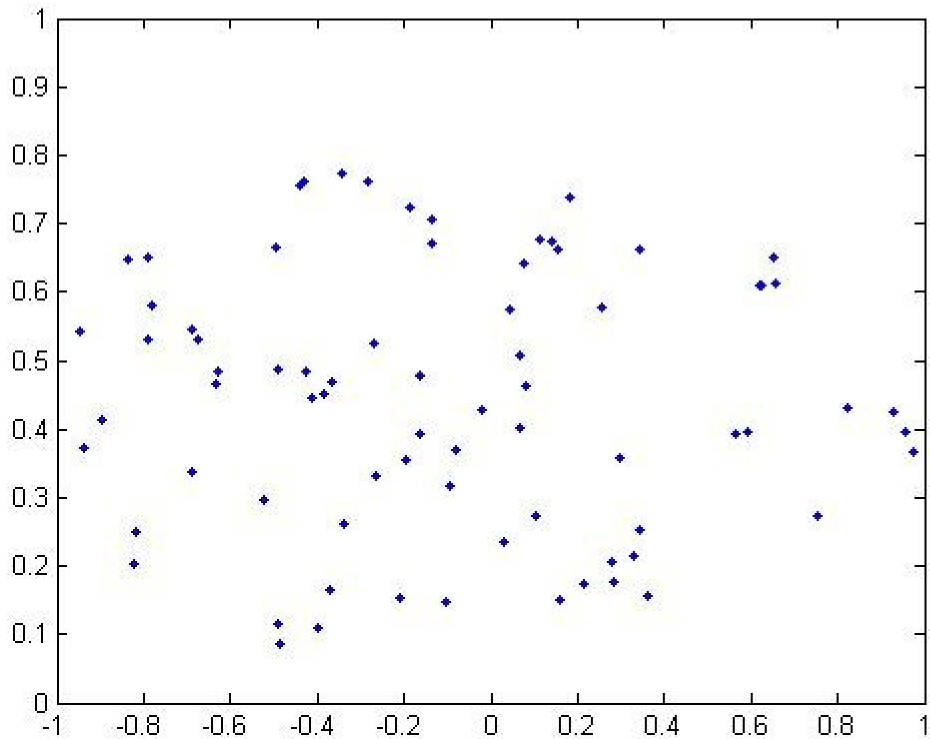
```
function[x,y] = project(ra, dec, c_ra , c_dec)

    theta_0 = dec - c_dec;
    epsilon = 10^(-6);
    error = 1+epsilon;
    while error > epsilon
        m = (2*theta_0+sin(2*theta_0)-pi*sin(dec - c_dec))/(2+2*cos(2*theta_0));
        theta_1 = theta_0 - m;
        error = abs(theta_1 - theta_0);
        theta_0 = theta_1;

    x = 2*sqrt(2)*(ra-c_ra)*cos(theta_0)/pi;
    y = sqrt(2)*sin(theta_0);
    end
end
mat = zeros(2,79);
for n = 1:79
   a=RA(n);
   b=Dec(n)*0.0174533;
   [x,y]= project(a,b,293,-2.8);
   mat(1,n) = x/15
   mat(2,n) = y
end

plot(79); hold on;
axis([-1 1 0 1])
for i=1:79
    a = mat(1,i);
    b = mat(2,i);
```

```
    plot(a,b,'b.');
end
```



Form this as a clustering problem X = DW. What do X,D,W look like? Our search set contains 79 stars. If D is n x k, what is n? What would be the effect of choosing smaller or larger values of k?

- X will be an n x 1 matrix containing the result of the data, D will be an n x k matrix containing the star data, and W will be a k x 1 matrix containing the weights to be put on each of the features. n is the number of stars, so in this case it is 79. A larger k would allow more features, and therefore would allow for a larger set of weights to be applied.

This brute-force method only finds k possible tests, where we choose k ahead of time. Why might the clustering approach be more preferable?

- it will allow us to find more tests that would work that may not be found when a limit is imposed on the testing process

Let UV^T be the SVD of the matrix SF^T. Show that the orthogonal T that minimizes $\|TF - S\|^2_F$ is equal to UV^T.

Use the Procrustes algorithm to find the transformation that best maps the A to B matrix found in the procrustes.mat file.

0.5000	0.8660
-0.8660	0.5000

Suppose we had k such possible matches {S_i,M_i}, and we want to determine the best one. Set up a cost/penalizer function that we are trying to minimize.

Lab:

1.

```
im = imread('diamond.png');
image(im);
B = (0.2989 * double(im(:,:,1)) + 0.5870 * double(im(:,:,2)) + 0.1140 *
double(im(:,:,3)))/255;
C=double(B);
for i=1:size(C,1)-2
    for j=1:size(C,2)-2
        Gx=((2*C(i+2,j+1)+C(i+2,j)+C(i+2,j+2))-(2*C(i,j+1)+C(i,j)+C(i,j+2)));
        Gy=((2*C(i+1,j+2)+C(i,j+2)+C(i+2,j+2))-(2*C(i+1,j)+C(i,j)+C(i+2,j)));
        B(i,j)=sqrt(Gx.^2+Gy.^2);

    end
end
```

```
2.(289,115) , (290,115)
  (180,202)
  (289,289) , (289,290)
  (399,202)
     the points give a very close match

3. (289,115), (180,202), (289,289)-> 77.1913 degrees
```

```
(RA,Dec,Mag)
```

-0.3431, 0.7738, 3.760,
-0.4849,0.0873, 4.0200,
0.7533,0.2747, 5.3000 → 77.1941 degrees

0.0308, 0.2356, 4.4500
0.0684, 0.508, 4.9300
-0.6276, 0.4838, 5.38 → 77.1953 degrees

0.3299, 0.2148, 4.71
-0.6350, 0.4660, 4.22
0.3439, 0.6632, 5.01 → 77.1958 degrees

4.

5.

Feedback:

- I ran out of time, so I wasn't able to finish 4&5.
- Interesting lab concept
- I had no python experience, so I had to figure out what some of the operations were to convert to matlab. Not really a problem with the lab, just explaining my process during the lab
- warm up was a good step by step process that also takes you through the lab
- not 100% sure what numbers in brackets meant(Contour of shape determined by [1]). A reference to earlier problem, or equation? Could number them to clear that up.
- Maybe put some more comments in the code to tell reader what is being calculated at some steps
- could possibly explain how hours was converted to degrees for the mollyweid. Also how to convert degrees to radians