Nicholas Bush

Math 50

1.

function [ one\_Norm, two\_Norm, infinity\_Norm ] = unitvector( vector )

%takes in a vector and outputs its unit vector using the one norm, two norm

%and infinity norm

[m, n] = size(vector);

if ((norm(vector, 1)== 0))

one\_Norm = zeroes(m, n);

else one\_Norm = (vector)./norm(vector,1);

end

if ((norm(vector, 2)== 0))

two\_Norm = zeroes(m, n);

else two\_Norm = (vector)./norm(vector,2);

end

if ((norm(vector, Inf)== 0))

infinity\_Norm = zeroes(m, n);

else infinity\_Norm = (vector)./norm(vector,Inf);

end

end

2.

a. to find the eigenvalues in matlab you must type in eig(x) where x is the vector you want the eigenvalues of.

b. to find eigenvectors in matlab you must type in [V,D] = eig(x) where x is your vector it will then return a matrix V with your eigenvectors.

c. a vector whose length is one.

d. [V,D] = eig(X,'nobalance') it does the eigenvector/ eigenvalue things without normalizing.

3.

A. [V,D] = eig(A,'nobalance')

Eigenvector V =

-1.0000 -0.3333 0 0

-0.5000 -1.0000 0 0

0 0 -1.0000 0.5000

0 0 -0.2500 -1.0000

Eigenvalues D =

-2 0 0 0

0 3 0 0

0 0 -6 0

0 0 0 3

B. [V,D] = eig(B,'nobalance')

Eigenvector V =

-1.0000 -1.0000 0 0

0 0 -1.0000 1.0000

-0.3333 -0.5000 0 0

0 0 0.7500 -1.0000

Eigenvalue D =

-2.0000 0 0 0

0 -1.0000 0 0

0 0 -2.0000 0

0 0 0 -1.0000

4.

the eigen values are -4, 1, and 3 because if we remove any of the central diagonal from a upper or lower triangular matrix its determinant will become zero and hence the values on the diagonal will be the eigenvalues.

function [ spectralradius ] = spectralradius( vector )

%finds spectral radius

X = eig(vector);

i=1;

spectralradius = 0;

while (i <= length(X))

if (spectralradius<abs(X(i)))

spectralradius = abs(X(i));

end

i= i+1;

end

end

5. In order to use this must have downloaded file called fuka.jpg and have it in matlab, I’ll include a picture of the results at the end of this.

clf

tic

% Load image

myPic = imread('fuka.jpg'); %JPG, GIF, PNG, etc. file

figure(1)

image(myPic), axis image

title('Original picture')

p = myPic; %color red

p( :, :, 2) = 0;

p( :, :, 3) = 0;

q = myPic;

q( :, :, 1) = 0;

q( :, :, 3) = 0;

r = myPic;

r( :, :, 1) = 0;

r( :, :, 2) = 0;

% Compare results

figure(2)

subplot(2,2,1)

imagesc(myPic), axis image, title('Original picture')

subplot(2,2,2)

imagesc(p), axis image, title('red image')

subplot(2,2,3)

imagesc(q), axis image, title('green image')

subplot(2,2,4)

imagesc(r), axis image, title('blue image')

toc



Personally I think the green one looks the best.

6. must also have fuka.jpg for this one too, results still included.

r = 40; %number of desired, principal components (1 < r < n)

clf

tic

% Load image

myPic = imread('fuka.jpg'); %JPG, GIF, PNG, etc. file

figure(1)

image(myPic), axis image

title('Original picture')

X = double(rgb2gray(myPic)); %convert to grayscale matrix

Y = double(rgb2gray(myPic));

% Intermediate calculations

[m, n] = size(X);

v = 1;

while (v<n)

b=1;

while (b<m)

X(b, v) = 255 - X(b, v);

b = b+1;

end

v=v+1;

end

% Compare results

figure(3)

subplot(3,1,1)

imagesc(myPic), axis image, title('Original picture')

subplot(3,1,2)

imagesc(Y), axis image, colormap(gray), title('Grayscale picture orig')

subplot(3,1,3)

imagesc(X), axis image, colormap(gray), title('Grayscale picture reverse')

toc



7. findings for 7

I chose many numbers but the ones that worked best were above 100.

as we increase the number r, the picture becomes more and more visually pleasing, however after a certain point it becomes nearly un noticeable (for my image that sweet spot seemed to be between 100 and 200), unless you have the difference chart directly beside the images.

8.

x = [-1, 1, 2];

y = [-4, -2, 5];

z = polyfit(x,y,2)

%2x^2 + x – 5 this is the polynomial that fits.

x=linspace(-1,4,100);

y = (2\*x).^2 +x -5;

plot(x,y)

9.

X= [32 31 53 28 27 36 42 30 34 46]

Y = [103 103 86 57 32 131 157 20 27 161]

z = polyfit(X,Y,1)

z =

3.4670 -36.7669

This means 3.4670 x -36.7669 (just proving I know what it means☺)

polyval(z,51.800)

ans =

142.8260

Because 142.8260 is not close to 329, the predicted value is not close to the true value.

10.



I think that the lim of x^n as n goes to infinity is merely a straight line up at the point where x = 1

Code:

x = linspace(0,1,1000);

y1 = x;

y2 = x.^2;

y3 = x.^3;

y4 = x.^4;

y5 = x.^5;

y6 = x.^6;

%i assume you wanted y = x^6 aswell although it is not stated... it does

%say all 6 graphs when there are only five on the assignment.

subplot(2,3,1)

plot(x,y1)

title('x vs y');

xlabel('x');

ylabel('y');

subplot(2,3,2)

plot(x,y2)

title('x vs y^2');

xlabel('x');

ylabel('y');

subplot(2,3,3)

plot(x,y3)

title('x vs y^3');

xlabel('x');

ylabel('y');

subplot(2,3,4)

plot(x,y4)

title('x vs y^4');

xlabel('x');

ylabel('y');

subplot(2,3,5)

plot(x,y5)

title('x vs y^5');

xlabel('x');

ylabel('y');

subplot(2,3,6)

plot(x,y6)

title('x vs y^6');

xlabel('x');

ylabel('y');