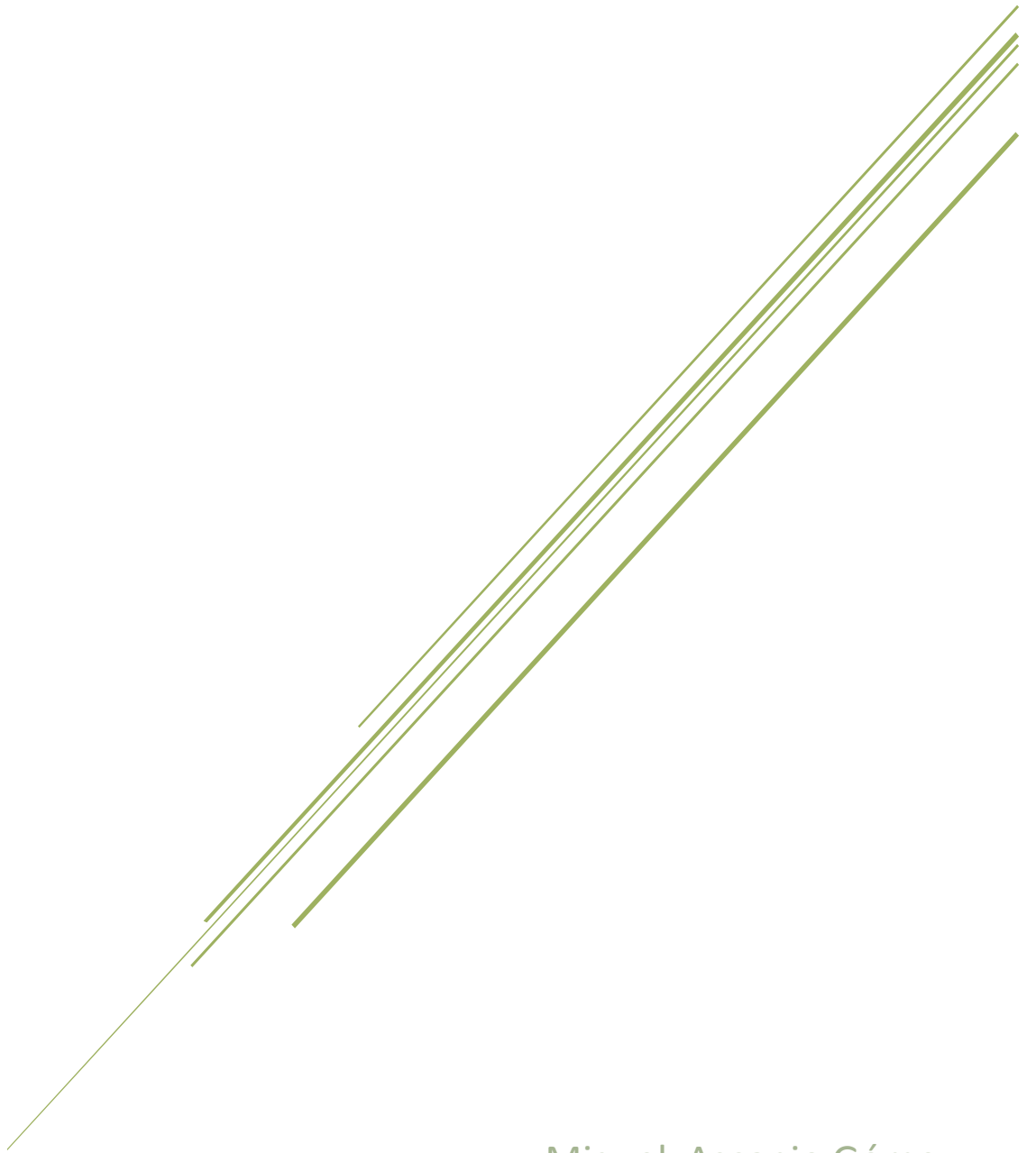


APRENDIZAJE AUTOMÁTICO

Práctica 7



Miguel Ascanio Gómez
Esther Ávila Benito

En esta práctica se presenta una implementación del algoritmo de clustering k-means aplicándolo a un ejemplo de datos. También se realiza una aplicación del citado algoritmo para la compresión de una imagen reduciendo su número de colores.

ALGORITMO K-MEANS

La función *findClosestCentroids* indica el centroide más cercano a cada ejemplo para agruparlos en la clase de dicho centroide.

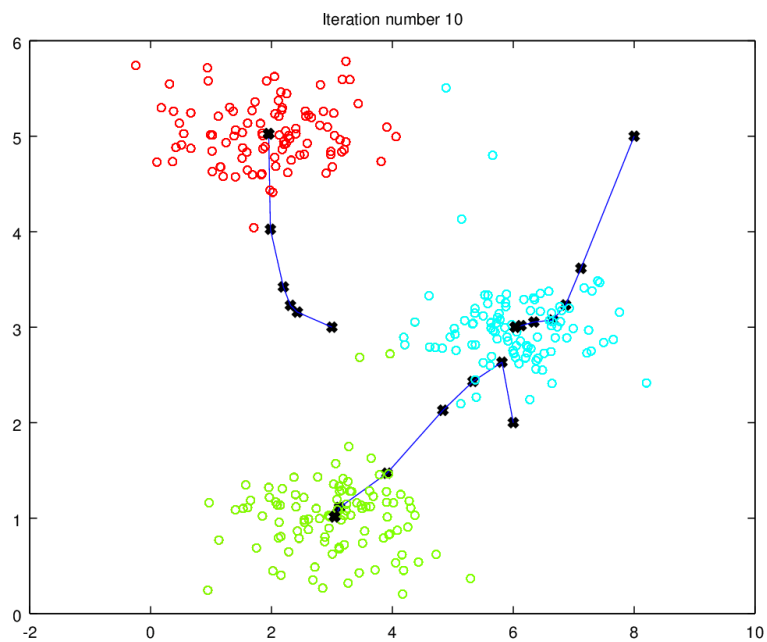
```
function idx = findClosestCentroids(X, centroids)
    m = size(X)(1);
    k = size(centroids)(1);

    for i=1 : m % Por cada ejemplo de entrenamiento
        min = inf;
        posMin = 0;
        for j=1 : k
            suma = sum((X(i,:) .- centroids(j,:)) .^ 2);
            if (suma < min)
                min = suma;
                posMin = j;
            endif
        endfor
        idx(i) = posMin;
    endfor

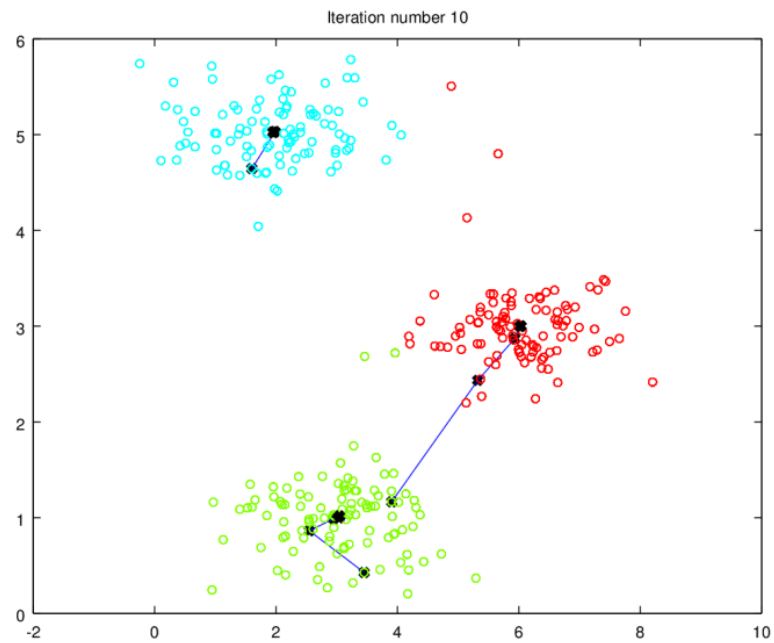
endfunction
```

También se ha realizado un cálculo de centroides aleatorios:

```
function centroids = randomCentroids(X, K)
    randidx = randperm(size(X,1));
    centroids = X(randidx(1:K),:);
endfunction
```



Agrupación de los ejemplos en las clases según los centroides obtenidos



Agrupación de los ejemplos según centroides aleatorios

La función `computeCentroids` calcula la nueva posición de los centroides a partir de los índices de los centroides asignados a cada ejemplo:

```
function centroids = computeCentroids(X, idx, K)
    for k = 1:K
        C = X(find(idx == k), :);
        centroids(k,:) = mean(C);
    endfor
endfunction
```

COMPRESIÓN DE IMÁGENES

En este apartado se realiza la compresión de una imagen reduciendo el número de colores. Tras la lectura de la imagen, obtenemos una matriz con todos los píxeles y la proporción de rojo, azul y verde de cada uno.

A partir del siguiente código cargamos la imagen original:

```
A = double(imread ("bird_small.png" )) / 255;
```

Con el algoritmo *k-means* expuesto anteriormente, se hallan los colores más representativos a los que se va a reducir la imagen. En este caso, los centroides contendrán los valores de rojo, azul y verde de los colores representativos.

```
res = reshape (A, 128*128,3);  
iniCent = randomCentroids(res, 16);  
  
[centroids, idx] = runkMeans(res, iniCent , 10, false);  
  
X = centroids(idx',:);  
  
sol(:,:,1) = reshape (X(:,1), 128,128);  
sol(:,:,2) = reshape (X(:,2), 128,128);  
sol(:,:,3) = reshape (X(:,3), 128,128);  
  
imagesc(sol);
```



Imagen original



Imagen reducida a 16 colores