Universidad de Cádiz

Reconocimiento de Patrones

Ingeniería en Informática

Examen – 17 de junio de 2010

Hora: 17 horas

PROBLEMA 1

El algoritmo "Fuzzy K-means" permite obtener K puntos representativos de un conjunto de datos, al igual que hace el algoritmo de las K-medias. La diferencia fundamental entre ambos es que mientras en el algoritmo de las K-medias, cada dato se asigna a un centroide de forma inequívoca, en el algoritmo "Fuzzy K-means" se supone que cada dato puede pertenecer a cualquier centroide con una cierta probabilidad. Obviamente si está más cerca de un centroide pertenecerá a éste con mayor probabilidad que a otro centroide más lejano. El algoritmo se puede describir como:

1. Inicializar los k clusters (por ejemplo, usando el método de las K-medias)
2. Repetir hasta alcanzar la convergencia *(el alumno tendrá que fijar este criterio)*
   a. Calcular un valor que mida la probabilidad de que cada dato pertenezca a cada centroide en función de la distancia al mismo.
   b. Recalcular los centroides, usando adecuadamente los datos y las probabilidades correspondientes obtenidas anteriormente.

En el desarrollo del algoritmo, recuerde que los valores de probabilidad han de estar comprendidos en el intervalo [0,1] y que la suma de las probabilidades de todos los posibles resultados de un experimento, suponiéndolos mutuamente excluyentes, ha de ser 1.

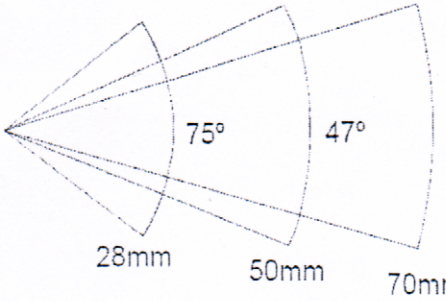
Se pide escribir el código MATLAB de una función con el algoritmo "Fuzzy K-means".

SOLUCION

```
function c = fuzzykmeans(x,K)
  c = kmeans(x,K);  % Inicializamos usando kmedias
  terminado = 0;
  c_old=c;
  it = 1;
  while terminado==0

    % Calculamos las distancia de cada dato a cada centroide
    for k=1:K
        d(k,:) = d_euclid(x,c(:,k))+1e-6;
    end

    % Calculamos la pertenencia
    a = (1./d).^3;   % Este término depende de la implementación
    suma = sum(a);
    u = a ./ repmat(suma,K,1);

    % Actualizamos los centroides
    for k=1:K,
        aux    = repmat(u(k,:),2,1);
        c(:,k) = sumpat(x.*aux) ./ sumpat(aux);
    end

    % Condicion de parada
    if (it>100) | (sumsqr(c_old - c)<1e-6),
        terminado=1;
    else
        it = it + 1;
        c_old=c;
    end
  end
end
```

PROBLEMA 2

En un folleto de instrucciones de una cámara fotográfica aparecen los siguientes datos de la medida de la distancia focal y el ángulo de visión correspondiente:



Zoom = { 28 ,50 ,70 }

Ángulo de visión = { 75° ,47° ,34° }

Se desea ajustar un modelo según la ecuación $angulo\_de\_vision = a \cdot zoom^b$, para calcular el ángulo de visión a partir del valor de zoom utilizando el método de los mínimos cuadrados.

a. Realice una función para calcular los coefs. a y b por el método de mínimos cuadrados
b. Realice una función para dibujar en una sola gráfica la curva en el intervalo 20..100 y los datos del problema.
c. ¿Puede calcular a y b con la calculadora? Si no dispone de ella, plantee qué cálculos hay que realizar a partir de los valores de zoom e angulo_de_vision dados.

SOLUCION

```
function [a,b]=ajuste_potencial(x,y),
  %A=[log(x) ones(3,1)];
  %b = log(y);
  %coefs = pinv(A)*b

  coefs = polyfit(log(x),log(y),1);
  a = exp(coefs(2));
  b = coefs(1);
end

function dibuja_ajuste_potencial(a,b,xi)
  yi = a * xi.^b;
  plot(xi,yi,'b');hold off
end
```

El programa principal podría ser algo así:

```
  x=[28 50 70]';
  y=[75 47 34]';
  xi = 20:100;
  [a,b]=ajuste_potencial(x,y)
  plot(x,y,'or');hold on;
  dibuja_ajuste_potencial(a,b,xi);hold off;
```

Para calcular a y b con la calculadora habría que calcular el logaritmo neperiano de x e y, y trabajar con ellos.

```
      xlog=[log(28) log(50) log(70)]= [3.3322    3.9120    4.2485]
      ylog=[log(75) log(47) log(34)]= [4.3175    3.8501    3.5264]

Luego calcularíamos los términos siguientes:
      Sxx = sum(xlog .* xlog)     = 44.4572
      Sx  = sum(xlog)             = 11.4927
      Sxy = sum(xlog.*ylog)       = 44.4303
      Sy  = sum(ylog)             = 11.6940
```

A continuación habría que resolver el sistema de ecuaciones siguiente:

$$\begin{bmatrix} Sxx & Sx \\ Sx & 3 \end{bmatrix} * \begin{bmatrix} C1 \\ C2 \end{bmatrix} = \begin{bmatrix} Sxy \\ Sy \end{bmatrix}$$

Para ello, multiplicamos la segunda ecuacion por Sx/3, el sistema tendrá la misma solución, quedando:

$$\begin{bmatrix} Sxx & Sx \\ Sx*Sx/3 & Sx \end{bmatrix} * \begin{bmatrix} C1 \\ C2 \end{bmatrix} = \begin{bmatrix} Sxy \\ Sx*Sy/3 \end{bmatrix}$$

Finalmente, restamos a la última fila la primera

$$\begin{bmatrix} Sxx & Sx \\ Sx*Sx/3-Sxx & 0 \end{bmatrix} * \begin{bmatrix} C1 \\ C2 \end{bmatrix} = \begin{bmatrix} Sxy \\ Sx*Sy/3-Sxy \end{bmatrix}$$

De donde, de la segunda ecuación del sistema, podemos obtener C1 como:

$$C1 = \frac{Sx*Sy/3-Sxy}{Sx*Sx/3-Sxx} = -0.8571$$

Y por tanto, de la primera ecuación obtenemos C2:

C2 = (Sxy-Sxx*a)/Sx = 7.1815

Finalmente, los coeficientes pedidos son:

a = exp(C2) = 1.3149e+003
b = C1 = -0.8571