



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA
Año 2018 - 1^{er} Cuatrimestre

PROCESAMIENTO DEL HABLA (86.53)

PARCIALITO

TEMA: Entrenamiento no supervisado

FECHA: 6 de mayo de 2019

INTEGRANTES:

Manso, Juan

<juanmanso@gmail.com>

- #96133

Índice

EM	1
Separo en test y train	1
Gráficos conjunto de puntos	1
Entrenamiento:	1
Preparo iteraciones:	1
Preparo el gráfico	2
Iteración principal	2
Plots de datos tras entrenamiento	3
Plot likelihood	11
Test	11
A. Funciones auxiliares	13
A.a. discriminante.m	13
A.b. estim_cov.m	13
A.c. estim_sigma.m	14
A.d. inicializacion.m	14
A.e. responsabilidad.m	16

EM

```
1 clear all
2 close all
3 %graphics_toolkit('gnuplot');
4 myGreen = [0 0.5 0];
5
6 a_total = load('c1.txt');
7 o_total = load('c2.txt');
8 u_total = load('c3.txt');
```

Separo en test y train

```
1 % a
2 ind_perm = randperm(length(a_total)); % Entreno en cada corrida con otras muestras
3 a_train = a_total(ind_perm(1:35),1:2);
4 a_test = a_total(ind_perm(36:end),1:2);
5 % o
6 ind_perm = randperm(length(o_total)); % Entreno en cada corrida con otras muestras
7 o_train = o_total(ind_perm(1:35),1:2);
8 o_test = o_total(ind_perm(36:end),1:2);
9 % u
10 ind_perm = randperm(length(u_total)); % Entreno en cada corrida con otras muestras
11 u_train = u_total(ind_perm(1:35),1:2);
12 u_test = u_total(ind_perm(36:end),1:2);
```

Gráficos conjunto de puntos

```

1 train = [a_train;o_train;u_train];
2
3 % Hago el 'scatter' de las muestras sin clasificar
4 figure
5 hold on
6 plot(train(:,1), train(:,2),'k.', 'MarkerSize', 10)
7 grid minor
8 close

```

Entrenamiento:

```

1 %Iniciación con puntos: usar 5 puntos de train con etiquetas para inicializar
2 %Iniciación supervisada
3 [a,b,c,d,e,f,g,h,i] = inicializacion(train, [3], length(a_train));
4 % Inicialización no supervisada (random)
5 %[a,b,c,d,e,f,g,h,i] = inicializacion(train, [5;'r'], length(a_train));
6
7 covg = d + e + f / 3;

```

Preparo iteraciones:

```

1 %Medias, covarianzas y pi's de las clases
2 ma = a; cova = covg; proba = g;
3 mo = b; covo = covg; proba = h;
4 mu = c; covu = covg; probu = i;
5
6 % Label
7 real_label = [ones(length(a_train),1);2*ones(length(o_train),1);3*ones(length(
   u_train),1)];
8
9 likelihood(2) = -10;
10 likelihood(1) = -10.1;
11 limlike = 0.01;
12 n = 2;

```

Preparo el gráfico

```

1 theta = linspace(0, 2*pi, 100);
2 rot = [cos(theta); sin(theta)];
3
4 figure(1); hold on;
5 % Medias con X
6 plot(ma(1),ma(2), 'rx','MarkerSize',10);
7 plot(mo(1),mo(2),'x','color',myGreen,'MarkerSize',10);
8 plot(mu(1),mu(2), 'bx','MarkerSize',10);
9 % Puntos bien clasificados
10 plot(train(:,1).*(real_label==1), train(:,2).*(real_label==1), 'r.', 'MarkerSize',1
   0);
11 plot(train(:,1).*(real_label==2), train(:,2).*(real_label==2), '.', 'color',
   myGreen, 'MarkerSize',10);
12 plot(train(:,1).*(real_label==3), train(:,2).*(real_label==3), 'b.', 'MarkerSize',1
   0);
13 % ELIPSES
14 ellipsea = (chol(cova)' * rot)' + ma;
15 ellipseo = (chol(covo)' * rot)' + mo;
16 ellipseu = (chol(covu)' * rot)' + mu;
17 plot(ellipsea(:,1),ellipsea(:,2),'r');
18 plot(ellipseo(:,1),ellipseo(:,2),'color', myGreen);

```

```

19 plot(ellipse(:,1),ellipse(:,2),'b');
20 axis tight
21
22 titles = 'Iteracion ';
23 title([titles, num2str(0)]);

```

Iteración principal

```

1 bool_plot = 1; % Si quiero ver los grafos de la iteración => 1. Sino 0.
2 N = length(train);
3
4 while( abs((likelihood(n-1) - likelihood(n))) > limlike && n<20)
5 % % % PASO E % % %
6 % Calculo de responsabilidades para cada muestra
7 for i = 1:length(train)
8 x = train(i,:);
9 res(1) = mvnpdf(x, ma, cova)*proba;
10 res(2) = mvnpdf(x, mo, covo)*probo;
11 res(3) = mvnpdf(x, mu, covu)*probu;
12 den = sum(res);
13
14 gama(i,:) = res/den;
15 end
16
17 % % % PASO M % % %
18 NA = sum(gama(:,1)); NO = sum(gama(:,2)); NU = sum(gama(:,3));
19 % Recalculo:
20 % media
21 ma = sum(train.*gama(:,1))/sum(gama(:,1));
22 mo = sum(train.*gama(:,2))/sum(gama(:,2));
23 mu = sum(train.*gama(:,3))/sum(gama(:,3));
24
25 % covarianza
26 cova = estim_cov(train, ma, gama(:,1));
27 covo = estim_cov(train, mo, gama(:,2));
28 covu = estim_cov(train, mu, gama(:,3));
29
30 % pi (probabilidad)
31 proba = sum(gama(:,1))/sum(sum(gama));
32 probo = NO/N;
33 probu = NU/N;
34
35 % % % LIKELIHOOD % % %
36 den = sum(mvnpdf(train,ma,cova)*proba) + sum(mvnpdf(train,mo,covo)*probo) + sum(
    mvnpdf(train,mu,covu)*probu);
37
38 % log_likelihood
39 likelihood(n+1) = sum(log(den));
40
41 if(bool_plot)
42 % Gráfico
43 figure(n); hold on;
44 % Medias con X
45 plot(mu(1),mu(2), 'bx','MarkerSize',10);
46 plot(ma(1),ma(2), 'rx','MarkerSize',10);
47 plot(mo(1),mo(2), 'x','color',myGreen,'MarkerSize',10);
48 % Puntos bien clasificados
49 plot(train(:,1).*(real_label==3), train(:,2).*(real_label==3), 'b.', '
    MarkerSize',10);
50 plot(train(:,1).*(real_label==1), train(:,2).*(real_label==1), 'r.', '
    MarkerSize',10);
51 plot(train(:,1).*(real_label==2), train(:,2).*(real_label==2), '.', 'color',
    myGreen, 'MarkerSize',10);
52 % Puntos clasificados
53 scatter(train(:,1), train(:,2), 10, gama);

```

```

54 % ELIPSES
55 ellipsea = (chol(cova)' * rot)' + ma;
56 ellipseo = (chol(covo)' * rot)' + mo;
57 ellipseu = (chol(covu)' * rot)' + mu;
58 plot(ellipseu(:,1),ellipseu(:,2),'b');
59 plot(ellipsea(:,1),ellipsea(:,2),'r');
60 plot(ellipseo(:,1),ellipseo(:,2),'color', myGreen);
61
62 xlabel('Primer formante [Hz]');
63 ylabel('Segundo formante [Hz]');
64 axis tight
65 title([titles, num2str(n-1)]);
66 end
67
68 n += 1;
69 end

```

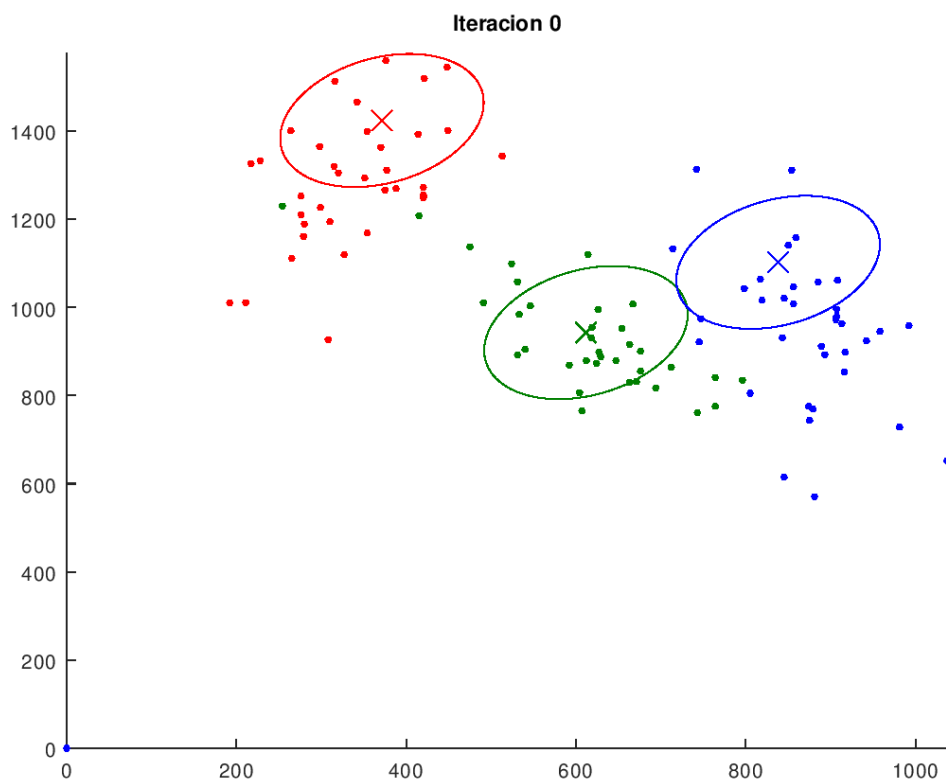


Figura 1: Antes de la iteración.

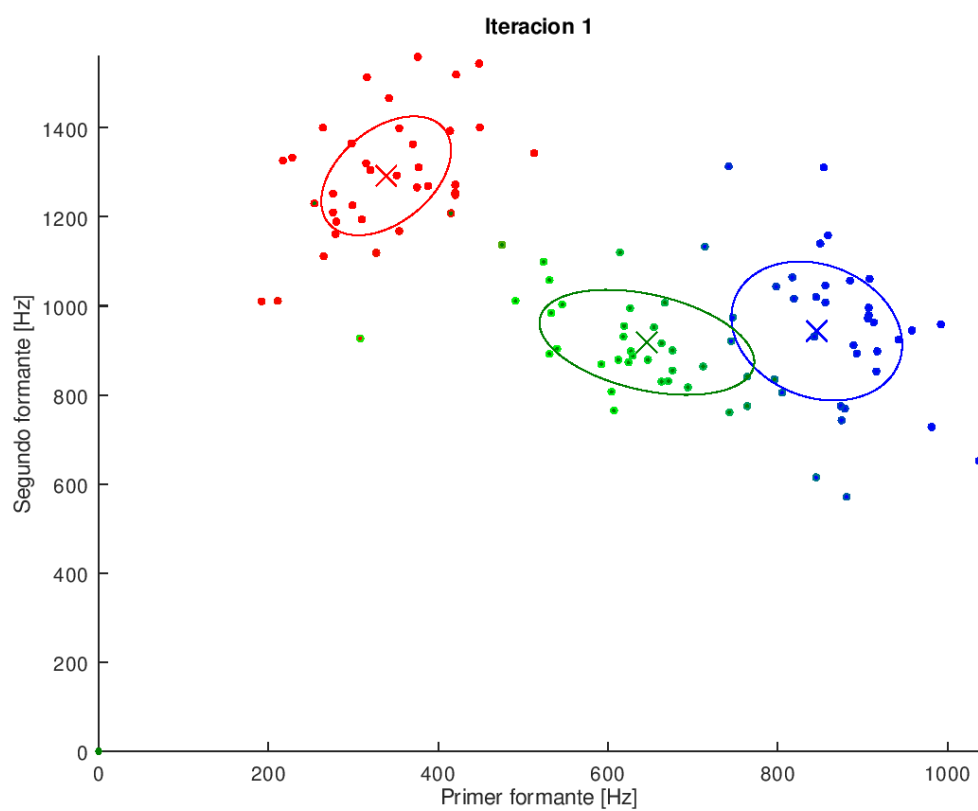


Figura 2: Tras la primer iteración.

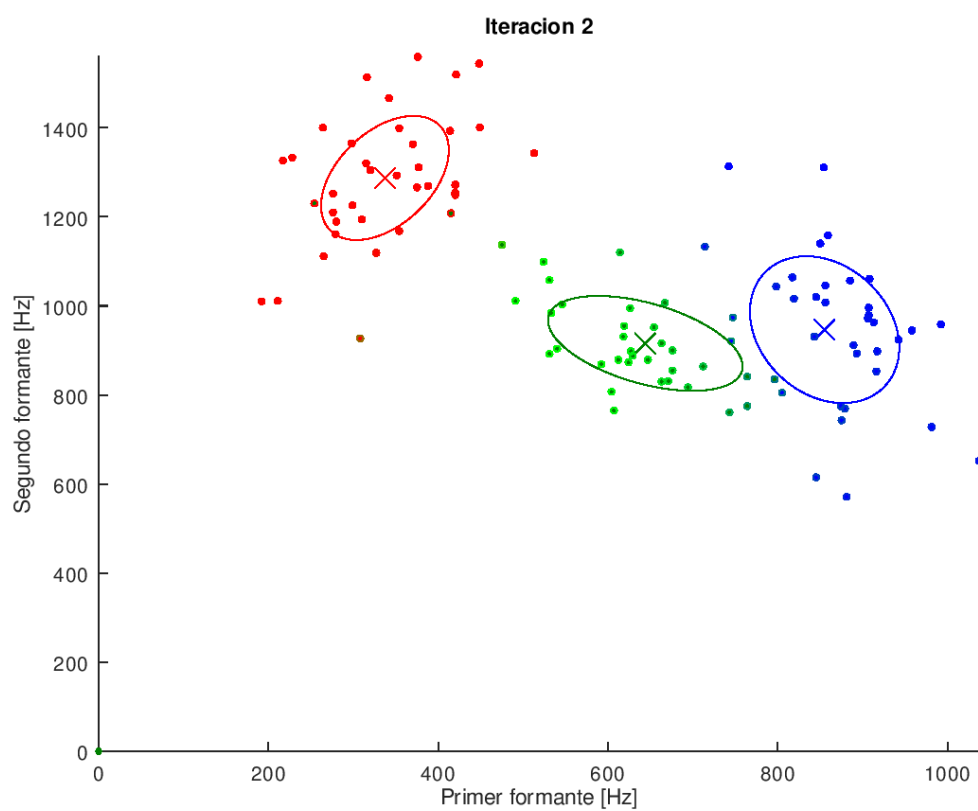


Figura 3: Tras la segunda iteación.

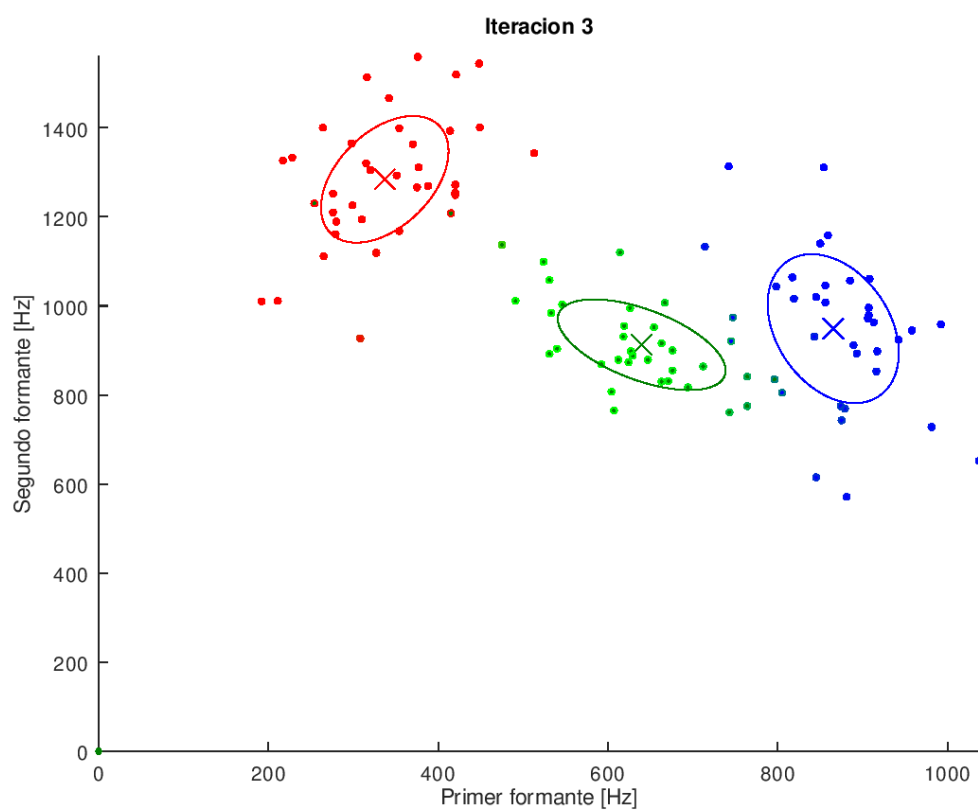


Figura 4: Tras la tercer iteración.

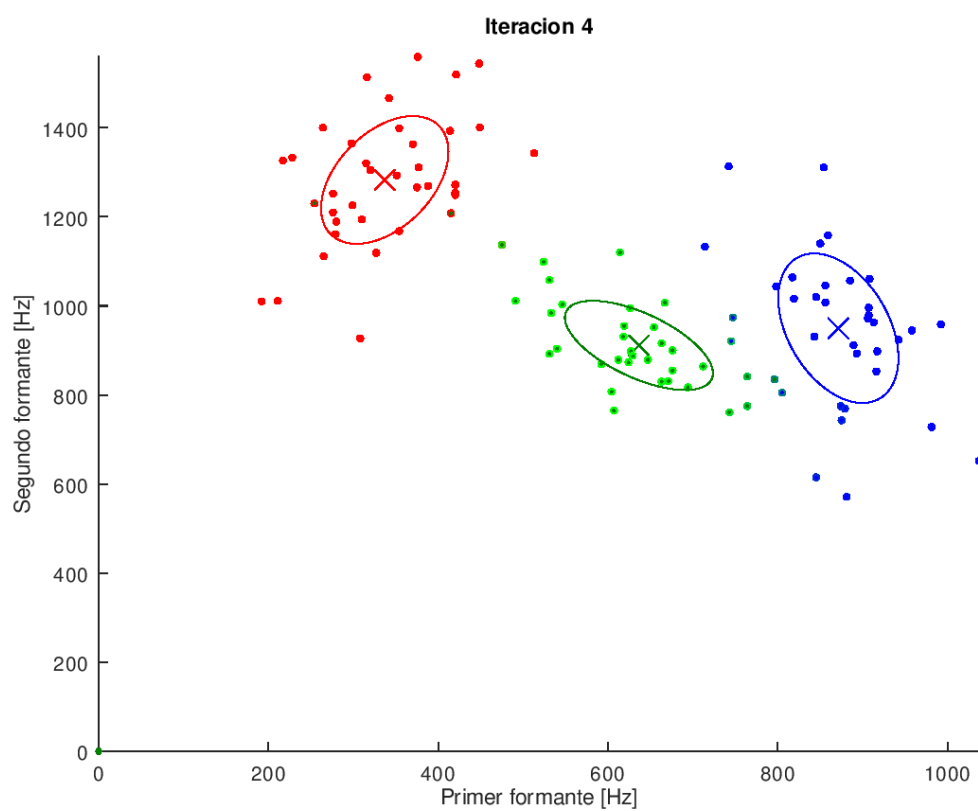


Figura 5: Tras la cuarta iteración.

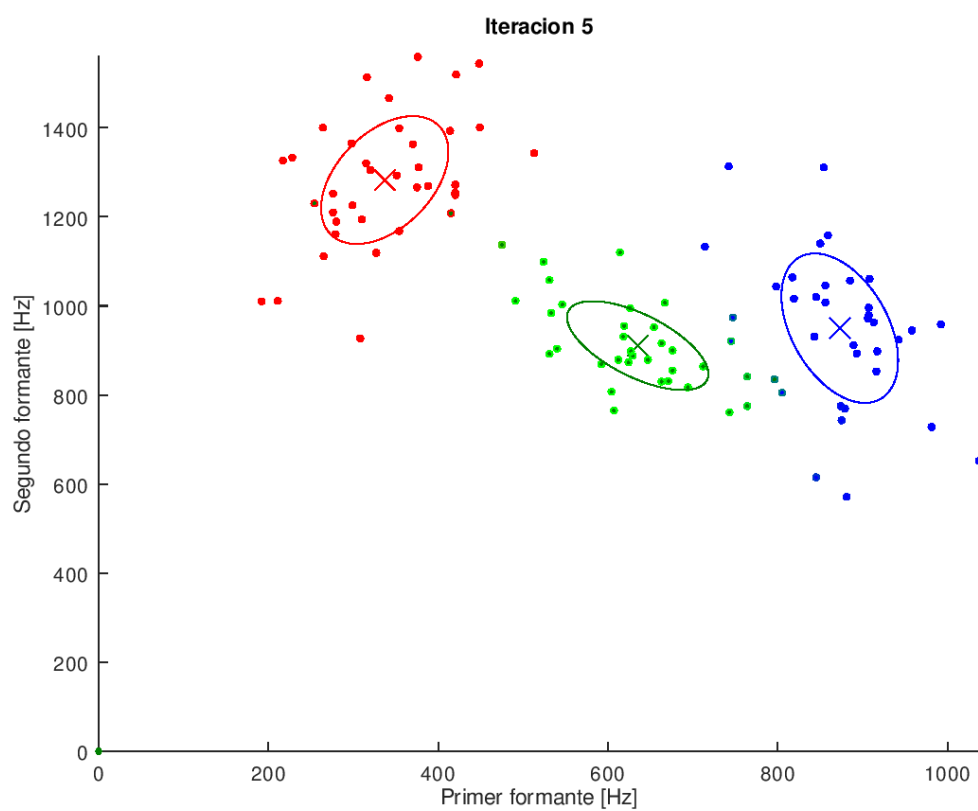


Figura 6: Tras la quinta iteación.

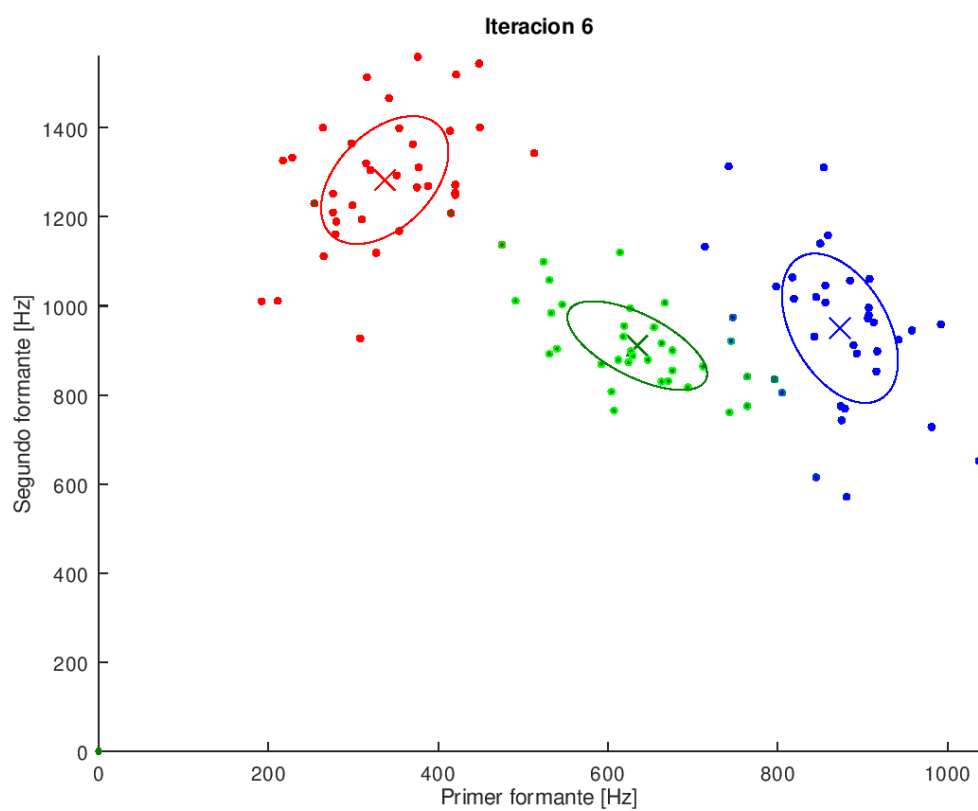


Figura 7: Tras la sexta y última iteración.

Plots de datos tras entrenamiento

Es el último gráfico de la iteración

Plot likelihood

```

1 figure
2 plot(likelihood);
3 title('Logaritmo de la verosimilitud en función de las iteraciones');
4 xlabel('Iteración');
5 ylabel('log(likelihood)');

```

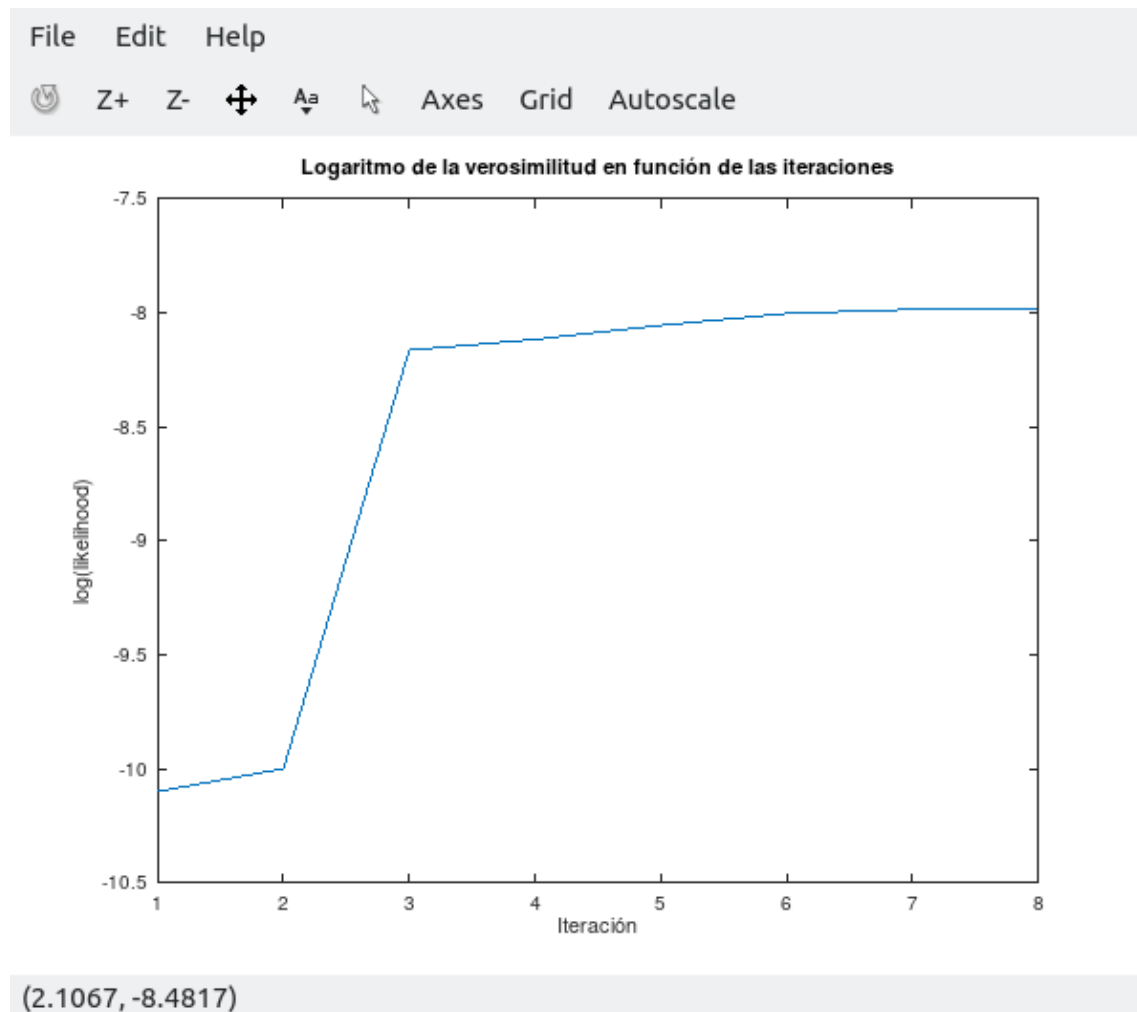


Figura 8: Verosimilitud en función de las iteraciones.

Notesé que hubieron 6 iteraciones y en el gráfico aparecen 8. Esto se debe a que para iniciar la iteración se tuvo que inicializar la verosimilitud para 2 valores previos que no corresponden a valores obtenidos de la verosimilitud.

Test

```

1 test = [a_test;o_test;u_test];

```

```

2 test_real_label = [ones(length(a_test),1);2*ones(length(o_test),1);3*ones(length(
3   u_test),1)];
4
5 for k=1:length(test)
6     disc(k,1) = discriminante(test(k,:), ma, cova, proba);
7     disc(k,2) = discriminante(test(k,:), mo, covu, probu);
8     disc(k,3) = discriminante(test(k,:), mu, covu, probu);
9     [a,b] = max(disc(k,:));
10    test_label(k) = b;
11 end
12
13 % Gráfico
14 figure()
15 hold on
16
17 % Medias
18 plot(ma(1),ma(2), 'rx','MarkerSize',20);
19 plot(mo(1),mo(2),'x','color',myGreen,'MarkerSize',20);
20 plot(mu(1),mu(2), 'bx','MarkerSize',20);
21
22 % Puntos encontrados
23 plot(test(:,1).*(test_label==1)', test(:,2).*(test_label==1)', 'ro');
24 plot(test(:,1).*(test_label==2)', test(:,2).*(test_label==2)', 'o', 'color',
25       myGreen);
26 plot(test(:,1).*(test_label==3)', test(:,2).*(test_label==3)', 'bo');
27
28 % Puntos reales
29 plot(test(:,1).*(test_real_label==1), test(:,2).*(test_real_label==1), 'rx');
30 plot(test(:,1).*(test_real_label==2), test(:,2).*(test_real_label==2), 'x', 'color',
31       myGreen);
32 plot(test(:,1).*(test_real_label==3), test(:,2).*(test_real_label==3), 'bx');
33
34 ErrorRatio = sum(test_label'==test_real_label)*100/length(test_label)
35
36 title(['Test con ErrorRatio de ' num2str(ErrorRatio,4) '%'])

```

1 ErrorRatio = 95.556

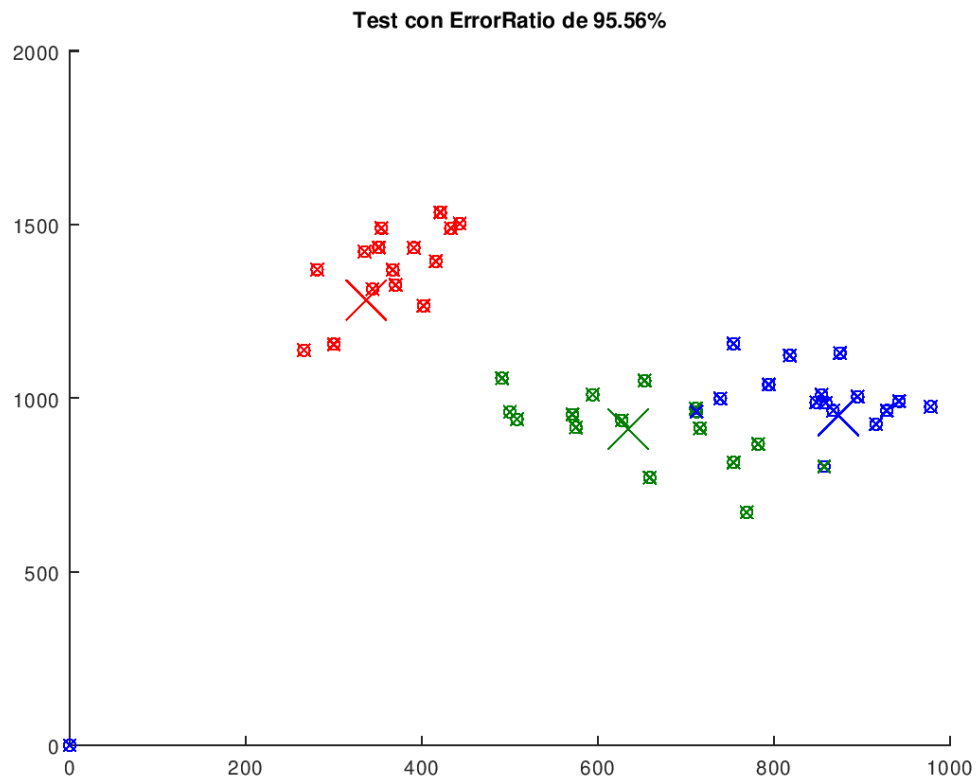


Figura 9: Puntos de test clasificados con un 95,56 % de precisión.

A. Funciones auxiliares

A.a. discriminante.m

Listing 1: discriminante.m

```

1 % Simplificación de la formula completa
2 %  $-\frac{1}{2} (x-\mu)' \text{inv}(\text{cov}) (x-\mu) - \frac{d}{2} \log(2\pi) - \frac{1}{2} \log(\det(\text{cov})) + \log(\text{prob})$ 
3
4
5 function discr = discriminante(x, mu, covarianza, prob)
6     variable = -1/2 .* (x-mu) * inv(covarianza) * (x - mu)';
7     constante = -log(det(covarianza)) + prob;
8
9     discr = variable + constante;
10 end

```

A.b. estim_cov.m

Listing 2: estim_cov.m

```

1
2 function covarianza = estim_cov(x, mu, res)
3     n = length(x);
4     covarianza = zeros(2,2);
5
6     for i = 1:n

```

```

7      covarianza = covarianza + (x(i,:)-mu)'*(x(i,:)-mu)*res(i);
8  end
9      N = sum(res);
10
11     covarianza = covarianza/N;
12 end

```

A.c. estim_sigma.m

Listing 3: estim_sigma.m

```

1
2 function sigma = estim_sigma(x, mu, N)
3     n = length(x);
4     sigma = zeros(2,2);
5
6     for i = 1:n
7         if(x(i,:)~=0)
8             sigma = sigma + (x(i,:)-mu)'*(x(i,:)-mu);
9         end
10    end
11
12    sigma = sigma/N;
13 end

```

A.d. inicializacion.m

Listing 4: inicializacion.m

```

1 %% Función inicializadora de EM y KMEANS
2
3 function [m1,m2,m3,cov1,cov2,cov3,prob1,prob2,prob3] = inicializacion(train,
4     indicador, n_clases)
5
6     n2 = n_clases; n3 = 2*n_clases;
7
8     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9     %% Inicialización supervisada %%
10    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11    if(isnumeric(indicador)) % Si recibo un número que indique cuántos puntos usar:
12        m1 = mean(train(1:indicador,:));
13        m2 = mean(train(n2+1:(n2+indicador),:));
14        m3 = mean(train(n3+1:(n3+indicador),:));
15
16        cov1 = estim_sigma(train(1:indicador,:), m1, indicador);
17        cov2 = estim_sigma(train(n2+1:(n2+indicador),:), m2, indicador);
18        cov3 = estim_sigma(train(n3+1:(n3+indicador),:), m3, indicador);
19
20        prob1 = 1/3;
21        prob2 = 1/3;
22        prob3 = 1/3;
23
24    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25    %% Inicialización no supervisada %%
26    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
27    else
28        m1=0; m2=0; m3=0; cov1=0; cov2=0; cov3=0; prob1=0; prob2=0; prob3=0;
29        ind = indicador(1); % Elimino el string para identificar
30
31        % Calculo la media global de los puntos de inicialización
32        mu = mean([train(1:ind,:); train(n2+1:(n2+ind),:); train(n3+1:(n3+ind),:)]);
33        train1 = train(1:ind,:); train2 = train(n2+1:(n2+ind),:); train3 = train(n3+1:(
34            n3+ind),:);

```

```

33
34 % Uniformizo la nube de datos a un circulo
35 train = [train1; train2; train3];
36 train_sinoff = train-mu;
37 mx = max(train_sinoff(:,1)); mix = min(train_sinoff(:,1));
38 train_unif(:,1) = train_sinoff(:,1)./(mx-mix)*2;
39 mx = max(train_sinoff(:,2)); mix = min(train_sinoff(:,2));
40 train_unif(:,2) = train_sinoff(:,2)./(mx-mix)*2;
41
42 [aux, index] = max(vecnorm(train_unif));
43 ang_aux = atan2(train_unif(2,index), train_unif(1,index));
44 train_unif = [cos(ang_aux) -sin(ang_aux); sin(ang_aux) cos(ang_aux)]*train_unif
45     '
46 train_unif = train_unif';
47
48 ang_train = atan2d(train_unif(:,2), train_unif(:,1));
49
50 lim1 = [+90; -30]; % [Límite superior; Límite inf]
51 lim2 = [-30; -150];
52 lim3 = [+90; -150];
53
54 % Genero los arreglos booleanos
55 bool_clase1 = (ang_train<=lim1(1)) & (ang_train>lim1(2));
56 bool_clase2 = (ang_train<=lim2(1)) & (ang_train>lim2(2));
57 bool_clase3 = (ang_train>=lim3(1)) | (ang_train<lim3(2));
58
59 n1 = sum(bool_clase1(:,1)!=0);
60 n2 = sum(bool_clase2(:,1)!=0);
61 n3 = sum(bool_clase3(:,1)!=0);
62
63 maxn = 4;
64
65 while((n1 < maxn || n2 < maxn || n3 < maxn) && lim1(1) < 180)
66     lim1 += 15; lim2 += 15; lim3 += 15;
67
68     bool_clase1 = (ang_train<=lim1(1)) & (ang_train>lim1(2));
69     bool_clase2 = (ang_train<=lim2(1)) & (ang_train>lim2(2));
70     bool_clase3 = (ang_train>=lim3(1)) | (ang_train<lim3(2));
71
72     n1 = sum(bool_clase1(:,1)!=0);
73     n2 = sum(bool_clase2(:,1)!=0);
74     n3 = sum(bool_clase3(:,1)!=0);
75
76     if((n1<maxn || n2 <maxn || n3<maxn)&& lim1(1)==180)
77         maxn -= 1;
78         lim1 -= 90; lim2 -= 90; lim3 -= 90;
79     end
80 end
81
82 % [lim1 lim2 lim3]
83
84 % Separo las muestras con rectas de 120
85 clase1 = train.*bool_clase1;
86 clase2 = train.*bool_clase2;
87 clase3 = train.*bool_clase3;
88
89 % [clase1 clase2 clase3]
90
91 % figure
92 % hold on
93 % plot(clase1(:,1), clase1(:,2),'r.', 'MarkerSize', 10)
94 % plot(clase2(:,1), clase2(:,2),'b.', 'MarkerSize', 10)
95 % plot(clase3(:,1), clase3(:,2),'g.', 'MarkerSize', 10)
96 % plot(mu(1),mu(2),'kx','MarkerSize',20)
97
98 % Media
99 ma = sum(clase1)./sum(clase1(:,1)!=0);

```



```

99 mo = sum(clase2)./sum(clase2(:,1)!=0);
100 mu = sum(clase3)./sum(clase3(:,1)!=0);
101
102 % Corrijo las clases
103 [aux, pos_max] = max([ma(1);mo(1);mu(1)]);
104 [aux, pos_min] = min([ma(1);mo(1);mu(1)]);
105 % [aux, pos_max] = max(vecnorm([ma;mo;mu]));
106 % [aux, pos_min] = min(vecnorm([ma;mo;mu]));
107 aux_clase1=clase1;
108 aux_clase2=clase2;
109 aux_clase3=clase3;
110 if(pos_max==2)
111     clase1 = aux_clase2; m1 = mo;
112     if(pos_min==3)
113         clase2 = aux_clase1; m2 = ma;
114         clase3 = aux_clase3; m3 = mu; %puts('2 1 3');
115     else
116         clase2 = aux_clase3; m2 = mu;
117         clase3 = aux_clase1; m3 = ma; %puts('2 3 1');
118     end
119 elseif(pos_max==3)
120     clase1 = aux_clase3; m1 = mu;
121     if(pos_min==1)
122         clase2 = aux_clase2; m2 = mo;
123         clase3 = aux_clase1; m3 = ma; %puts('3 2 1');
124     else
125         clase2 = aux_clase1; m2 = ma;
126         clase3 = aux_clase2; m3 = mo; %puts('3 1 2');
127     end
128 elseif(pos_max==1)
129     % Ya está bien
130     clase1 = aux_clase1; m1 = ma;
131     if(pos_min==2)
132         clase2 = aux_clase3; m2 = mu;
133         clase3 = aux_clase2; m3 = mo; %puts('1 3 2');
134     else
135         % Están bien
136         clase2 = aux_clase2; m2 = mo;
137         clase3 = aux_clase3; m3 = mu; %puts('1 2 3');
138     end
139 end
140
141
142 % Covarianza
143 cov1 = estim_sigma(clase1, m1, ind);
144 cov2 = estim_sigma(clase2, m2, ind);
145 cov3 = estim_sigma(clase3, m3, ind);
146
147 % Probabilidades (ocurrencias/totales)
148 prob1 = sum(clase1(:,1)!=0)/length(clase1);
149 prob2 = sum(clase2(:,1)!=0)/length(clase2);
150 prob3 = sum(clase3(:,1)!=0)/length(clase3);
151
152 % axis tight
153 % plot(m1(1),m1(2),'rx','MarkerSize',10)
154 % plot(m2(1),m2(2),'bx','MarkerSize',10)
155 % plot(m3(1),m3(2),'gx','MarkerSize',10)
156 end
157 end

```

A.e. responsabilidad.m

Listing 5: responsabilidad.m

```

1
2 function [res] = responsabilidad(pos, x, probs, mus, covs)

```

```
3  num = probs(pos) * mvnpdf(x, mus(pos,:), covs(pos));
4
5  den = 0;
6  for i = length(mus)
7      den += probs(i) * mvnpdf(x, mus(i), covs(i));
8  end
9
10 res = num/den;
11 end
```