Práctica de redes competitivas

Vamos a utilizar redes competitivas con topología para nuestro ejemplo de datos sobre zonas residenciales.

1. Tratamiento mínimo

Usamos netlab. Nos planteamos un agrupamiento de los datos residenciales, imponiendo una cierta topología entre las clases. Vamos a proponer 6 clases.

1. Cargamos los datos

2. Creamos la red con `redt = som(13, [3, 2]);` donde hemos elegido un patrón de 3 por 2 (netlab sólo trabaja en dos dimensiones)

3. Hacemos 50 pasadas como mínimo, pudiendo plantearnos hasta 200, con los comandos:

```
options(1) = 1;
options(6) = 1;%Ajuste tras cada pasada completa.
%Poner a 0 si la dimensión de la entrada es muy alta

options(14) = 50;%Número de iteraciones
options(17) = 1;%Distancia de vecindad inicial
options(15) = 0;%Distancia de vecindad final
options(18)=0.05; %Tasa de ajuste inicial (por caso)
options(16)=0.01; %Tasa de ajuste final
redt = somtrain(redt, options, p);
```

   y vemos los centroides con `pesos = sompak(redt)` Cada fila se corresponde con una categoría y cada columna con una variable. Expresa resumidamente las diferencias entre clases.

2. Afinado de ajuste

Ahora vamos a tener en cuenta los datos de salida en el agrupamiento y a elegir con más cuidado los parámetros que hemos cogido antes de manera irreflexiva.

1. Escalado de los datos. Antes hemos cogido los datos simplemente llevados a una escala [0,1]. Otra posibilidad es la normalización a media 0 y varianza 1 (`zscore`)

2. Variables elegidas. Hemos cogido las variables elegidas sin más. ¿Necesitamos todas? ¿Son directamente esas las que dan mejor agrupamiento? ¿Y la variable de salida? Una posibilidad fácil es concatenar variables. Si una variable aparece varias veces, es una manera trucada de aumentar su importancia.

3. Cantidad de grupos. ¿Por qué dos por tres? Existen dos posibilidades sensatas:

   a) Poner una cantidad claramente elevada de grupos y luego reagruparlos

   b) Hacer un agrupamiento previo que nos dé una idea

4. Comprobación: sea cuál sea la combinación que probemos, hay medidas que nos pueden indicar si la cosa va bien. A continuación se comentan algunas, aunque no son directas de obtener:

   a) El error de topología: fracción de puntos cuyo segundo mejor no es adyacente. Idealmente debería ser 0

   b) Error de cuantización: la distancia media entre cada ítem y su «representante» Idealmente debería ser baja.

   c) Correlación entre la distancia entre puntos y la distancia entre sus «representantes» Idealmente debería ser 1

5. Los grupos, ¿qué representan? Podemos seleccionar aquellos pesos donde la diferencia sea mayor.

2.1. Caso ejemplo

Vamos a tomar varias decisiones de entre lo comentado anteriormente:

- Vamos a añadir la variable de salida.

- Haremos un tanteo con el agrupamiento jerárquico. De él sacaremos una idea de cantidad de grupos.

- Tras ajustar la red comprobaremos la calidad de su agrupamiento
- Probaremos la red con algún valor para ver qué pasa
- Crearemos una red sólo para entradas que podremos usar para hacer predicciones. Probaremos alguna.

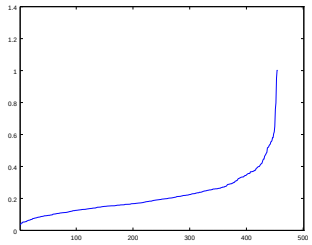
Veamos los comandos:

1. Cargar los datos, separar una parte para pruebas y juntar entradas con salida

```
[p1,t1,nin,nout,ndata]=datread('casas.dat');
orden=randperm(506);
pt=p1(orden,:);
tt=t1(orden);
pprueba=pt(1:50,:);
tprueba=tt(1:50);
p=pt(51:end,:);
t=tt(51:end);
datos=[p t];
```

2. Probar un agrupamiento jerárquico para estimar una cantidad de clases. Miramos como va variando la distancia entre grupos.

```
arbol=linkage(datos,'single','euclidean');%Tantear jerárquico
plot(arbol(:,3));%Distancias entre nodos agrupados
```



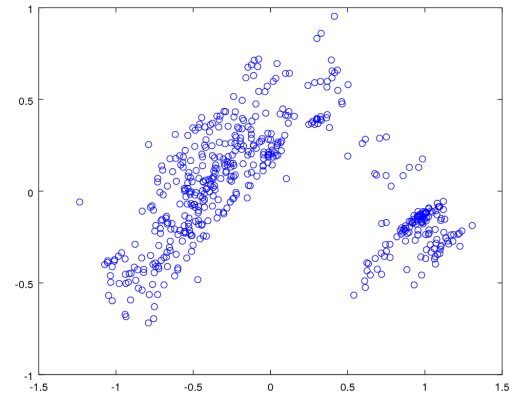
Viendo lo anterior, podemos quizá admitir hasta, por ejemplo, 420 uniones, lo que nos dejaría con 36 grupos. Toma tu propia decisión.

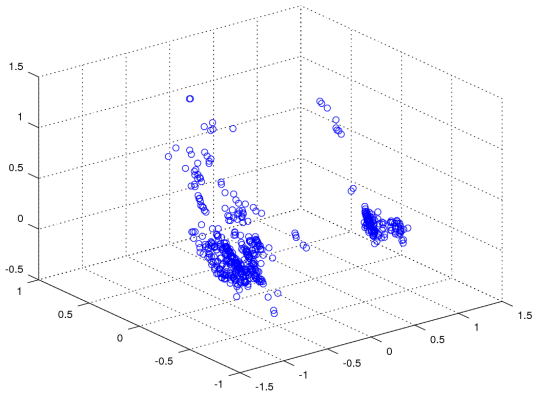
3. Otra posibilidad: dibuja un gráfico tridimensional con los primeros componentes principales (más de unas pocas dimensiones no se visualizan bien) y mira cuántos grupos ves. Tira hacia arriba, porque hay dimensiones que no estás viendo. Podría ser algo como:

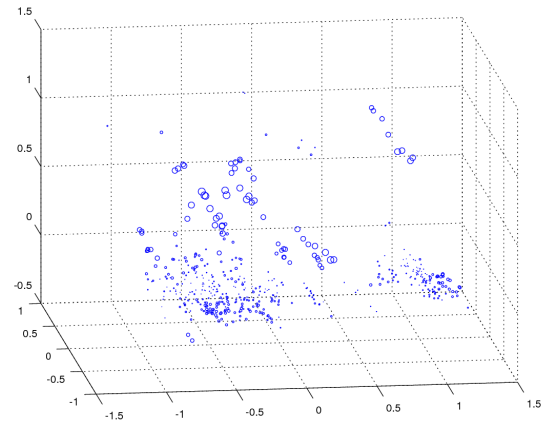
```
[coorprin,coord]=princomp(datos);
scatter(coord(:,1),coord(:,2))
scatter3(coord(:,1),coord(:,2),coord(:,3))
```

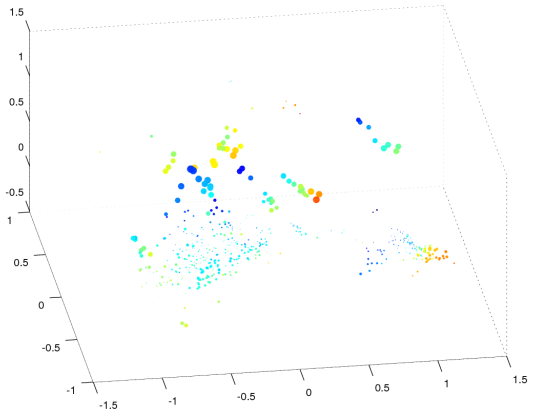
```
scatter3(coord(:,1),coord(:,2),coord(:,3),coord(:,4)*10)
scatter3(coord(:,1),coord(:,2),coord(:,3),coord(:,4)*10,coord(:,5),'f
```

Que nos darían (recuerda rotar los gráficos tridimensionales para apreciar bien las posiciones):



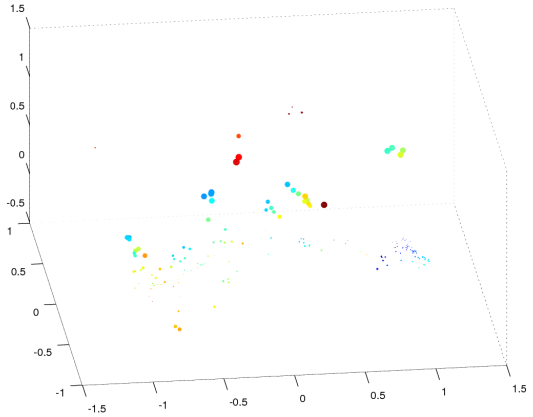


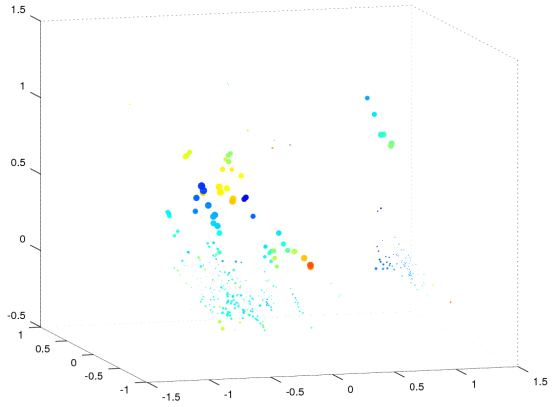


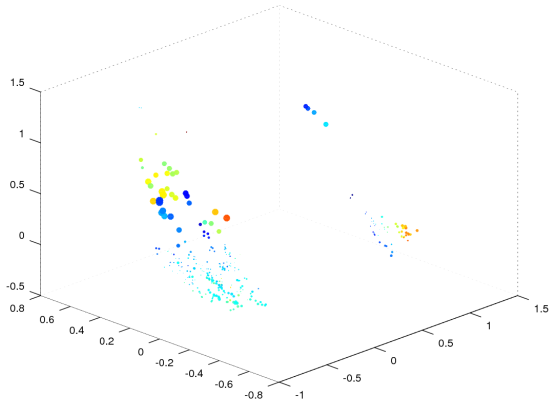


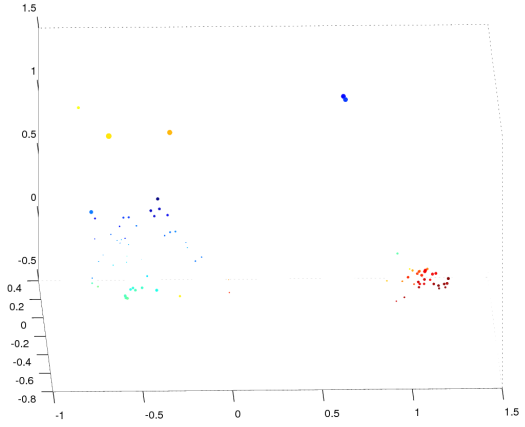
Incluso podemos acceder a una sexta coordenada rebanando el espacio:

```
pca6=coord(:,6);
m6=min(pca6);
max6=max(pca6);
cor6(1)=m6;
ran6=max6-m6;
sal6=ran6/5;
for i=2:6
        cor6(i)=m6+(i-1)*sal6;
end
for i=2:5
        reban=find(abs(pca6-cor6(i))<sal6);
        figure;scatter3(coord(reban,1),coord(reban,2),coord(reban,3),
end
```









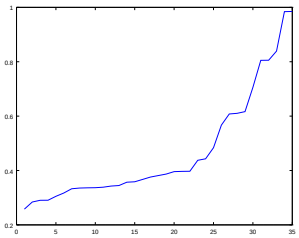
4. Crear una red de agrupamiento, con un número de clases similar (por ejemplo, 6 por 6 dan 36):

```
redtot=som(14,[6 6]);
options(1) = 1;
options(6) = 1;%Ajuste tras cada pasada completa
```

```
options(14) = 100;%Número de iteraciones
options(17) = 4;%Vecindad inicial
options(15) = 0;%Vecindad final
options(18)=0.1; %Tasa de ajuste inicial
options(16)=0.01; %Tasa de ajuste final
redtot = somtrain(redtot, options, datos);
pesostot = sompak(redtot);
```

5. Si esa cantidad de grupos nos parecen muchos, podemos reagrupar las clases, creando «superclases» Para este agrupamiento, las entradas son los centroides obtenidos en el paso anterior. Podemos hacer con ellos un agrupamiento jerárquico.

```
arbolpesos=linkage(pesostot,'single','euclidean');%vamos a reagruparl
plot(arbolpesos(:,3));
```



Si, por ejemplo, cortamos en el 24º agrupamiento nos quedarían 12 grupos, que podríamos ajustar con:

```
redgran=som(14,[4 3]);
options(1) = 1;
options(6) = 1;%Ajuste tras cada pasada completa
options(14) = 50;%Número de iteraciones
options(17) = 2;%Vecindad inicial
options(15) = 0;%Vecindad final
options(18)=0.1; %Tasa de ajuste inicial
options(16)=0.01; %Tasa de ajuste final
redgran = somtrain(redgran, options, pesostot);
pesosgran = sompak(redgran);
```

6. Vamos a comparar los agrupamientos, usando los conjuntos de prueba que habíamos separado:

a) el dado por las 36 clases

b) el de las superclases obtenidas por agrupamiento de éstas.

Miraremos la relación entre la distancia al grupo propio con la distancia al segundo:

```
prueba=[pprueba tprueba];
[dista,grupa]=somfwd(redtot,prueba);
dista=sort(dista,2);
[distb,grupb]=somfwd(redgran,pesostot(grupa,:));
distb=sort(distb,2);
plot(dista(:,1),dista(:,2),'xr',distb(:,1),distb(:,2),...
'+b',dista(:,1),dista(:,1))
```

El gráfico anterior nos da pistas sobre el error de cuantización y la fiabilidad de la clasificación. Si queremos echarle un vistazo también al error de topología y a la correlación de distancias, podemos hacer, por ejemplo, para la primera red:

```
[dista,grupa]=somfwd(redtot,prueba);
[dista,quienes]=sort(dista,2);
[fils,cols]=ind2sub([6 6],quienes);
grupdismin=[fils(:,1) cols(:,1)];
grupdis2=[fils(:,2) cols(:,2)];
disgrup1=abs(grupdismin-grupdis2);
diffc=max(disgrup1,[],2);
hist(diffc) %Gráfica de problemas de topología. Idealmente todos los
length(find(diffc>1))/length(diffc) %Error de topología. Idealmente 0
distpuntos=dist2(prueba,prueba);
distgrupos=dist2(grupdismin,grupdismin);
dispun1=triu(distpuntos,1,'pack');
disgru1=triu(distgrupos,1,'pack');
corr(dispun1,disgru1) %Correlación de distancias entre puntos y entr
plot(dispun1,disgru1,'x') %Gráfica de relación entre ambas distancias
```

7. Vamos a qué tal es representado un punto particular por el centroide correspondiente elegido por la red, también en ambos casos, fijándonos en la variable de salida:

```
%Probamos con, por ejemplo, el punto 25 de la prueba
salreal=tprueba(25)
salreda=pesostot(grupa(25),14)
salredb=pesosgran(grupb(25),14)
```