Reducción de dimensión

Vamos a ensayar algunos métodos utilizando unos datos de espectrometría de masas de muestras potendialmente cancerígenas. Nos dan el espectro caracterizado con 10 000 puntos para cada muestra. Los datos están comprimidos en el archivo adjunto.

Para cargar los datos utilizamos el comando:

```
load arcene;
```

Nos aparecerán la variable `datos`. Los datos son 100 muestras, cada una con sus 10000 puntos de espectro.

Una primera cosa que podemos hacer es quitar las variables que sean claramente irrelevantes. Desde un punto de vista no lineal, estamos hablando de información mutua.

La toolbox de estadística tiene algunos algoritmos para crear conjuntos de variables de dimensión reducida. Aparte usaremos otra toolbox específica: "*Dimensionality Reduction*"; tiene muchos, y alguna función auxiliar facilitadora.

¡Ojo, cuidado! Con 100 puntos, sacar conclusiones por encima de 100 dimensiones no tiene sentido.

Proyección en componentes principales

Es un método lineal, pero suele ayudar bastante. Tenemos varios comandos para ello. El de la estadística es el que da más información, pero no es el más fácil de aplicar, así que miraremos la información en uno y después usaremos el otro.

1. Obtengamos los 10 000 ejes ortogonales ordenados por varianza de mayor a menor. En realidad no serán 10 000, porque no puede calcular más de 100 si sólo tiene 100 puntos. `[coorprin,nuevaentrada,varianza]=pca(datos);`
   Esto nos da las coordenadas de los versores principales, las coordenadas de los puntos originales en esos ejes y la varianza en cada eje.
2. Para comparar varianzas mejor, vamos a ponerlas en tanto por uno relativo al total: `porunoacumvar=cumsum(varianza)/sum(varianza);`
3. La idea es coger unos cuantos que tengan claramente más varianza que los demás y que entre todos sumen una fracción muy alta de la varianza total. Elige hasta donde veas que ya sólo se gana marginalmente (no tiene por qué ser claro).
4. Lo que nos interesa es cuántos componentes queremos. Si por ejemplo queremos coger hasta el 15, sería:

   ```
   nuevos=compute_mapping(datos,'PCA',15);
   ```

Proyección de Sammon

Esta es una proyección no lineal y originalmente no supervisada. El uso más fácil es con la toolbox especializada en reducción de dimensión.

1. Tenemos que elegir en cuántas dimensiones proyectamos. Hay tres opciones:
   - Ir probando. Puede resultar pesado.
   - Elegir la misma cantidad que ha salido en el análisis anterior.
   - Elegir con base a una estimación de dimensión independiente.

     Si queremos esto último, el comando es (con uno de los métodos que trae la tool-

box):

```
ndims = round(intrinsic_dim(datos, 'MLE'));
```

No tiene mucho sentido aceptar dimensiones por encima del número de puntos y además puede liar a los algoritmos. En vez de MLE puedes probar GMST

2. Hacemos la proyección:

```
nuevasent2 = compute_mapping(datos, 'Sammon', ndims);
```

Proyección SNE

Este es otro caso de proyección no lineal basada exclusivamente en los datos. Se parece bastante al anterior, pero podemos probarlo. Sólo está en la toolbox de reducción de dimensión. Es todo igual que en el caso anterior pero cambiando *Sammon* por *t-SNE*