VNIVERSITAT DE VALÈNCIA

MASTER DE INGENIERÍA BIOMÉDICA.

Guía para la realización de los trabajos (III)

Preprocesado de datos: outliers; PCA; clustering; SOM.

Modelos lineales

Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

VNIVERSITAT DE VALÈNCIA

Datos incompletos; atípicos, ...

En primer lugar hay que cargar el conjunto de datos (la manera más sencilla es siguiendo la secuencia File-Import Data). Una vez que tengamos el conjunto de datos hay que determinar una primera selección de las variables de forma sencilla: aquellas que hay pocos datos son eliminadas. Este criterio depende del criterio del analista de datos y dependerá de la proporción entre número de patrones completos e incompletos.

El siguiente paso es determinar posibles outliers en las variables. Un criterio que se sigue es eliminar aquellos datos alejados más de 3 desviaciones estándar del valor medio (usar las instrucciones mean, std y find!!!). SI SE TIENEN POCOS DATOS NO SE SUELE APLICAR

Una forma de rellenar es comparar patrones entre sí; si se tienen dos patrones con valores parecidos de las variables.....podemos poner como valor que falta el que se tiene!!!; si, por ejemplo, dos pacientes con edades parecidas, mismo sexo, parecida evolución temporal del fármaco nos falta la dosis inicial en uno de ellos podemos poner la del otro!!!!!. Puede ser computacionalmente intensivo pues hay que comparar los patrones que se tienen entre sí (evidentemente se comparan las variables que se tienen). La instrucción implicada en Matlab es norm (mira la ayuda!!!! y pregunta si no lo ves claro!!!).

Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

VNIVERSITAT DE VALÈNCIA

Transformaciones de los datos.

Función	Misión	Matlab
$y_k = \frac{x_k - m_x}{\sigma_x}$	La nueva variable tiene valor medio 0 y varianza 1.	y=zscore(x); o bien usa las instrucciones mean y std
$y_k = \frac{x_k - x_{mi}}{x_{ma} - x_{mi}} \cdot [y_{ma} - y_{mi}] + y_{mi}$	La variable está en [y_{mi}, y_{ma}]	directa de implementar!!!; necesitas las instrucciones min y max
$y_k = \tanh\left(\frac{x_k - m_x}{\sigma_x}\right)$	La variable está en [-1,1]	Usa la primera instrucción y, posteriormente, usa na instrucción tanh

Aquí m_x y σ_x son el valor medio y la desviación estándar de la variable x; x_{mi} y x_{ma} son, respectivamente, el valor mínimo y máximo de dicha variable. La más usada es la primera (estandarización)

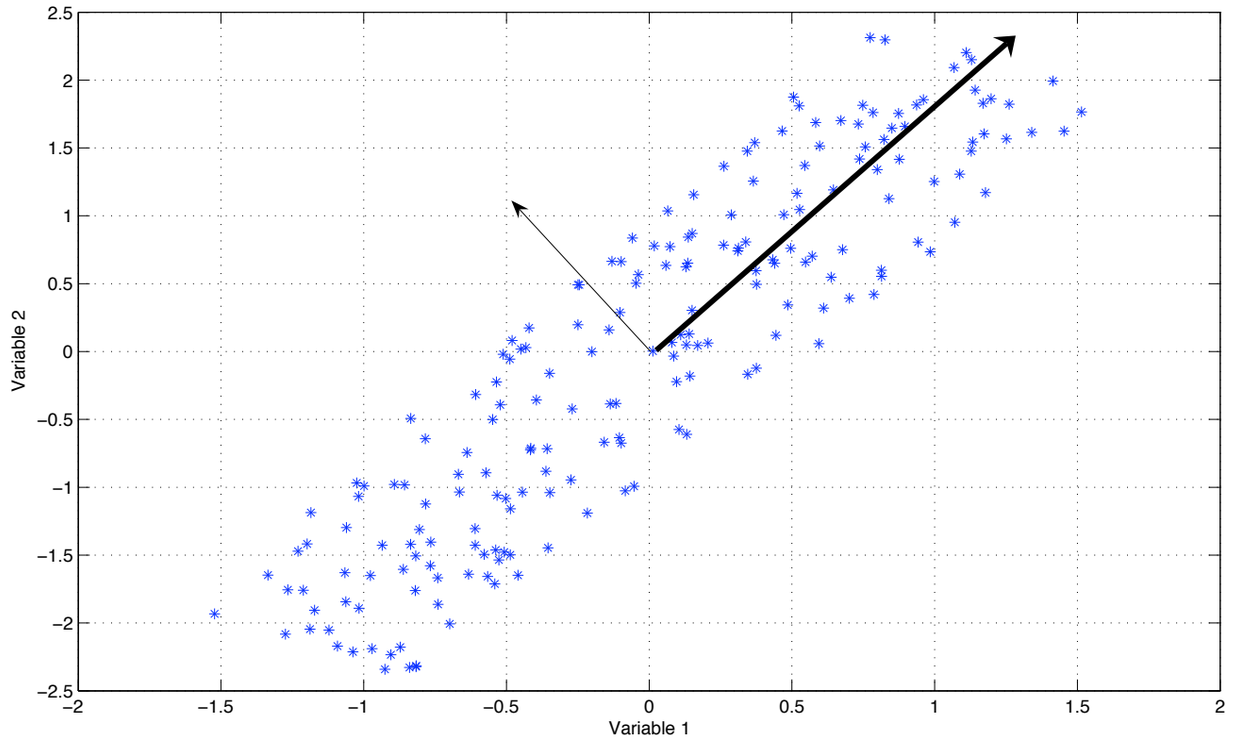
En cuanto a las variables discretas se puede usar cualquier codificación siempre que se evite el 0 (cuando lleguemos a modelos no lineales se comentará este hecho). A modo de ejemplo una codificación del tipo 0/1 se suele sustituir por -1/1.

Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

VNIVERSITAT DE VALÈNCIA

Análisis de Componentes Principales, PCA

Estamos interesados en encontrar las direcciones de máxima dispersión porque, en principio, SON LAS QUE APORTAN MAYOR INFORMACIÓN.



Pasos para realizar una PCA.

1. Eliminamos el valor medio de los vectores que se tienen (1)
2. Calculamos la matriz de autocorrelación (2)
3. Calculamos la diagonalización de esa matriz (3)
4. Determinamos los vectores propios asociados al mayor autovalor (4)
5. Proyectamos cada vector en esos valores propios (5)

Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

VNIVERSITAT DE VALÈNCIA

Análisis de Componentes Principales, PCA. MATLAB

El siguiente programa determina las direcciones principales de la matriz X; hay que fijarse en el valor de los valores propios que definen "la elongación" de los datos. A mayor diferencia entre ellos significa que hay direcciones "privilegiadas" mientras que, si son todos iguales, tenemos una distribución circular (esférica, hiperesférica dependiendo de la dimensionalidad del problema) de los datos

```
clear
clc
close all
[a,b]=size(X);
m=mean(X);
C=zeros(b);
for t=1:a
C=C+((X(t,:)-m)')*(X(t,:)-m);
end
C=(1/a)*C;
[vectores,valores]=eig(C);
```

Prueba el ejemplo anterior con las siguientes matrices: a)X=randn(100,3); b) CC=zeros(100,3); CC(:,1)=10; CC(:,2)=5; CC(:,3)=1; X=CC.*randn(100,3); intenta interpretar los resultados (si no puedes PIDE AYUDA!!!!!).

Una medida que se utiliza para conocer la importancia de cada dirección principal es determinar el cociente de cada valor propio por la suma de todos ellos (se tiene entonces un valor entre 0 y 1 en la nueva medida).

Si al analizar tus datos tienes valores propios con esta medida cercana a la unidad tienes que fijarte en los vectores propios.....(¿?)

Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

VNIVERSITAT DE VALÈNCIA

Clustering (I)

El objetivo de estos métodos es determinar grupos dentro de los datos que quedan definidos por lo que se conoce como centroides o prototipos. Aquí se tienen dos misiones: a) analizar las características de los prototipos y b) determinar qué pacientes pertenecen a cada uno de los grupos

Existen gran cantidad de algoritmos de clustering; en una primera aproximación se estudian dos: el que se conoce como "duro" (HCM) y el "borroso" (FCM). La diferencia estriba en cómo asignamos los patrones a los diferentes grupos; o bien pertenece o no (HCM) o se plantea un grado de pertenencia a cada grupo (FCM)

La instrucción en MATLAB es **[centros, U, J] = fcm(datos, n_clusters,opciones).**

Aquí datos es nuestra matriz de datos; n_clusters es el número de grupos que suponemos hay (ES NECESARIO VARIAR ESTE PARÁMETRO). En cuanto a opciones es un vector de 4 componentes en el que a) la primera componente define si queremos un clustering borroso (2, valor por defecto) o uno duro (valores por encima de 5; es el factor m que aparece en las transparencias de teoría b) máximo número de iteraciones, por defecto 100 c) mínima mejora en la función de coste entre iteraciones, defecto 1e-5, d) se muestra (1) ,o no (0), información por pantalla por defecto se tiene un valor de1.

Clustering (II). Ejemplo en Matlab

El fichero ejemplo_clustering.m genera dos grupos en los que escoge en número de puntos y la dispersión de cada uno de los grupos (prueba con grupos separados y solapados). Este programa genera un fichero matriz_datos que utilizaremos en las siguientes instrucciones

```
load matriz_datos
n_clusters=2;
grado=2;
nit=150;
prec=1e-5;
vis=0;
[centros, U, J] = fcm(Datos,n_clusters,[grado nit prec vis]);
```

Comprobar los resultados obtenidos (centros) y cómo se asignan los datos a los diferentes *clusters*. Varía los parámetros del algoritmo y analiza los resultados.

Debéis aplicar este algoritmo a vuestros datos variando el número de grupos y analizando los centros que se obtienen. Posteriormente observa qué patrones pertenecen a cada grupo.

Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

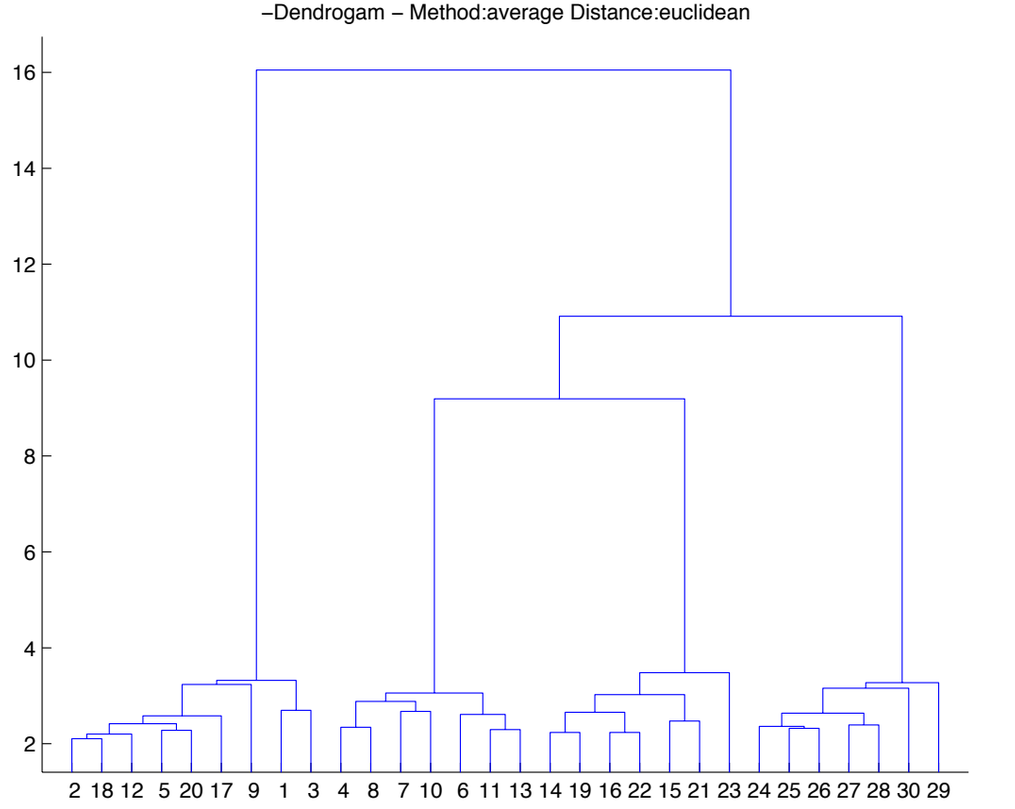
VNIVERSITAT DE VALÈNCIA

Clustering (III)

Existe otra aproximación al clustering que consiste en seguir **estrategias divisivas** (se parte de un único cluster y se van obteniendo los diferentes clusters mediante su división) **o de unificación** (en un principio se tienen tantos clusters como patrones y se van obteniendo clusters de forma iterativa).

El fichero **cluster_datos_EDA.m** implementa un ejemplo de esta aproximación. Se generan los datos y se van creando los clusters (estrategia divisiva) usando la instrucción **linkage** (analiza las diferentes posibilidades que tienes usando la ayuda de Matlab).

Finalmente se representa el **dendrograma** que muestra, de manera visual, cómo se van generando los diferentes grupos. En las siguientes líneas del programa tenéis una análisis estadístico de los diferentes clusters que vais obteniendo. Modifica el programa para adaptarlo a tus datos (¡RECUERDA QUE AGRUPAR NO ES LO MISMO QUE CLASIFICAR!



Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

VNIVERSITAT DE VALÈNCIA

Mapas Autoorganizados (SOM)

Para implementar estos modelos neuronales es necesario leer la práctica correspondiente que tenéis en la página web de la asignatura. El único problema es instalar la librería del SOM (*som toolbox*); para ello, simplemente, hay que añadir al *path* de Matlab (directorios de búsqueda de funciones de Matlab) el directorio donde se encuentra dicha librería. La secuencia es: *File-Set Path* y, a continuación, se escoge el directorio de la librería con la opción *Add with Subfolders*

El programa base que tienes que utilizar es el fichero SADC_Tema3_SOM_Colesterol que va junto con el fichero SADC_Tema3_SOM_Colesterol_Data. Tienes que modificar ese fichero para cargar tus datos. Realiza diferentes pruebas variando los parámetros del programa como son la función vecindad, la inicialización, el tipo de algoritmo, etc. USA LA AYUDA DE SOM_MAKE

Una vez que obtengas el mapa tienes que interpretar el mapa de componentes del SOM en las diferentes zonas que te marca el clustering del mapa

Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

VNIVERSITAT DE VALÈNCIA

Modelos lineales (I).

Las instrucciones básicas para problemas de predicción y modelización son **polyfit** (regresión univariante y polinómica); **regress** (regresión multivariante) y **robustfit** (regresión robusta). UTILIZA LA AYUDA DE MATLAB!!!

La única dificultad de estas instrucciones está en **regress**; hay que añadir una primera columna de unos (se supone que se tiene un término constante en el modelo). La forma de hacerlo es muy sencilla; si X es la matriz de entrada se realizan las siguientes instrucciones:

```
[a,b]=size(X);
XX=ones(a,b+1);
XX(:,2:end)=X;
```

Se ha desarrollado un programa modelos_lineales.m que implementa lo que se comenta en el fichero de texto modelos_lineales.pdf; ejecútalo por partes y comprueba los resultados obtenidos. Para ejecutar determinadas líneas sólo tienes que seleccionar con el ratón las que no quieres ejecutar y sigues la secuencia de menús Text-Comment

Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

VNIVERSITAT DE VALÈNCIA

Modelos lineales (II).

Cuando se tiene un problema de clasificación y se quiere resolver con una regresión logística la instrucción a utilizar es **glmfit**. Esta instrucción implementa lo que se conoce como *modelos lineales generalizados*. *Como siempre USA LA AYUDA DE MATLAB!!!!*

La forma de utilizarla es: **b=glmfit(X,y,'binomial','link','logit');** aquí X es nuestra matriz con las variables de entrada al modelo; y es un vector de 0/1 (vector columna); si tienes los datos codificados como -1/1 debes cambiar la codificación.

Una vez obtenido el modelo (los coeficientes b de la regresión logística) se pueden obtener las salidas de dicho modelo usando la instrucción:

yfit=glmval(b,X,'logit');

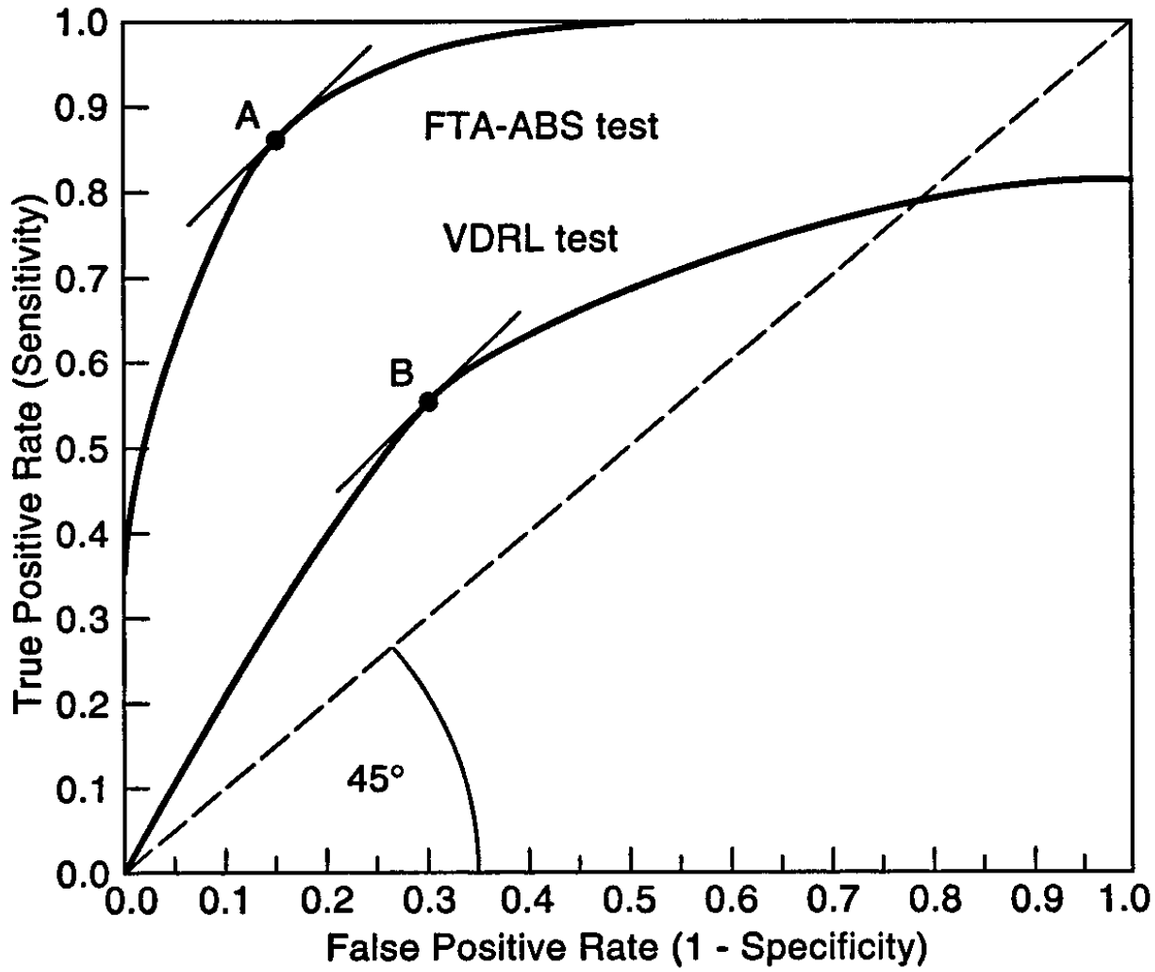
RECUERDA QUE SIEMPRE TIENES QUE ANALIZAR EL MODELO OBTENIDO.

Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

VNIVERSITAT DE VALÈNCIA

Evaluación de clasificadores (I)

Una vez que se tienen los diferentes modelos hay que evaluar su calidad. Existen muchos índices pero nosotros nos basaremos en el **area bajo la curva (A.U.C).** Para construirla se varía un parámetro (normalmente el umbral con el que vamos a discretizar la salida del modelo) obteniéndose diferentes valores de sensibilidad y especificidad para el modelo



Este índice puede ir de 0 a 1 pero el valor 0.5 corresponde a una clasificador aleatorio (1 es uno perfecto!).

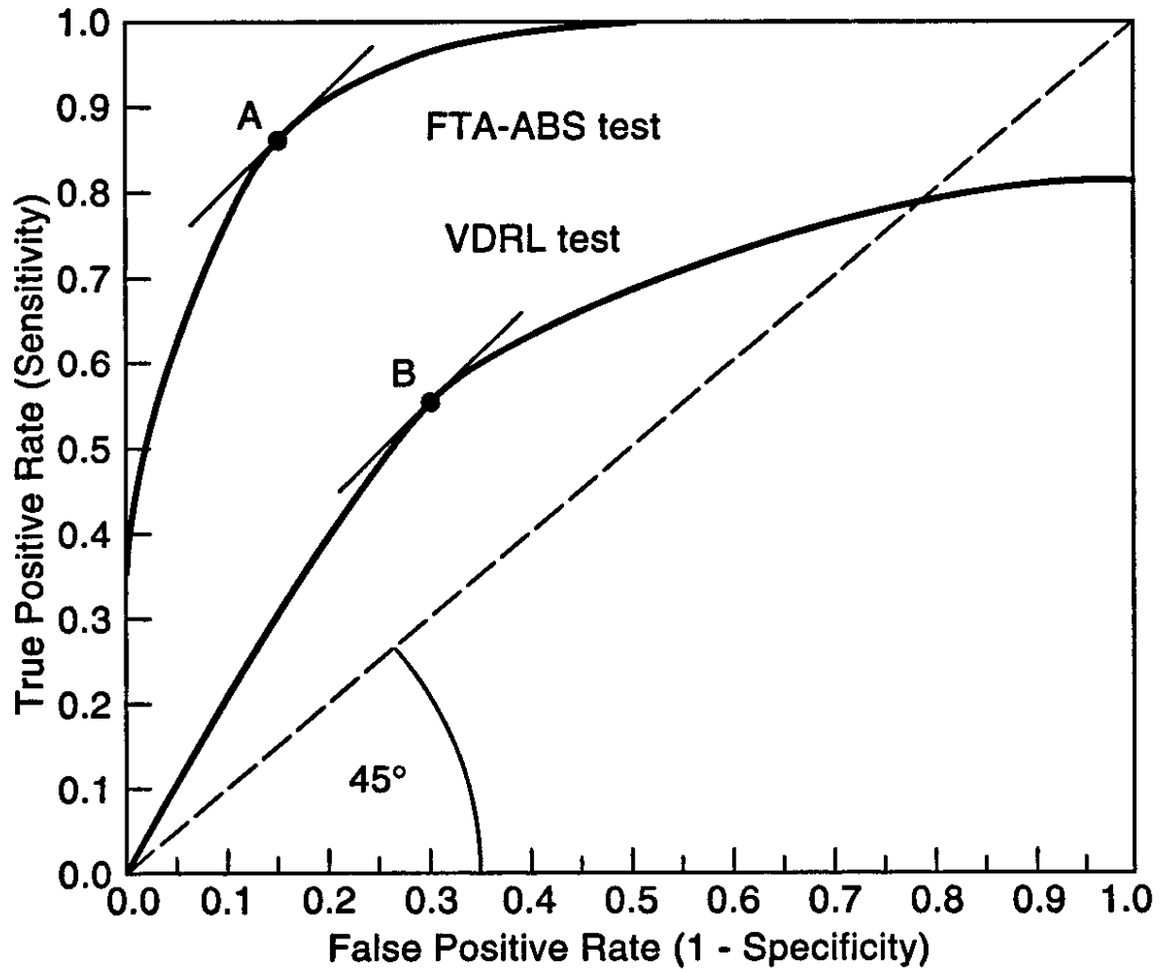
Hay que determinar la AUC para el conjunto de entrenamiento y para el de generalización, en el primer caso indica cómo de bien se ajusta el modelo a los datos y, en el segundo, lo bien que generaliza.

Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

VNIVERSITAT DE VALÈNCIA

Evaluación de clasificadores (II)

Una vez seleccionado el modelo mediante el índice AUC hay que determinar el valor que discretiza la salida del modelo. Este valor se corresponde al valor del umbral que proporciona el valor de sensibilidad, especificidad más cercano a la esquina (0,1) de la representación que proporciona el AUC (puntos A y B en la figura)



La función **roc_mib** proporciona todos los índices del clasificador así como los umbrales utilizados. Usa la ayuda de la función para obtener más información.

Una vez ejecutada esta función tienes que lanzar la función **plot_roc** (de nuevo mira la ayuda) para que te represente la curva ROC.

VNIVERSITAT DE VALÈNCIA

MASTER DE INGENIERÍA BIOMÉDICA.

Guía para la realización de los trabajos (III)

Preprocesado de datos: outliers; PCA; clustering; SOM.

Modelos lineales

Profesores: Emilio Soria y Antonio José Serrano, Dpto Ingeniería Electrónica, ETSE

VNIVERSITAT DE VALÈNCIA