Proyecto

Por entregar dos dias antes del examen de probabilidad y estadística; El proyecto se puede hacer en equipos de dos personas. Hay que entregar un pequeño reporte. Cuida también la redacción.

1. Clasificación de dígitos

Considera el conjunto *digitos* de dígitos escritos a mano (tamaño 16x16). Puedes leer el archivo con los siguientes comandos:

```
data<-scan("digitos")
m<-matrix(data,ncol=257,byrow=T)
```

El resultado es una matriz (1000x257) donde cada reglón corresponde a una imagen de un dígito: en la primera columna está el valor del dígito que está representado en la imagen y en las siguientes 256 columnas los valores de los pixeles de la imagen. Por ejemplo, para ver el primer imagen (rotado) puedes teclear:

```
image(matrix(m[1,-1],nrow=16,byrow=T), col=(0:255)/255)
```

El método de clasificación *k-vecino más cercano* consiste en buscar las k observaciones más cercanas a una observación (imagen) dada, y asignar a esta observación la categoría que más veces ocurre entre sus k vecinos.

Divide estos datos en dos grupos A y B con respectivamente 70 % y 30 % de las imágenes del arreglo m. Es recomendable conservar la proporción de cada dígito en ambos grupos.

Para diferentes valores de k, construye un clasificador basado en *k-vecino más cercano* usando A y calcula el error de clasificación (=porcentaje de observaciones mal clasificadas) usando el conjunto B. Grafica el error del clasificador usando las observaciones en el archivo B y el error del clasificador usando las observaciones en el archivo A contra k. Explica porque las dos curvas tienen una tendencia contraria. ¿Cuál elección de k te parece óptimo?

Considera todas las imagenes que representan 1: usa clustering jerárquica para detectar alguna(s) imagenes atípicas. Verifícalo visualmente.

Comandos útiles

`sample()`: para muestrear;

`knn()`: método del vecino más cercano (cargar primero *library(class)*);
`m[m[,1]==1,]`: para seleccionar todas los reglones que tienen el valor uno en la primera columna.

2. Clasificación basada en un modelo probabilístico

Considera Y una variable aleatoria binaria que indica si una persona tiene una enfermedad o no (codificado como 1 o 0) y X una variable continua que representa una medicíon particular de utilidad para diagnosticar la enfermedad.

En el ejemplo que estudiaremos a continuación, la enfermedad es *coronory heart disease*(CHD). Los datos están en
http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/datasets/SAheart.data;
una descripción se encuentra en
http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/datasets/SAheart.info..
Aúnque se obtiene por cada paciente varios tipos de mediciones, en lo que sigue nos concentramos en la medición *systelotic blood pressure (SBP)* (=X).

1. Encuentra una estimación de la densidad de SBP para el grupo que **no** tiene CHD, usando kernels. Visualiza el/los resultado(s).

   Divide las observaciones al azar en dos grupos: uno con la mayor parte de los datos (grupo A) y otro con las demás (grupo B). Grafica el ancho de banda h versus la verosimilitud ($= \prod_{x_i \in A} \hat{f}(x_i)$) de los datos de grupo A basada en la estimación de la densidad usando A, $\hat{f}(\cdot)$. También gráfica el ancho de banda h versus la verosimilitud de los datos de grupo B basada en la estimación de la densidad usando A. Argumenta que las gráficas deben tener una *tendencia* contraria.

   Probablemente tendrás problemas de estabilidad numérica. Eventualmente transforma la verosimilitud (por ejemplo tomando el logaritmo).

2. Usando la regla de Bayes se sabe que:

$$P(Y = 1|X = x) = \frac{P(X = x|Y = 1)P(Y = 1)}{P(X = x|Y = 1)P(Y = 1) + P(X = x|Y = 0)P(Y = 0)}.$$

   Esta probabilidad es muy informativa para formarse una idea que probable es que alguién tenga CHD en base de su valor de SBP.

Grafica una estimación de esta probabilidad (aposteriori) en función de x aproximando $P(X = x|Y = 1)$ y $P(X = x|Y = 0)$ con estimadores basados en kernels (como hecho en el incico anterior).

Comandos útiles
La función `density()` calcula el estimador de Kernel; para evaluar la estimación en un punto particular, x, usa:

```
d<-density( ...)
s<-smooth.spline(d) (genera una funcion de interpolacion)
predict(s,x)
```

Cuidado con el hecho que el resultado de `predict()` no es un escalar!

3. Clasificación de textos

Este ejercicio es sobre como usar el clasificador ingenuo Bayesiano para clasificar textos según su tema. Para entender el clasificador ingenuo Bayesiano, lee las secciones 1 y 2 del capítulo:
`http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf`

Nos limitamos al caso de clasificación binaria (i.e., tenemos nada más dos categorías: $Y = 0$ o $Y = 1$). Introducimos la variable binaria X_i para indicar si la palabra clave i aparece o no en un documento dado.

Para sacar y contar palabras en un texto y para obtener las probabilidades $P(X_i = 1|Y = y)$, usaremos el software *rainbow* bajo linux:
`http://www.cs.cmu.edu/ mccallum/bow/rainbow/`

Para seleccionar las palabras claves usaremos las X_i con mayor dependencia con Y: mediremos la dependencia entre Y y X_i a través de la Información Mutua.

Como textos de prueba se pueden tomar dos newsgroups del conjunto *20_newsgroups*; en la página de *rainbow* aparece la liga. Por supuesto puedes tomar tu propio conjunto de datos.

El resultado de este ejercicio debe ser un texto dirigido a un compañero de generación y que ilustra con ejemplos concretos como funciona el clasificador ingenuo Bayesiano en el contexto de la clasificación de textos.

J. Van Horebeek
Noviembre 2006