

# Práctica 1: Clasificación

Tratamiento de Datos  
Curso 2013–2014

19 de noviembre de 2013

## Introducción

En esta práctica analizaremos los métodos de regresión logística y las máquinas de vectores soporte utilizando datos generados artificialmente. A continuación, analizaremos el comportamiento de estos clasificadores utilizando una base de datos reales.

### 1. Análisis de un problema sintético bidimensional

Cargue los datos contenidos en el fichero `'datosP1.mat'`. El fichero contiene dos variables,  $x$  e  $y$ . La primera es una matriz de observaciones. Cada fila de la matriz es una observación bidimensional. La segunda contiene la categoría a la que pertenece cada observación (0 ó 1).

#### 1.1. Regresión logística: clasificador lineal

En primer lugar, analizaremos el comportamiento de un modelo de regresión logística.

1. Visualice las observaciones sobre el plano, utilizando un marcador diferente para los datos de cada categoría.
2. Utilizando la función `glmfit`, ajuste un modelo de regresión logística a los datos.
3. Utilizando la función `contourf`, visualice el mapa de probabilidades a posteriori estimadas para la categoría 1.
4. Determine las tasas de error de entrenamiento y validación.
5. Determine la verosimilitud de los datos de entrenamiento y test para el modelo obtenido.

#### 1.2. Regresión logística: clasificador no lineal

A continuación, analizaremos el comportamiento de un clasificador no lineal basado en un modelo de regresión logística con términos polinómicos.

1. Utilizando la función `glmfit`, ajuste un modelo de regresión logística con términos polinómicos de grado 1, 2 y 3 a los datos.

2. Utilizando la función `contourf`, visualice el mapa de probabilidades a posteriori estimadas para la categoría 1.
3. Calcule las tasas de error de entrenamiento y validación.
4. Calcule las verosimilitud de los datos de entrenamiento y test para el modelo obtenido.

### 1.3. Máquina de vectores soporte

Evaluaremos a continuación las prestaciones de un clasificador basado en máquinas de vectores soporte (SVM, *Support Vector Machine*).

Utilizaremos, para ello, una SVM con núcleos gaussianos de desviación típica 0.5.

1. Utilizando la función `svmtrain`, entrene una SVM con los datos de entrenamiento, visualizando la frontera de decisión, tomando  $C = 0.1$ .
2. Haciendo un barrido sobre el parámetro  $C$ , visualice las tasas de error de entrenamiento y validación, en función de  $C$ , y elija un valor de acuerdo con el error de validación.

### 1.4. Evaluación final

Seleccione, de acuerdo con las tasas de error de validación, el clasificador que mejores prestaciones ha obtenido, de entre todos los analizados en las secciones anteriores, y determine su tasa de error con los datos de test.

## 2. Clasificación con datos reales

Aplicaremos en esta sección los clasificadores anteriores a una base de datos reales multidimensionales. La visualización de los datos en el espacio de entrada ya no es posible, pero podemos evaluar las prestaciones utilizando las tasas de error.

Trabajaremos con el `cancer_dataset`.

### 2.1. Preparación de datos

Antes de entrenar el clasificador, conviene normalizarlos, para que la escala de todas las variables sea similar.

Las siguientes líneas de código realizan una transformación lineal que hace que los datos de entrenamiento tengan media nula y varianza unidad. Exactamente la misma transformación ha de aplicarse a los datos de validación y de test:

```
mx = mean(xTrain); stdx = std(xTrain);  
xTrain = (xTrain - ones(nTrain,1)*mx)./(ones(nTrain, 1)*stdx);  
xVal = (xVal - ones(nVal,1)*mx)./(ones(nVal,1)*stdx);  
xTest = (xTest - ones(nTest,1)*mx)./(ones(nTest,1)*stdx);
```

Realizaremos todo el resto de la práctica con los datos normalizados.

## 2.2. Clasificación con máquinas de vectores soporte

En este apartado entrenaremos una máquina de vectores soporte con núcleos gaussianos. Utilizaremos el conjunto de validación para determinar los valores de los hiperparámetros *sigma* y *C*.

1. Entrene la SVM para diferentes valores de  $\sigma$  entre -0.02 y 8 y valores de  $C$  entre 0.001 y 1000. Determine las tasas de error de entrenamiento y validación en cada caso, y represente gráficamente (por ejemplo, mediante la función `contourf`), dichas tasas de error.
2. Determine los valores de anchura del núcleo,  $\sigma$ , y del parámetro  $C$  que minimizan la tasa de error de validación.
3. Evalúe las prestaciones del clasificador utilizando los datos de test, para los valores de los hiperparámetros seleccionados en el apartado anterior.