

Práctica 2

Mikel de Velasco, Ieltzu Irazu, M Inés Fernandez

14 de octubre de 2015

1 Introducción

Se pide diseñar un clasificador que implemente el método k-NN básico con la distancia de Minkowski según la expresión (1) donde n representa el número de atributos empleados para caracterizar las muestras.

$$d(a, b) = \left[\sum_{i=1}^n |a_i - b_i|^m \right]^{\frac{1}{m}} \quad (1)$$

El clasificador permitirá seleccionar tanto el número de vecinos a explorar (k) como el parámetro m de la expresión (1). Elegir el lenguaje de programación que se considere más apropiado para el diseño. Para inferir el clasificador se dispone de un conjunto de datos (Diabetes.arff). El conjunto de datos dispone de 768 instancias para inferir el modelo. Para describir las instancias se utilizan 8 atributos más la clase. Los atributos son los siguientes:

- Number of times pregnant
- Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- Diastolic blood pressure
- Triceps skin fold thickness
- 2-Hour serum insulin
- Body mass index
- Diabetes pedigree function
- Age
- Class variable

La clase que hay que determinar es si el paciente tiene diabetes o no.

1.1 Parámetro k

El parámetro K es el parámetro que utilizaremos para indicar cuantos vecinos vamos a usar para evaluar el clasificador.

1.2 Parámetro m

El parámetro m será el parámetro que utilizaremos para saber que distancia tenemos que utilizar para evaluar nuestro clasificador.

2 Metodología

En este apartado desarrollaremos nuestro propio algoritmo K-Nearest Neighbors y describiremos el diseño y cómo hemos implementado este algoritmo. Además haremos una breve descripción de cómo hacer funcionar el programa.

El algoritmo que hemos desarrollado es básicamente una adaptación del clasificador IBk ya que hemos cogido su código y le hemos cambiado la forma de estimar que estaban utilizando.

Nuestro programa puede ser ejecutado desde eclipse de una manera muy fácil. Si ejecutas el Probador.java conseguirás una ejecución total del programa y obtendrás los resultados en la carpeta de ficheros. Por otro lado, también se pueden ejecutar de forma individual. Se diferenciarían 3 ejecuciones diferentes. Preprocesador.java nos sirve para quitar todo lo malo que venga en los datos (Extreme values, outliers,...).

Como parámetros de entrada tendremos los siguientes

Después ejecutaríamos Modelo.java y con ello conseguiríamos obtener los dos modelos de los que estamos tratando en la práctica.

Como parámetros de entrada tendremos los siguientes

Por ultimo, ejecutaríamos el Clasificador.java y con ello conseguiríamos los resultados que necesitamos para comparar los modelos anteriores.

Como parametros de entrada tendremos los siguientes

3 Resultados

En esta seccion mostraremos los resultados que hemos obtenido de las ejecuciones anteriores

4 Conclusiones

Realizar un clasificador desde cero ha sido una tarea complicada a que nos hemos basado en el modelo del IBk, y le hemos ajustado el pseudocódigo que teníamos. Ha sido una tarea entretenida y muy dinámica pero muy costosa. Como conclusión tenemos que decir que no merece la pena crear un modelo propio habiendo ya el IBk.

5 Valoración Subjetiva

Ieltzu: Personalmente ha sido una tarea que no me ha gustado mucho. He visto innecesaria la practica y para lo único que me ha servido es para darme cuenta de lo difícil que es implementar un clasificador propio.

Mikel:

Maria:

References