



UNIVERSIDAD
DE GRANADA

METAHEURÍSTICAS
GRADO EN INGENIERÍA INFORMÁTICA

PRÁCTICA 1

PROBLEMA DEL APRENDIZAJE DE PESOS EN
CARACTERÍSTICAS (APC)

Autor

Vladislav Nikolov Vasilev

NIE

X8743846M

E-Mail

vladis890@gmail.com

Grupo de prácticas

MH3 Jueves 17:30-19:30

Rama

Computación y Sistemas Inteligentes



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

CURSO 2018-2019

Índice

1. Descripción del problema	2
2. Descripción de los algoritmos	3
3. Descripción del método de búsqueda	3
Referencias	4

1. Descripción del problema

El problema que se aborda en esta práctica es el Aprendizaje de Pesos en Características (APC). Es un problema típico de *machine learning* en el cuál se pretende optimizar el rendimiento de un clasificador basado en vecinos más cercanos. Esto se consigue mediante la ponderación de las características de entrada con un vector de pesos W , el cuál utiliza codificación real (cada $w_i \in W$ es un número real), con el objetivo de modificar sus valores a la hora de calcular la distancia. Cada vector W se expresa como $W = \{w_1, w_2, \dots, w_n\}$, siendo n el número de dimensiones del vector de características, y cumpliéndose además que $\forall w_i \in W, w_i \in [0, 1]$.

El clasificador considerado para este problema es el 1-NN (genéricamente, un clasificador k -NN, con k vecinos, siendo en este caso $k = 1$), es decir, aquél que clasifica un elemento según su primer vecino más cercano utilizando alguna medida de distancia (en este caso, utilizando la distancia Euclídea). Cabe destacar que no en todos los casos se usará el clasificador 1-NN ya que se pueden dar casos en los que el vecino más cercano de un elemento sea él mismo. Por ese motivo, en algunas técnicas/algoritmos se usará un 1-NN con el criterio de *leave-one-out*, es decir, que se busca el vecino más cercano pero excluyéndose a él mismo.

El objetivo propuesto es aprender el vector de pesos W mediante una serie de algoritmos, de tal forma que al optimizar el clasificador se mejore tanto la precisión de éste como su complejidad, es decir, que se considere un menor número de características. Estos dos parámetros, a los que llamaremos *tasa_clas* y *tasa_red*, respectivamente, se pueden expresar de la siguiente forma:

$$tasa_clas = 100 \cdot \frac{n^\circ \text{ instancias bien clasificadas en } T}{n^\circ \text{ instancias en } T}$$
$$tasa_red = 100 \cdot \frac{n^\circ \text{ valores } w_i < 0.2}{n^\circ \text{ características}}$$

siendo T el tamaño del conjunto de datos sobre el que se evalúa el clasificador.

Por tanto, al combinarlos en una única función a la que llamaremos $F(W)$, la cuál será nuestra función objetivo a optimizar, tenemos que:

$$F(W) = \alpha \cdot tasa_clas(W) + (1 - \alpha) \cdot tasa_red(W)$$

siendo α la importancia que se le asigna a la tasa de clasificación y a la de reducción, cumpliendo que $\alpha \in [0, 1]$. En este caso, se utiliza un $\alpha = 0.5$ para dar la misma importancia a ambos, con lo cuál se pretende que se reduzcan al máximo el número de características conservando una *tasa_clas* alta.

2. Descripción de los algoritmos

3. Descripción del método de búsqueda

Referencias

- [1] Texto referencia
<https://url.referencia.com>