E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Aprendizaje de distancias basadas en disimilitudes para el algoritmo de clasificación kNN

Grado en Ingeniería Informática

Trabajo Fin de Grado

Mikel Xabier Uriz Martin

Mikel Galar Idoate

Pamplona, 26 de enero de 2015

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

RESUMEN

El objetivo de este proyecto es el de tratar de mejorar el algoritmo KNN (k vecinos más cercanos) sustituyendo la distancia Euclidea clásica por disimilitudes parametrizadas que serán ajustadas utilizando un algoritmo genético. La idea es que el algoritmo genético aprenda diferentes parámetros para luego calcular las distancias entre instancias utilizando esos parámetros, en vez de utilizar otras distancias clásicas como la Euclidea.

También consideramos la opción de poder realizar la selección de instancias y de atributos, de esta manera, el algoritmo genético podrá excluir las instancias que sean ruido. Al utilizar esta técnica se acelerara el cálculo de las distancias, ya que al disminuir el número de instancias y de atributos, se requieren menos cálculos a la hora de calcular las distancias.

Al final, realizaremos una comparativa con las diversas variantes que se puedan dar y el algoritmo KNN original, para ver si existe mejora a la hora de clasificar.

Palabras clave: k-vecinos más cercanos, disimilitudes, algoritmos genéticos, clasificación.

Índice

1. Introducción

Este proyecto consiste en desarrollar un sistema de clasificación basado en el algoritmo de los k vecinos más cercanos. Más concretamente, tratamos de aprender diferentes parámetros para la fase de cálculo de distancias del algoritmo de los k vecinos más cercanos donde introducimos el uso de disimilitudes.

1.1. Sistemas de clasificación

Un sistema de clasificación trata de clasificar en diferentes clases o categorías una serie de ejemplos o instancias que representan cierta información de un problema. En el ámbito del aprendizaje automático, el objetivo de estos sistemas es aprender a decidir cuál es la clase a la que pertenecen los ejemplos nuevos sin etiquetar. Existen dos tipos clasificación:

- **Clasificación supervisada:** En este tipo de clasificación, se tiene un conjunto de datos de los cuales ya sabemos su clasificación, llamados instancias de entrenamiento o conjunto de entrenamiento. Utilizando dichos ejemplos hacemos que el sistema "aprenda" ciertos parámetros los cuales serán utilizados para clasificar nuevas instancias.
- **Clasificación no supervisada:** En este tipo de clasificación no se tiene un conjunto de datos de los cuales se sabe la clase. En este caso el sistema aparte de clasificar tiene que establecer las clases.

1.2. Algoritmo de los K vecinos más cercanos

El algoritmo los K vecinos más cercanos (KNN) es un algoritmo de clasificación supervisada. Este algoritmo trata de clasificar el ejemplo utilizando un conjunto de datos de los cuales ya se conoce su clase. Primero calcula la distancia entre el ejemplo que se quiere clasificar y todos los elementos del conjunto y selecciona los K ejemplos con menor distancia. La clase para el ejemplo que se quiere clasificar será la clase que más se repita en esos K ejemplos.

1.3. Algoritmos genéticos

Los algoritmos genéticos son algoritmos que realizan búsquedas basadas en heurísticas que imitan el proceso de selección natural. El objetivo de estos algoritmos es buscar una solución para problemas de optimización y búsqueda.

Los algoritmos genéticos utilizan una población que son los candidatos a la solución. Cada candidato es un cromosoma que está compuesto por genes, que codifican los parámetros que se quieren "aprender". Estos candidatos van evolucionando y en cada generación van quedando los que mejor se adaptan, es decir, los que más se aproximan a la mejor solución en base a la función objetivo establecida.

Primero se evalúa cada candidato y se obtiene una puntuación relacionada con la bondad de esa solución. Cuanto mayor sea esta puntuación mayor será la probabilidad de reproducción de este individuo. Cuando dos individuos se cruzan se obtienen descendientes que comparten características (genes) de los padres. Si algún individuo no consigue adaptarse bien, su probabilidad de reproducción será menor. De esta manera, generación a generación se van consiguiendo individuos que se adaptan mejor, es decir, se van encontrando mejores soluciones al problema dado.

1.4. Objetivos

El objetivo de este proyecto es el de utilizar un algoritmo genético para aprender diferentes parámetros que se utilizaran para calcular las distancias entre los ejemplos en el algoritmo de los K vecinos más cercanos. La distancia es calculada utilizando medidas de disimilitud restringidas, que miden lo distintos que son dos ejemplos. De esta manera tratamos de mejorar al algoritmo original de los K vecinos más cercanos, y a algunas de sus variantes.

2. Técnicas utilizadas

2.1. Medidas de disimilitud

Estas medidas son las que utilizaremos para calcular la distancia entre ejemplos. Las medidas de disimilitudes pueden construirse por medio de unas funciones llamadas automorfismos. Estas funciones las representamos con el símbolo φ y son funciones continuas y estrictamente crecientes. La definición de la función es la siguiente: φ: $[a,b] \rightarrow [a,b]$ donde $\varphi(a) = a$ y $\varphi(b) = b$ siendo $[a,b] \subset R$.

En este proyecto trabajamos con números reales entre 0 y 1, por lo tanto nuestro valor de a sera 0 y el de b 1.

Por otra parte, la función de automorfismo que hemos elegido es $\varphi(x) = x^p$, siendo p un número real en el rango $(0, \infty)$. Como se puede observar, esta función cumple las condiciones: $\varphi(0) = 0^p = 0$ y $\varphi(1) = 1^p = 1$.

La distancia que utilizamos se basa en medidas de disimilitud restringidas. Estas funciones tienen que cumplir las siguientes propiedades:

1) $d(x,y) = d(y,x)\ para\ todo\ x,y\ \in [0,1]$
2) $d(x,y) = 1\ si\ y\ solo\ si\ x = 0\ e\ y = 1\ o\ x = 1\ e\ y = 0$
3) $d(x,y) = 0\ si\ y\ solo\ si\ x = y$
4) $Para\ todo\ x,y,z \in [0,1], si\ x \leq y \leq z\ entonces\ d(x,y) \leq d(x,z)\ y\ d(y,z) \leq d(x,z)$

Las medidas de disimilitud restringidas se construyen a partir de dos automorfismos φ_1 y φ_2 en el intervalo $[0,1]$. La distancia que tenemos es $d(x,y) = \varphi_1^{-1}(|\varphi_2(x) - \varphi_2(y)|)$ que devuelve un número real entre 0 y 1 que indica lo diferente que son los elementos, siendo 1 que son totalmente diferentes y 0 que son el mismo.

Utilizando los automorfismos citados, la función de distancia que obtenemos es $d(x,y) = (|x^{p2} - y^{p2}|)^{1/p1}$. Ya que el primero automorfismo es la inversa de la función $\varphi(x) = x^p$ se obtiene que $\varphi^{-1}(x) = x^{1/p} = \sqrt[p]{x}$. Con esto se puede definir la función de la distancia como $d(x,y) = \sqrt[p1]{|x^{p2} - y^{p2}|}$.

Esta función de distancia se aplica entre dos números, y normalmente los ejemplos vienen definidos por varios valores en forma de vector, por lo tanto podemos calcular la distancia entre los elementos del vector, pero no la distancia entre dos

ejemplos. Para calcular la distancia general, tenemos que agregar los resultados. Para ello definimos una función $D: F(U)xF(U) \rightarrow [0,1]$ donde se cumple:

1) $D(A,B) = D(B,A)\ para\ todo\ A,B \in F(U)$
2) $D(A,B) = 0\ si\ y\ solo\ si\ A = B$
3) $D(A,B) =$
   $1\ si\ y\ solo\ si\ A\ y\ B\ son\ conjuntos\ no\ fuzzy\ complementarios$
4) $Si\ A \leq A' \leq B' \leq B\ entonces\ D(A,B) \geq D(A',B')$

Definimos la función de agregación $M: [0,1]^n \rightarrow [0,1]$ y que cumple:

1) $M(x_1, \dots, x_n) = 0\ si\ y\ solo\ si\ x_1 = \cdots = x_n = 0$
2) $M(x_1, \dots, x_n) = 1\ si\ y\ solo\ si\ x_1 = \cdots = x_n = 1$
3) $M\ es\ creciente$

Finalmente obtenemos la siguiente función de disimilitud:

$$D(A,B) = M_{i=1}^{n} d(x_i, y_i)$$

En nuestro caso hemos elegido la media aritmética como función de agregación. Uniendo todas las funciones, la función de disimilitud resultante es:

$$D(A,B) = \frac{1}{n}\sum_{i=1}^{n} d(x_i, y_i) = \frac{1}{n}\sum_{i=1}^{n}\left[\sqrt[p1_i]{|x_i^{p2_i} - y_i^{p2_i}|}\right]$$

2.2. Algoritmo KNN

Como ya se ha dicho, el algoritmo de los K vecinos más cercanos es un algoritmo de clasificación supervisada. Es un algoritmo muy fácil de entender e implementar y que funciona bien en varios casos. Dado un conjunto de entrenamiento y un ejemplo para clasificar, el algoritmo devuelve la clase mayoritaria de los K ejemplos más cercanos.

A diferencia de otros algoritmos que utilizando el conjunto de entrenamiento aprenden algunos parámetros y luego utilizando estos últimos clasifican la nueva instancia, el algoritmo KNN clasifica la nueva instancia utilizando el conjunto de entrenamiento sin necesidad de aprender ningún parámetro. El algoritmo de KNN es el siguiente:

ALGORITMO 1

ENTRADA: D, el conjunto de entrenamiento, el elemento para test, **z**, que es el vector a clasificar, y **L**, el conjunto de clases.

SALIDA: $c_z \in L$, la clase de z

para cada objeto y ∈ D **hacer**

Calcular d(**z**, **y**), la distancia entre z e y;

fin

Seleccionar N ⊆ D, los k elementos más cercanos a z dentro del conjunto de entrenamiento;

$$c_z = argmax_{v \in L} \sum_{y \in N} I\left(v = class(c_y)\right);$$

donde I (·) es una función que devuelve 1 si el argumento es cierto, 0 en otro caso.

El algoritmo recibe un conjunto de entrenamiento, una instancia a clasificar y un vector que contiene las diferentes clases del conjunto. Por cada instancia del conjunto de entrenamiento, se calcula la distancia entre esa instancia y la instancia que se quiere clasificar. Una vez que tenemos todas las distancias calculadas, seleccionamos las k con menor distancia. Para terminar agregamos los resultados obtenidos en las distancias de las k instancias seleccionadas para obtener una predicción de la clase final.

Una de las cosas a tener en cuenta es el valor de k, es decir, el número de vecinos a tener en cuenta para clasificar la nueva instancia. Si el valor es muy pequeño, entonces el resultado puede ser sensible a instancias que son ruido. Si el valor es muy alto, dentro de los k más cercanos podemos incluir instancias que son de otra clase. Esto se puede ver en la siguiente figura:

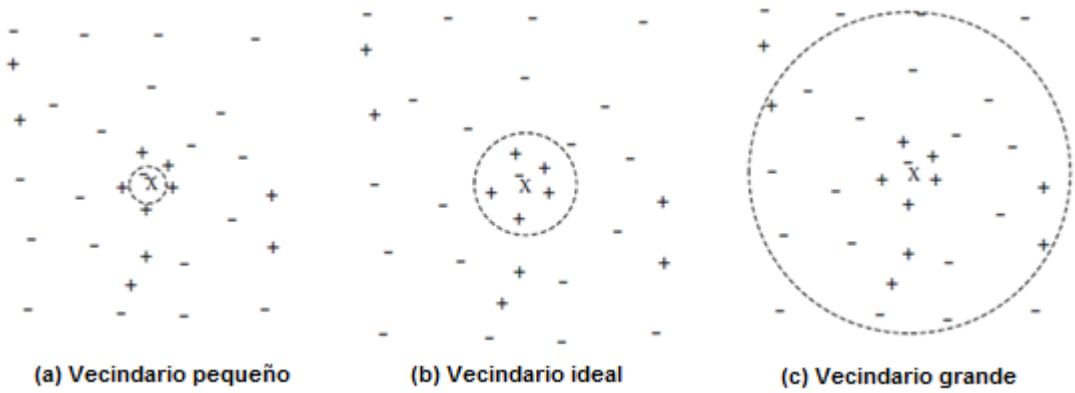

Figura 1. Diferentes tipos de vecindarios según el valor de k.

Otra cosa que hay que tener en cuenta es como realizar la agregación final. Una de la forma de hacerlo es por el método del voto. Cada una de las k instancias seleccionadas vota por la clase a la que pertenece y la clase mayoritaria es la que se le asigna a la instancia a clasificar. Otra forma de hacer la agregación es a través del voto ponderado. Cada una de las k instancias vota por la clase a la que pertenece pero con

un peso. Este peso podría depender de la distancia que hay entre él y la instancia a clasificar, cuanto menos distancia mayor será el peso del voto. En el caso de este proyecto aplicaremos el voto sin ponderar.

Por último, otra de las cosas que hay que seleccionar en este algoritmo es la distancia. Normalmente se aplica la distancia Euclidea o la de Manhattan, aunque hay otro tipo de distancias que se pueden aplicar. En este caso emplearemos las distancias creadas mediante disimilitudes explicadas en el apartado anterior.

2.3. Algoritmos genéticos

Como ya se ha explicado en la sección 1.3., los algoritmos genéticos imitan la evolución natural. Una población inicial va evolucionando y obteniendo mejores resultados cruzando individuos de la población y/o mutándolos. Cada individuo, también llamado cromosoma, se representa como una cadena de valores, donde cada elemento se llama gen, y cada uno de estos genes hace referencia a un parámetro que se quiere ajustar. El esquema de un algoritmo genético es el siguiente:

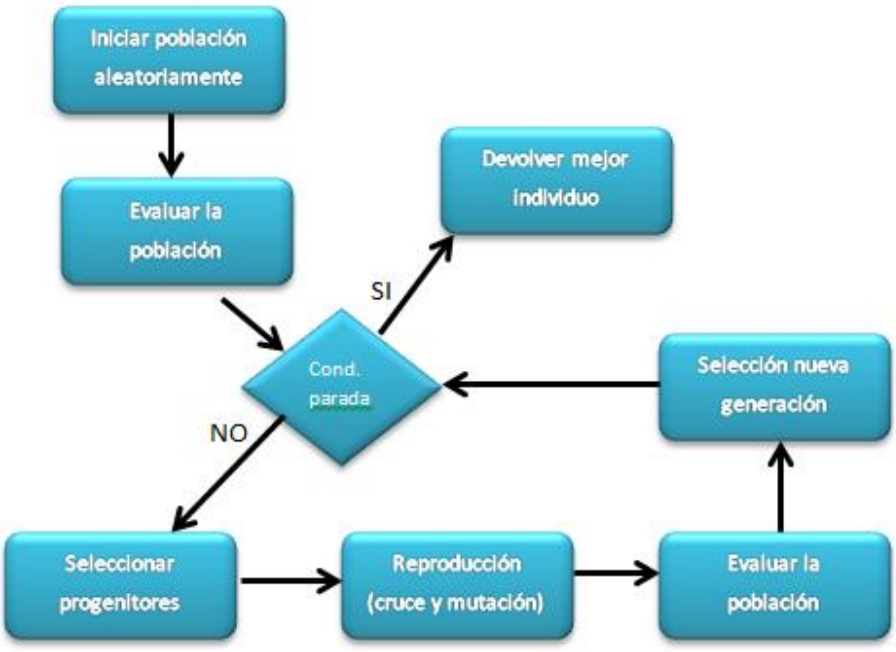

Figura 2. Esquema general de un algoritmo genético.

Primero iniciamos una población aleatoriamente, dando un valor aleatorio a cada gen de cada cromosoma. Una vez que tenemos la población, evaluamos cada cromosoma. En este punto ya tenemos como de bueno es cada individuo. Si esta

población satisface la condición de parada (que se supere el número de iteraciones, que se encuentre la solución óptima,…) devolvemos el mejor cromosoma. Si no satisface esa condición, seleccionamos los progenitores y se realiza la reproducción, cruzando los progenitores y si se da el caso, se mutan alguno de los genes de los descendientes que se crean. Una vez que tenemos los descendientes, los evaluamos y seleccionamos los mejores cromosomas (de la población anterior y los descendientes creados) sin superar el tamaño de la población anterior. Se vuelve a comprobar la condición de parada y si se satisface se devuelve el mejor cromosoma. Si no, se repiten de nuevo estos pasos. La población que se genera generación tras generación, siempre será mejor o igual a la anterior, ya cada generación nos quedamos con los mejores.

En este algoritmo, hay que tener en cuenta los siguientes aspectos:

- **Representación:** Lo primero que tenemos que decidir es como vamos a representar el cromosoma. Tenemos que decidir el número de genes y tipo de cada gen. Por ejemplo, el problema del viajante tendremos tantos genes como destinos tengamos y cada uno de estos genes será un número entero que represente cada destino. Otro tipo de representación podría ser un numero binario (0 o 1) por gen, donde cada gen representa cada uno de los ejemplos del conjunto de entrenamiento y 1 significa que si utilizamos ese ejemplo para clasificar y un 0 que no (selección de instancias).
- **Función de evaluación:** Es la función que se utiliza para calcular lo bueno que es cada uno de los cromosomas de la población. El valor que devuelve esta función se llama fitness. Si el problema es de maximización, cuanto mayor sea el fitness mejor es el cromosoma, al contrario que en un problema de minimización, cuanto menos fitness mejor cromosoma. En el caso del problema del viajante tendríamos la función de evaluación como una función que calcula el total de la distancia que se recorre. En el caso de la selección de instancias, podría ser la precisión que se obtiene después de seleccionar alguna de los ejemplos.
- **Condición de parada:** pueden ser varias, que se llegue a la mejor solución, que se supere el número de iteraciones previamente escogido,…
- **Selección de progenitores:** Es el criterio que se sigue a la hora de escoger los padres para la reproducción. Hay varias formas. Las más usadas son:
    - **Selección elitista:** Se garantiza la selección de los miembros más aptos de cada generación.
    - **Selección proporcional a la aptitud:** los individuos más aptos tienen mayor probabilidad de ser seleccionados, pero no la certeza.
    - **Selección del método de la ruleta:** A cada uno de los individuos de la población se le asigna una parte proporcional a su valor fitness, de tal manera que la suma de todos los porcentajes sea la unidad. Los mejores individuos tendrán mayor porcentaje que los peores. Se genera un

número aleatorio, y el individuo que este en esa posición es el seleccionado.

- **Selección por método del torneo:** En este caso se eligen aleatoriamente n individuos (normalmente n es 2) y se selecciona el mejor de los n. También se puede hacer con un método probabilístico. En vez de pasar el mejor, se genera un número aleatorio en el intervalo $[0,1]$, y si ese número es mayor que una cota fijada previamente se escoge el mejor individuo, en caso contrario el menos apto.

- **Cruce:** El cruce consiste en combinar dos cromosomas, llamados progenitores, y generar nuevos cromosomas, llamados descendientes. Estos descendientes comparten alguna de las características (genes) de los progenitores. Hay varios métodos de cruce, y difieren dependiendo de la representación. Por ejemplo, alguno de los cruces para cromosomas binarios:
  - **Cruce sobre un punto:** Se elige un punto aleatorio del cromosoma. La primera parte del primer progenitor y la segunda parte del segundo, será el primer descendiente, y la primera parte del segundo progenitor y la segunda parte del primero, será el segundo descendiente. Ejemplo:

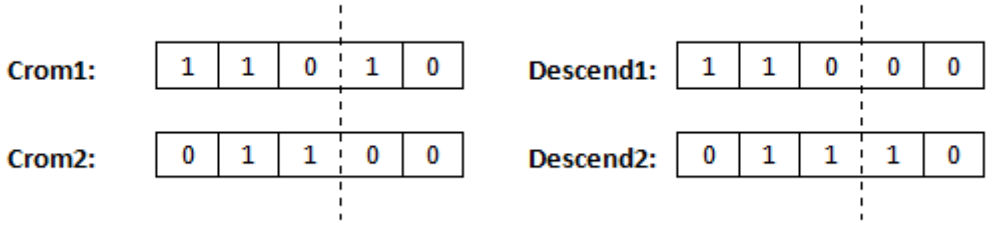

Figura 3. Ejemplo de cruce sobre un punto

  - **Cruce multipunto:** Es la generalización del anterior tomando varios puntos.

Por ejemplo para los reales tenemos otros métodos diferentes. El siguiente es uno:

  - **Recombinación simple:** Se elige un punto al azar. Los genes desde el inicio hasta el punto formaran parte de los genes de los descendientes, y el resto son calculados utilizando la media aritmética. Ejemplo:

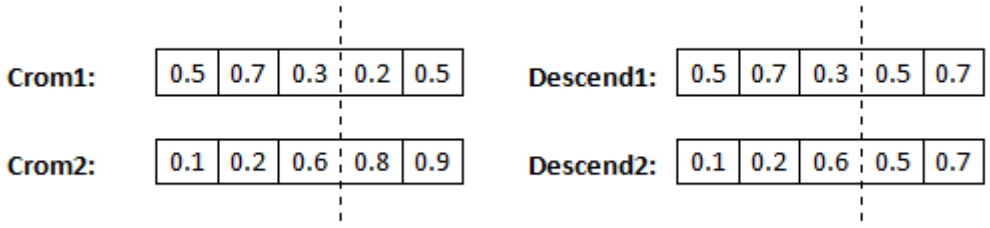

Figura 4. Ejemplo de recombinación simple

- **Mutación:** La mutación es la alteración de alguno de los genes de un cromosoma. Por ejemplo, en un cromosoma donde cada gen es binario, su valor puede cambiar de 0 a 1 o de 1 a 0 con la probabilidad que establezcamos. En general esta probabilidad es muy baja, se intenta que muten pocos genes por generación.
- **Selección de la siguiente generación:** Una vez creados los descendientes y unir los descendientes con los progenitores, tenemos que hacer la selección de la siguiente generación. Existen varias técnicas:
    - **Selección por edad:** Cada individuo solo está presente en el algoritmo durante un número fijado de generaciones. Esto puede llevar a la eliminación de los mejores adaptados.
    - **Selección basada en la adaptación:** En este grupo se encuentran los métodos de la ruleta y del torneo, anteriormente mencionados. A parte de estos existen otros:
        - **Reemplazar el peor:** Se eliminan los peores individuos. Puede producir convergencia prematura.
        - **Elitismo:** La idea de este método es la de mantener siempre en la población al mejor adaptados entre los progenitores y los descendientes.

El pseudo-código del algoritmo genético es:

ALGORITMO 2

```
poblacion = generarPoblacionAleatoriamente()
generacion = 0
evaluar (poblacion)
mientras (no condicionDeParada) hacer
    padres = seleccionProgenitores (poblacion)
    hijos = cruzar (padres)
    hijos = mutar (hijos)
    evaluar (hijos)
    poblacion = seleccionSiguientePoblacion (hijos, poblacion)
    generacion = generacion + 1
fin mientras
devolver mejor individuo
```

2.4. Algoritmo CHC

El algoritmo CHC es un algoritmo genético sencillo y robusto que normalmente ofrece buenos resultados. Originalmente este algoritmo estaba pensado para utilizar cromosomas con representación binaria, pero actualmente también se puede utilizar con codificación real. En este proyecto se utilizan ambos tipos de cromosomas.

Una de las características de este algoritmo es la selección elitista, es decir, los mejores individuos de la población son los que se seleccionan para la nueva población. Otra característica importante es que este algoritmo tiene un mecanismo de prevención de incesto que evita que, los cromosomas seleccionados para procrear, lo hagan si son muy parecidos. Para esto calculamos la distancia de Hamming entre los cromosomas, y si la mitad de esa distancia es mayor que un umbral d, estos cromosomas no son cruzados. Normalmente ese umbral d es un cuarto de la longitud del cromosoma, que se va reduciendo 1 cada vez que no obtenemos nuevos descendientes. Tambien hay que decir que este algoritmo no tiene mutacion. Cuando la población generación a generación no varía, la población es restablecida. Normalmente se utiliza el umbral d, cuando este es 0 se incia el proceso de reincio. En este proceso se mantiene el mejor individuo y el resto es generado aleatoriamente puediendo incluir parte de los genes del mejor.

El pseudo-codigo del algoritmo CHC es el siguiente:

ALGORITMO 3

```
poblacion = generarPoblacionAleatoriamente()
d = LongitudCromosoma / 4
evaluar (poblacion)
mientras (no condicionDeParada) hacer
    padres = seleccionProgenitores (poblacion)
    hijos = cruzar (padres)
    evaluar (hijos)
    seleccionSiguientePoblacion (hijos, poblacion)
    si poblacion no ha cambiado entonces
        d = d – 1
    fin si
    si d < 0 entonces
        poblacion = reiniciar (poblacion)
        d = LongitudCromosoma / 4
    fin si
fin mientras
devolver mejor individuo
```

3. Propuestas

3.1. Algoritmo KNN con medidas de disimilitud

Nuestro objetivo consiste en utilizar las medidas de disimilitud previamente comentadas para calcular la distancia entre instancias en el algoritmo KNN. Como ya hemos dicho, estas medidas emplean funciones llamadas automorfismos. En este caso utilizaremos los siguientes automorfismos:

$$\varphi_1 = x^a \quad \varphi_2 = x^b$$

Del primer automorfismo tenemos que realizar la inversa, por la definición de la disimilitud. Por lo tanto nos queda:

$$\varphi_1 = x^{1/a}$$

La función de disimilitud que nos queda es la siguiente:

$$d(x,y) = (|x^b - y^b|)^{1/a}$$

Por lo tanto esa es la función que utilizaremos para calcular la distancia (en un atributo) entre la instancia x y la instancia y. Utilizando la agregación antes comentada:

$$D(X,Y) = \frac{1}{n}\sum_{i=1}^{n}\left[(|X(i)^{b_i} - Y(i)^{b_i}|)^{1/a_i}\right]$$

Por cada uno de los atributos tenemos un valor a y un valor b. Se calcula la disimilitud entre el atributo i de la instancia X y el atributo i de la instancia Y con el valor a_i y b_i, y finalmente se calcula la media de todas las disimilitudes.

Pero antes de calcular la distancia, tenemos que convertir el gen, que es un número real en el rango $(0,1)$, a un real en el rango $(0,\infty)$. Para esto utilizamos la siguiente función:

$$f(x) = \begin{cases} (2 * x)^2, & x \leq 0.5 \\ \dfrac{1}{(2 * (1-x))^2}, & x > 0.5 \end{cases}$$

Con esta función, convertimos los números del $(0,0.5]$ a $(0,1]$ y los$(0.5,1)$ a $(1,\infty)$.

Como se puede observar dándole ciertos valores a a y a b podemos obtener la distancia Euclidea y la de Manhattan. La distancia de Manhattan es la suma de las

diferencias en valor absoluto, si los valores de a y de b obtienen el valor 1, recuperamos esa función. Por tanto, tenemos que buscar los valores tales que el resultado de aplicarlos a la función sea 1.

$$(2 * x)^2 = 1 \rightarrow 2 * x = \sqrt{1} \rightarrow x = \frac{1}{2}$$

Dando el valor 0.5 a cada gen obtenemos la distancia Manhattan:

$$D(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(|X(i)^1 - Y(i)^1|)^{1/1} = \frac{1}{n}\sum_{i=1}^{n}|X(i) - Y(i)|$$

Para la distancia Euclidea, tenemos que elevar cada valor a 1, por lo tanto tenemos que b es 1. Luego tenemos que elevar al cuadrado la diferencia, por lo tanto a tiene que ser 0.5.

Ya hemos calculado antes que para conseguir el valor 1 tenemos que utilizar 0.5. Para conseguir el valor 0.5:

$$(2 * x)^2 = 0.5 \rightarrow 2 * x = \sqrt{0.5} \rightarrow x = \frac{\sqrt{0.5}}{2}$$

Ese es el valor que tenemos que utilizar para obtener el valor 0.5. Finalmente tenemos:

$$D(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(|X(i)^1 - Y(i)^1|)^{1/0.5} = \frac{1}{n}\sum_{i=1}^{n}(|X(i) - Y(i)|)^2$$

En la formula anterior faltaría hacer la raíz cuadrada del resultado final, pero como la raíz es una función continua, no se altera el orden, es decir, si $x \leq y$ entonces $\sqrt{x} \leq \sqrt{y}$. Teniendo la distancia $d_1 = D(x,y)$ y $d_2 = D(x,z)$ si $d_1 \leq d_2$, se cumple que $\sqrt{d_1} \leq \sqrt{d_2}$.

Por lo tanto nos queda el algoritmo KNN normal, pero utilizando las medidas de disimilitud como distancia.

3.2. Algoritmo genético

Como ya se ha dicho el algoritmo genético que vamos a utilizar es el CHC. Los parámetros que va a tratar de ajustar este algoritmo son los valores de a y b que se utilizan en los automorfismos. Los aspectos que tenemos que tener en cuenta son:

- **Representación:** Cada atributo necesita de dos parámetros (un a y un b), por consiguiente, cada cromosoma tendrá tantos genes como atributos multiplicado por 2. Cada uno de estos valores, será un número real, en el intervalo $(0,1)$.
- **Función de evaluación:** Como lo que queremos es obtener una mejor precisión, el valor por el cual ordenaremos la población será este. La función que utilizaremos, será el algoritmo KNN previamente explicado, donde le pasamos el cromosoma con los valores de los a's y de las b's (para poder calcular la distancia basándose en la disimilitud) y nos devuelve la precisión.
- **Condición de parada:** Hay tres casos en los que el algoritmo dejaría de iterar y devolvería el mejor individuo:
    - Se encuentra la solución óptima. En este caso, el mejor individuo ha obtenido 100% de aciertos de clasificación.
    - Se ha superado el número de iteraciones previamente fijado.
    - Como es propio de CHC, se superan el número máximo de reinicios sin mejorarse la población.
- **Selección de progenitores:** En este caso, seleccionamos todos los individuos de la población y los cruzamos aleatoriamente entre ellos.
- **Cruce:** Para el cruce de reales, se va a utilizar el cruce PCBLX. Este método, dados dos cromosomas genera otros dos nuevos. El método es el siguiente: Tenemos el cromosoma $a = [a_1, a_2, a_3 \dots, a_n]$ y $b = [b_1, b_2, b_3 \dots, b_n]$. Por cada gen se calcula un valor $I_i = |a_i - b_i|$. Se calculan los valores $l1_i = \max(0,\ a_i - I_i)$, $u1_i = \min(1,\ a_i + I_i)$, $l2_i = \max(0,\ b_i - I_i)$ y $u2_i = \min(1,\ b_i + I_i)$. Para el primer descendiente, por cada gen se genera un número aleatorio en el rango $[l1_i, u1_i]$ y para el segundo en el rango $[l2_i, u2_i]$.
- **Mutación:** Como ya se ha dicho, el CHC no realiza el proceso de mutación. A cambio de este, se utiliza un método de reinicio, donde si la población no cambia durante una serie de generaciones, se elimina toda la población excepto el mejor y se generan el resto aleatoriamente.
- **Selección de la siguiente iteración:** Como ya se ha dicho, el CHC es un algoritmo que hace selección elitista, es decir, se seleccionan los mejores individuos para la siguiente población.

3.3. Selección de instancias.

Otra de las cosas que se añade al algoritmo genético es la posibilidad de hacer selección de instancias. En este caso para la representación utilizaremos números binarios, donde la longitud del cromosoma será el número de instancias de

entrenamiento que tengamos. Un 1 en la posición i significa que la instancia numero i es utilizada para clasificar, y un 0 que no.

A la función de evaluación tenemos que añadir un término que penalice la reducción de instancias masiva. Si no añadiéramos este término, probablemente el número de instancias sería muy bajo, ya que con pocas instancias es más probable tener un porcentaje de clasificación mayor.

Definimos el ratio de reducción como:

$$ReductionIS = 1.0 - \frac{\#InstancesSelected}{N}$$

Donde $\#InstancesSelected$ es el número de genes a 1 y N es el número de instancias.

Con todo esto definimos el nuevo fitness como:

$$Fitness = \alpha * classRate + (1 - \alpha) * ReductionIS$$

Donde $classRate$ es el porcentaje de clasificación y α un valor real para dar peso a cada término. En este caso, si α vale 1, el ratio de clasificación sería el fitness total, y no se tiene en cuenta el término de la reducción.

Como en este caso se tratan de genes binarios el método de cruce tiene que ser otro diferente al de la parte de los reales. En este caso empleamos HUX que es un método de cruce que intercambia exactamente la mitad de los bits que son distintos entre los padres. Esta función recibe adicionalmente un parámetro para indicar la probabilidad de que un bit durante el cruce pase de ser 0 a 1.

3.4. Selección de características

Como en el caso anterior, para la representación se utilizan números binarios. En este caso, tendremos tantos genes como atributos tenga el conjunto de instancias de entrenamiento. Del mismo modo, si tenemos un 1 en la posición i se utiliza el atributo i para clasificar. Si hay un 0, no se utiliza.

De la misma manera que en el caso anterior, tenemos que modificar la función de evaluación para evitar una alta reducción de atributos. Primero definimos el ratio de reducción:

$$ReductionFS = 1.0 - \frac{\#FeaturesSelected}{M}$$

Donde $\#FeatureSelected$ es el número de atributos seleccionados y M es el número de atributos. Con esto, la función de evaluación:

$$Fitness = \beta * classRate + (1 - \beta) * ReductionFS$$

Donde $classRate$ es el porcentaje de aciertos y β un valor real para dar peso a cada término. Como antes, si β vale 1, se carga todo el peso sobre el ratio de clasificación.

En este caso como también tenemos genes binarios, utilizamos la función HUX anteriormente descrita.

3.5. Modelo completo

El modelo completo es la combinación de los últimos apartados. Por una parte tenemos la parte real y por otro la parte en binario (una parte para la selección de instancias y otra para la selección de características). Por lo tanto nuestro cromosoma tendrá $NumeroAtributos * 3 + NumeroEjemplos$, tantos genes como atributos multiplicado por 2 por la parte real, también número de atributos por la parte binaria de selección de atributos y tantos genes como instancias de entrenamiento por la parte de selección de instancias.

No siempre se va a utilizar la selección de instancias o la selección de características. Por lo tanto un parámetro de la función será si utilizamos selección de instancias, selección de características o ambas. En este caso el número de genes será el mismo, con la diferencia que los genes que correspondan a la selección que no se quiere utilizar, tendrán todos el valor 1 (se seleccionan siempre).

Para la función de evaluación, combinamos las anteriores y obtenemos la siguiente función:

$$fitness = \alpha * \beta * classRate + (1 - \alpha) * ReductionIS + (1 - \beta) * ReductionFS$$

Donde los ratios de reducción de instancias y de características son las mencionadas anteriormente. Para los valores de α y β tenemos que tener en cuenta que tipo de reducción se va a realizar. Si no se utiliza reducción de instancias, el parámetro α será 1, de esta manera eliminamos la parte de la reducción de instancias y damos todo el peso a la selección de características y el porcentaje de instancias bien clasificadas. Los mismo sucede con la reducción de características, si no se utiliza β valdrá 1 eliminado la parte de la reducción de características. Si no se utiliza ninguno de los dos métodos, todo el peso nos quedaría en el ratio de clasificación. En cualquier otro caso, se asigna a α y β los valores 0.6 y 0.99 respectivamente.

Para realizar el cruce, por definición del algoritmo CHC, tenemos que calcular la distancia de Hamming entre los progenitores, y solo cruzar aquellos cuya diferencia sea mayor que un umbral establecido. Pero la distancia de Hamming se utiliza para números binarios. Por lo tanto hay que convertir los números reales a binario.

DE NÚMEROS REALES A BINARIO: CÓDIGO GRAY

Lo que se quiere hacer es obtener números binarios para calcular la distancia entre dos cromosomas. Pero por ejemplo la diferencia entre 3 y 4 es 1, pero si pasamos los números a binario y calculamos la distancia de Hamming: $3 \rightarrow 011$ y $4 \rightarrow 100$, la distancia es el número de bits diferentes por posición, como tenemos 3 bits, y todos son diferentes, la distancia total es 3. Por esta razón vamos a utilizar la codificación de gray para los números binarios, que se cumple que entre dos números consecutivos su distancia de Hamming es 1. Por ejemplo la codificación de gray con 3 bits es la siguiente:

$$0 \rightarrow 000$$
$$1 \rightarrow 001$$
$$2 \rightarrow 011$$
$$3 \rightarrow 010$$
$$4 \rightarrow 110$$
$$5 \rightarrow 111$$
$$6 \rightarrow 101$$
$$7 \rightarrow 100$$

En el ejemplo se ve como la distancia de Hamming entre un elemento y el siguiente (o el anterior) es 1.

El proceso para convertir un número real, teniendo en cuenta que estos números reales están dentro del rango $(0,1)$, es el siguiente:

Lo primero que hacemos es convertir el real a entero. Para ello cada valor real lo convertimos a entero con la siguiente función:

$$f(x) = round(x * 2^{BITGEN} + 0.5)$$

Donde BITGEN es el número de bits que vamos a usar para representar cada valor real. En este trabajo este valor es 30.

Una vez que tenemos el número entero lo convertimos a binario de forma normal. Y una vez que tenemos el número en binario, convertimos este a binario pero en codificación gray. Para realizar este proceso hay que desplazar el número original un bit hacia la derecha, y calcular la operación OR exclusivo entre el número original y el número desplazado sin acarreo.

Por ejemplo para el número 6 (110), y su valor desplazado (11) calculamos la operación OR exclusivo (bit a bit): $110\ xor\ 011 \rightarrow 1\ xor\ 0 = 1, 1\ xor\ 1 = 0, 0\ xor\ 1 = 1$. Resultado 101.

Una vez que ya podemos calcular las distancias, calculamos la distancia de cada una de las partes de los cromosomas y realizamos el cruce.

Por una parte cruzamos la parte real del cromosoma utilizando el cruce PCBLX anteriormente mencionado y para la parte binario utilizamos el método HUX, también explicado anteriormente.

Como se mencionó en la sección 3.1., dándole ciertos valores a a y a b, podemos recuperar las distancias euclidea y manhattan. En un principio, la población inicial se genera aleatoriamente, pero en este caso introducimos dos cromosomas que recuperen estas dos distancias dándole los valores que hemos calculado anteriormente y el resto lo generamos aleatoriamente.

Introduciendo estos cromosomas, partimos de que entrenamos el algoritmo con estas distancias. En consecuencia, el algoritmo como mínimo obtendrá el mismo porcentaje de acierto, en el conjunto de entrenamiento, que el algoritmo original.

4. Marco experimental

4.1. Métodos en la comparativa

Para empezar vamos a utilizar el método descrito en la sección 3, con todas sus variantes:

1. **KNN con medidas de disimilitud**
2. **KNN con medidas de disimilitud + selección de instancias**
3. **KNN con medidas de disimilitud + selección de características**
4. **KNN con medidas de disimilitud + selección de instancias + selección de características**

También se van a utilizar el algoritmo KNN original, con todas las posibles variantes:

5. **KNN original**
6. **KNN original + selección de instancias**
7. **KNN original + selección de características**
8. **KNN original + selección de instancias + selección de características.**

Por otra parte, también se ha implementado el algoritmo KNN con pesos. Se utiliza el algoritmo genético descrito en la sección 3. En este caso, por cada atributo tenemos un gen (en la parte de los reales, la parte de los binarios es igual) que será el peso que se le da a cada atributo. La distancia que se utiliza es la distancia Euclidea, multiplicando cada distancia entre atributos por un peso $w\epsilon[0,1]$.

$$D(X,Y) = \sum_{i=1}^{n} w_i * (|X(i) - Y(i)|)^2$$

Para elegir un método u otro, se indica por medio de un parámetro. Tenemos otros cuatro métodos:

9. **KNN pesos**
10. **KNN pesos + selección de instancias**
11. **KNN pesos + selección de características**
12. **KNN pesos + selección de instancias + selección de características**

Además se tiene la posibilidad de ejecutar el algoritmo genético con selección de instancias en paralelo. Hasta ahora todos los valores de los métodos de selección se calculaban conjuntamente con los valores reales. Pero se ha añadido la opción de

ejecutar primero selección con el algoritmo KNN original, y después aprender los parámetros para las distancias. En la comparativa, únicamente se utiliza la selección de instancias en paralelo. Se indica por parámetro la elección entre serie y paralelo.

13. **KNN disimilitudes + selección de instancias en paralelo** (primero selección de instancias con KNN original y después KNN disimilitudes)

Por último, para tener más métodos para la comparativa se han utilizado varios métodos ya implementados por KEEL. Se dividen en 2 grupos, los de selección de instancias y los de selección de características:

Selección de características:

14. **CW**
15. **FS-GGA**
16. **FS-SSGA**

Selección de instancias:

17. **IS-AllKNN**
18. **IS-CHC**
19. **IS-ENN**
20. **IS-RNG**
21. **IS-SSMA**

4.2. Datasets

En este trabajo, para calcular las distancias se utilizan medidas de disimilitud, que como se ha dicho, son función cuya entrada son números reales en el rango $[0,1]$. Por lo tanto todos los datos que utilicemos tienen que cumplir esta condición. Cuando leemos un fichero, los datos no tienen por qué cumplir esta condición, por lo tanto, a la hora de leer los datos tenemos que hacer una conversión. Tenemos 3 posibles tipos de datos:

- **Reales:** En este caso se nos indica en la cabecera del dataset en que rango están estos datos. Con esta información, utilizamos la siguiente función para convertir los números al rango que necesitamos:

$$f(x) = \frac{x - min}{max - min}$$

Con esta función, si x es igual que el minimo, la función devuelve 0, y si es el máximo 1.

- **Enteros:** Como en el caso anterior, se nos dice el rango en el que están los datos. Utilizamos la función anterior para convertir los números enteros a reales en el rango $[0,1]$.
- **Nominales:** En este caso, en la cabecera se indica todos los posibles valores que puede tomar este atributo. Para su paso a reales utilizamos la siguiente función:

$$f(i) = \frac{i}{n-1}$$

Donde n es el número de valores diferentes que puede tomar e i es el índice del valor que deseamos convertir (el primer elemento tiene como índice 0, y el ultimo $n-1$). De esta manera el primer valor de la lista se convierte a 0, y el último a 1. El resto en números reales dentro del rango deseado.

En la siguiente tabla se muestran los datasets utilizados, con la cantidad de atributos de cada tipo, el número de ejemplos y el número de clases.

Dataset	Ejemplos	Real	Integer	Nominal	Clases
abalone	418	7	0	1	28
autos	159	15	0	10	6
balance	625	4	0	0	3
car	1728	0	0	6	4
cleveland	297	13	0	0	5
contraceptive	1473	0	9	0	3
dermatology	358	0	34	0	6
ecoli	336	7	0	0	8
glass	214	9	0	0	7
hayes-roth	132	0	4	0	3
iris	150	4	0	0	3
led7digit	500	7	0	0	10
letter	2000	0	16	0	26
newthyroid	215	4	1	0	3
pageblocks	548	4	6	0	5
penbased	1100	0	16	0	10
satimage	643	0	36	0	7
segment	2310	19	0	0	7
shuttle	2175	0	9	0	7
tae	151	0	5	0	3
thyroid	720	6	15	0	3
vehicle	846	0	18	0	4
vowel	990	10	3	0	11

wine	178	13	0	0	3
yeast	1484	8	0	0	10
appendicitis	106	7	0	0	2
australian	690	3	5	6	2
automobile	150	15	0	10	6
banana	5300	2	0	0	2
bands	365	13	6	0	2
bupa	345	1	5	0	2
crx	653	3	3	9	2
german	1000	0	7	13	2
ionosphere	351	32	1	0	2
mammographic	830	0	5	0	2
monk-2	432	0	6	0	2
movement_libras	360	90	0	0	15
phoneme	5404	5	0	0	2
saheart	462	5	3	1	2
sonar	208	60	0	0	2
spectfheart	267	0	44	0	2
titanic	2201	3	0	0	2
wdbc	569	30	0	0	2

Tabla 1. Dataset utilizados

Todos estos datasets están particionados en 5 partes (folders). Cada una de estas partes ha sido creada del dataset original, incluyendo una parte del mismo. Luego a la hora de ejecutar el algoritmo, se hace 5 veces, una por cada folder, y el resultado final es la agregación de estos 5 resultados, utilizando la media aritmética.

Los datasets en los que nos hemos centrado son datasets con atributos numéricos ya que la disimilitud no tiene tanto sentido cuando tenemos atributos nominales, aunque hemos utilizado alguno con este tipo de atributos.

4.3. Configuración de parámetros para cada método

Por cada uno de los métodos, se van a listar sus parámetros. Todos los métodos han sido ejecutados con $k = 1$, es decir, solo se tiene en cuenta el vecino más cercano aunque la implementación admite cualquier valor de k.

La función que se ha creado para ejecutar el algoritmo genético recibe 6 parámetros a parte del dataset que se quiere ejecutar. Los parámetros son:

- **L:** Tamaño de la población, es decir, número de individuos para el algoritmo genético.
- **n_iters:** Numero de iteraciones que va a realizar el algoritmo genético.
- **k:** Numero de vecinos.
- **op:** Valor que indica que tipo de aprendizaje de distancias usar. El valor 1 indica distancias basadas en medidas de disimilitud, y el valor 2 indica KNN con pesos.
- **algs:** es un vector de 3 elementos, cuyos valores pueden ser 0 o 1. La posición 1 indica el aprendizaje de distancias (ya sea pesos o disimilitudes), la segunda la selección de instancias y la ultima la selección de características. Si tenemos un 1 decimos que utilizamos el método correspondiente y un 0 indica que no.
- **sep:** valor que indica si queremos hacer el proceso en serie o en paralelo. Un 1 indica que queremos hacerlo en paralelo, cualquier otro valor en serie.

Todas las ejecuciones que se hagan con el algoritmo implementado van a tener ciertos parámetros iguales. El número de individuos (L) va a ser 50, k (número de vecinos) ya se ha dicho que va a valer 1 y n_iters (número de iteraciones) va a ser 10000. El resto varía según el método:

1. **KNN original:** El valor de op no es necesario, ya que no se aprenden distancias, el valor algs será $[0,0,0]$ (no utilizar ningún tipo de aprendizaje) y sep sera 0 para hacerlo en serie.
2. **KNN original + selección de instancias:** Igual que el método anterior excepto algs que es $[0,1,0]$
3. **KNN original + selección de características:** Igual que los anteriores excepto algs que es $[0,0,1]$
4. **KNN original + selección de instancias + selección de características:** Igual que los anteriores excepto algs que es $[0,1,1]$
5. **KNN con pesos:** El valor de op tiene que ser 2 para utilizar pesos, algs $[1,0,0]$ y sep 0.
6. **KNN con pesos + selección de instancias:** Igual que el anterior pero con algs $[1,1,0]$.
7. **KNN con pesos + selección de características**: Igual que el anterior pero con algs $[1,0,1]$.
8. **KNN con pesos + selección de instancias + selección de características**: Igual pero con algs $[1,1,1]$.
9. **KNN con disimilitudes:** Para empezar todos los métodos utilizan op 1 para utilizar disimilitudes y sep 0 para hacerlo todo en serie. Luego dependiendo si se quiere utilizar selección de instancias y/o selección de características se empleara el parámetro algs igual que en el KNN con pesos dependiendo de lo que se quiera usar.

10. **KNN con disimilitudes + selección de instancias en paralelo:** Igual que el método de KNN con disimilitudes + selección de instancias excepto sep que será 1.
11. **Métodos de KEEL:** Estos métodos tienen varios parámetros pero difieren unos de otros. Todos los para metros k serán 1 como se ha dicho, excepto en 2 métodos. En los métodos IS-AllKNN y IS-ENN, se utiliza un k con valor 3, ya que este lo emplea para hacer selección de instancias y después el algoritmo KNN clasificara utilizando k con valor 1. El resto de parámetros son los valores por defecto que trae KEEL, excepto para los de tamaño de la población que es 50 como ya se ha dicho, y el número de iteraciones 10000.

4.4. Test estadísticos y evaluación

En este trabajo se utilizan dos tipos de test para poder decir cuál de los métodos anteriormente citados es mejor. Estos test se basan en un estudio estadístico, y compara diferentes métodos diciendo cual es mejor. Los test son el test de Wilcoxon y el test de Friedman.

Test de Wilcoxon: Este test se utiliza para comparar un método frente a otro. Para comprobar cuál de los dos métodos es mejor utiliza los datos obtenidos en los diferentes conjuntos utilizados en la ejecución. Cuantos más datasets, más preciso será el resultado. Este test, va comparando los resultados obtenidos en los diferentes datasets y va asignando rangos al algoritmo que gana en cada caso. Después con los resultados de estos rangos se calcula un número llamado p_{valor} el cual indica si hay independencia estadística. Si el p_{valor} es menor o igual a 0.05, entonces el método con más rangos es mejor estadísticamente que el otro. En caso contrario no se puede afirmar que haya diferencias estadísticas.

Test de Friedman: Este test, es utilizado para comparar varios métodos, al contrario que el anterior que solo compara 2. Al igual que en el anterior, hace uso de los datos obtenidos en los diferentes conjuntos utilizados en la ejecución, cuantos más datasets, más preciso será el resultado. Este método va calculando los rangos y se calcula también un p_{valor} que indica si hay diferencia estadística o no. En este caso, si p_{valor} es menor o igual a 0.05, se dice que el método con menos rangos (en el anterior era el que más) es estadísticamente mejor y en caso contrario no se puede afirmar que sea mejor.

5. Estudio experimental

5.1. Introducción y objetivos

El objetivo de este proyecto es mejorar el ratio de clasificación del algoritmo KNN aprendiendo las distancias. A parte de mejorar el algoritmo original de KNN, queremos ver si el propuesto supera también al ya existente algoritmo de KNN con pesos en los atributos. También se han expuesto métodos de selección de instancias y de características, vamos a ver cuál de todas las variantes es mejor. Y con todo esto, también utilizaremos métodos existentes implementados en KEEL para tener más métodos en la comparativa.

Con todos los métodos comentados tenemos 21 diferentes métodos a comparar, por lo tanto no podemos hacer una comparativa todos frente a todos, vamos a realizar una comparativa por fases utilizando los test de Wilcoxon y de Friedman anteriormente citados.

5.2. Resultados

En este trabajo se van a exponer 5 tipos de tablas: una para los resultados de entrenamiento, otra para los resultados de test, otras 2 con los ratios de reducción en selección de instancias y características, y otra con los tiempos de ejecución. Los métodos de KEEL únicamente aparecerán en las dos primeras: resultados de entrenamiento y de test. En los casos de los resultados de entrenamiento, muestran el porcentaje de aciertos obtenido con un valor entre 0 y 1. En los casos de selección, muestran el ratio de reducción correspondiente, también con un número entre 0 y 1. Por último, la tabla de los tiempos, muestran el tiempo que han tardado en ejecutar cada método en cada dataset. Este tiempo se muestra en segundos.

En las columnas aparecerá la combinatoria del parámetro algs comentado en la sección 4.3, junto al parámetro op. Los métodos de selección de instancias y de características solo aparecerán donde se haya usado el método, ya que si no se utiliza el método el ratio de reducción es 0.

Las filas de las tablas corresponden a los datasets, y las columnas a cada método. Aparte de los datasets, se ha agregado una fila con la media de los resultados.

Una de las cosas que se pueden observar, es que el método propuesto (KNN con disimilitudes) siempre consigue un mejor resultado o como mínimo igual que el KNN original (columnas 1 y 5 de la primer parte de los datos de entrenamiento). Como se dijo en la sección 3.1, en el algoritmo genético se añade un cromosoma que utiliza la misma distancia que el algoritmo KNN original. Por lo tanto, si el algoritmo genético no consigue encontrar una solución mejor, como mínimo va a tener la misma capacidad que el KNN original.

	0 0 0	0 0 1 (FS)	0 1 0 (IS)	0 1 1 (IFS)	1 0 0	
	op = 2	op = 2	op = 2	op = 2	op=1(disimil.)	op = 2
abalone	,1353	,2249	,4180	,4133	,2658	,2703
autos	,7280	,8837	,8163	,7955	,9386	,9356
balance	,7776	,7776	,8900	,8932	,8448	,7928
car	,8666	,9274	,8694	,8883	,9773	,9531
cleveland	,5236	,5985	,6448	,6439	,6507	,6600
contraceptive	,4252	,4817	,5326	,5614	,5243	,5178
dermatology	,9539	,9797	,9721	,9588	,9965	,9993
ecoli	,7828	,7843	,8379	,8022	,8624	,8327
glass	,7078	,7757	,7640	,7780	,8808	,8538
hayes-roth	,7083	,8296	,7592	,8448	,8599	,7747
iris	,9550	,9700	,9850	,9767	,9817	,9783
led7digit	,5934	,5943	,7517	,7447	,5934	,6049
letter	,8164	,8575	,7341	,7533	,8839	,8913
newthyroid	,9628	,9698	,9628	,9698	,9919	,9872
pageblocks	,9361	,9562	,9439	,9489	,9772	,9644
penbased	,9716	,9725	,9373	,9102	,9875	,9873
satimage	,8624	,8892	,8799	,8573	,9137	,9238
segment	,9683	,9748	,9304	,9361	,9922	,9885
shuttle	,9962	,9970	,9948	,9956	,9988	,9987
tae	,6092	,6175	,6970	,6787	,7301	,7350
thyroid	,8937	,9781	,9271	,9719	,9781	,9875
vehicle	,6892	,7408	,7012	,7216	,7931	,7899
vowel	,9879	,9894	,8715	,8381	,9960	,9970
wine	,9593	,9958	,9916	,9747	1,0000	,9986
yeast	,5189	,5219	,5878	,5822	,5719	,5642
appendicitis	,8019	,8798	,8892	,9056	,9316	,9104
australian	,8076	,8533	,8739	,8663	,8793	,8772
automobile	,7293	,8803	,8301	,7985	,9433	,9449
banana	,8682	,8682	,8995	,8971	,8792	,8739
bands	,7082	,7630	,7363	,7295	,8651	,8630
bupa	,6232	,6558	,7297	,7333	,7420	,7348
crx	,8113	,8595	,8725	,8668	,8882	,8940
german	,6903	,7338	,7470	,7555	,7873	,7955
ionosphere	,8640	,9359	,9174	,9088	,9822	,9672
mammographic	,7521	,7732	,8389	,8352	,7994	,7831
monk-2	,7714	1,0000	,8379	,9716	1,0000	1,0000
movement_libras	,8360	,8694	,8202	,8204	,8902	,8840
phoneme	,8961	,8961	,8369	,8354	,9155	,9133
saheart	,6423	,6818	,7749	,7473	,7706	,7462
sonar	,8606	,9243	,8643	,8570	,9772	,9664
spectfheart	,7013	,8324	,8258	,8436	,9195	,9260
titanic	,7354	,7357	,7918	,7833	,7360	,7354
wdbc	,9574	,9758	,9684	,9745	,9908	,9934
MEDIA	**,7764**	**,8234**	**,8245**	**,8272**	**,8625**	**,8557**

Tabla 2. 1º parte de train

	1 0 1 (FS)		1 1 0 (IS)		1 1 1 (IFS)		IS-REFS
	op=1 disimil.	op = 2	op=1 disimil.	op = 2	op=1 disimil.	op = 2	op=1 disimil.
abalone	,2460	,2453	,5018	,4343	,3902	,4124	,4573
autos	,9214	,9278	,8601	,8429	,8079	,8188	,8554
balance	,8460	,7928	,9096	,9052	,9004	,8996	,8992
car	,9770	,9531	,9372	,9226	,9489	,9220	,9031
cleveland	,6187	,6061	,6827	,6776	,6397	,6406	,6860
contraceptive	,5287	,5039	,5796	,5935	,5933	,5855	,5779
dermatology	,9888	,9881	,9951	,9881	,9651	,9707	,9902
ecoli	,8542	,8312	,8654	,8521	,7992	,8119	,8550
glass	,8621	,8445	,8459	,8084	,7770	,7886	,8131
hayes-roth	,8580	,7917	,8921	,8618	,8940	,8655	,8372
iris	,9783	,9783	,9800	,9917	,9650	,9783	,9917
led7digit	,5943	,6040	,7593	,7624	,7444	,7369	,7536
letter	,8841	,8861	,6924	,7347	,6699	,7311	,8030
newthyroid	,9860	,9884	,9953	,9930	,9802	,9826	,9849
pageblocks	,9694	,9631	,9767	,9662	,9544	,9576	,9640
penbased	,9843	,9839	,9116	,9346	,8736	,9214	,9564
satimage	,9071	,9160	,8732	,8822	,8472	,8737	,9024
segment	,9889	,9847	,9500	,9556	,9573	,9571	,9563
shuttle	,9977	,9978	,9963	,9963	,9963	,9963	,9960
tae	,7185	,7085	,6920	,7003	,6587	,6803	,7102
thyroid	,9875	,9830	,9802	,9795	,9726	,9740	,9313
vehicle	,7713	,7710	,7184	,7580	,7092	,7597	,7653
vowel	,9955	,9960	,8351	,8187	,7593	,8192	,9212
wine	,9958	,9986	,9930	,9916	,9860	,9846	,9986
yeast	,5655	,5568	,6043	,6216	,5878	,6147	,6103
appendicitis	,9152	,9057	,9316	,9222	,9152	,9151	,9293
australian	,8703	,8775	,8924	,8906	,8681	,8725	,8902
automobile	,9326	,9244	,8616	,8461	,8161	,8097	,8612
banana	,8790	,8739	,8984	,9012	,8990	,9026	,9033
bands	,8349	,8130	,7829	,7699	,7116	,7315	,7925
bupa	,7087	,7275	,7652	,7761	,7428	,7478	,7652
crx	,8871	,8832	,9020	,8901	,8641	,8710	,8905
german	,7360	,7323	,7800	,7890	,7575	,7548	,7848
ionosphere	,9843	,9615	,9558	,9444	,9231	,9202	,9544
mammographic	,7925	,7865	,8618	,8596	,8497	,8536	,8539
monk-2	1,0000	1,0000	,9792	1,0000	,9942	1,0000	,9722
movement_libras	,9007	,9041	,7588	,7686	,7601	,7603	,8436
phoneme	,9145	,9123	,8369	,8425	,8480	,8411	,8403
saheart	,6975	,7219	,7873	,7857	,7533	,7565	,7901
sonar	,9675	,9615	,9062	,8654	,8306	,8245	,9387
spectfheart	,8811	,8914	,8344	,8576	,7987	,8305	,9064
titanic	,7360	,7357	,7918	,7918	,7863	,7863	,7916
wdbc	,9842	,9815	,9886	,9837	,9701	,9758	,9895
MEDIA	**,8523**	**,8464**	**,8498**	**,8479**	**,8248**	**,8334**	**,8562**

Tabla 3. 2º parte de train

	CW	FS-GGA	FS-SSGA	IS-ALLKNN	IS-CHC	IS-ENN	IS-RNG	IS-SSMA
abalone	,1353	,2249	,2172	,6406	,2873	,7545	,2707	,3142
autos	,7280	,7328	,5672	,9788	,6745	,9531	,7125	,7409
balance	,6496	,7776	,6200	,9978	,9072	,9813	,8644	,9068
car	,7458	,8238	,7157	,9884	,8827	,9489	,8764	,9494
cleveland	,2863	,5639	,5084	,9763	,6271	,9621	,6070	,6297
contraceptive	,4270	,4732	,4530	,9009	,5384	,8983	,5642	,5905
dermatology	,3032	,9790	,4945	,9985	,9651	,9842	,9714	,9783
ecoli	,7828	,7843	,7367	,9856	,8283	,9833	,8461	,8528
glass	,7008	,7687	,7266	,9873	,7032	,9685	,7441	,7580
hayes-roth	,2972	,3921	,3921	,6808	,6610	,7128	,4471	,6666
iris	,9567	,9600	,9600	1,0000	,9750	1,0000	,9583	,9833
led7digit	,3686	,3305	,0946	,7121	,6364	,7399	,5264	,3865
letter	,8110	,8678	,3059	,9730	,5385	,9586	,8132	,7356
newthyroid	,9628	,9698	,9663	,9962	,9802	,9963	,9651	,9814
pageblocks	,9334	,9539	,9503	,9975	,9402	,9976	,9434	,9430
penbased	,9716	,9750	,7205	,9988	,9114	,9955	,9727	,9657
satimage	,8624	,9063	,7904	,9889	,8686	,9817	,8943	,8970
segment	,9639	,9686	,9614	,9964	,9226	,9933	,9654	,9641
shuttle	,9962	,9883	,9930	1,0000	,9931	,9995	,9960	,9945
tae	,4188	,4186	,3938	,8504	,5960	,8679	,5709	,5959
thyroid	,1212	,9792	,9705	,9963	,9274	,9962	,9316	,9378
vehicle	,6886	,7485	,6419	,9610	,6785	,9380	,7503	,7754
vowel	,9879	,9907	,8957	,9968	,7045	,9961	,9848	,8361
wine	,9578	,9902	,9438	,9970	,9761	,9868	,9635	,9874
yeast	,5007	,5057	,4348	,9493	,6038	,9269	,6402	,6582
appendicitis	,8019	,8798	,8633	,9968	,8939	,9971	,8915	,8963
australian	,7696	,8558	,8134	,9921	,8826	,9914	,8765	,8851
automobile	,7293	,7464	,4355	,9691	,6666	,9523	,6982	,7135
banana	,6784	,8683	,8683	,9963	,9003	,9945	,9156	,9141
bands	,4699	,7781	,5555	,9445	,7315	,9322	,7685	,7555
bupa	,6094	,6312	,6196	,9355	,7022	,8979	,7348	,7717
crx	,8097	,8541	,7994	,9924	,8733	,9895	,8760	,8886
german	,3153	,7383	,6998	,9760	,7635	,9576	,7605	,7995
ionosphere	,6588	,9487	,8504	,9929	,9053	,9773	,8825	,9302
mammographic	,7545	,7581	,7587	,9878	,8440	,9699	,8334	,8509
monk-2	,8409	,7778	,7778	,9593	,9734	,8422	,7813	,9543
movement_libras	,5321	,8603	,4936	,9657	,6522	,9379	,8035	,7491
phoneme	,8935	,8935	,8498	,9931	,8313	,9866	,9109	,8890
saheart	,6412	,6786	,6569	,9769	,7625	,9524	,7608	,7933
sonar	,6027	,9724	,7404	,9893	,8390	,9623	,8774	,8703
spectfheart	,2060	,8689	,7865	,9653	,8146	,9543	,8118	,8483
titanic	,6711	,6711	,6711	1,0000	,7902	,8726	,7368	,7509
wdbc	,9569	,9811	,9438	,9991	,9750	,9936	,9714	,9807
MEDIA	,6628	,7869	,6893	,9577	,7937	,9461	,8063	,8202

Tabla 4. 3º parte de train

	0 0 0	0 0 1 (FS)	0 1 0 (IS)	0 1 1 (IFS)	1 0 0	
	op = 2	op = 2	op = 2	op = 2	op=1(disimil.)	op = 2
abalone	,1571	,1963	,2551	,2389	,1892	,2092
autos	,7487	,8565	,7252	,7106	,9075	,8513
balance	,8029	,8029	,8735	,8800	,8495	,7853
car	,8883	,9305	,8495	,8831	,9658	,9612
cleveland	,5524	,5114	,5761	,5724	,5186	,5358
contraceptive	,4406	,4725	,4575	,5269	,4739	,4732
dermatology	,9550	,9609	,9385	,9355	,9691	,9608
ecoli	,7894	,7626	,8175	,7785	,7775	,7623
glass	,7128	,7332	,6741	,6782	,7895	,7707
hayes-roth	,7451	,8571	,6143	,7808	,8187	,7423
iris	,9467	,9467	,9667	,9400	,9467	,9333
led7digit	,5188	,5163	,6390	,6528	,5188	,5113
letter	,8364	,8702	,6466	,7050	,8732	,8787
newthyroid	,9674	,9674	,9395	,9256	,9581	,9395
pageblocks	,9401	,9435	,9363	,9455	,9490	,9364
penbased	,9737	,9619	,9182	,8819	,9718	,9746
satimage	,8786	,8726	,8603	,8258	,8741	,8678
segment	,9723	,9706	,9178	,9277	,9844	,9784
shuttle	,9936	,9977	,9945	,9950	,9968	,9972
tae	,6148	,5557	,5337	,4946	,6277	,6010
thyroid	,9028	,9791	,9251	,9709	,9695	,9556
vehicle	,7022	,7056	,6218	,6680	,7056	,7141
vowel	,9939	,9828	,8374	,7980	,9848	,9919
wine	,9546	,9438	,9496	,9668	,9779	,9435
yeast	,5465	,5317	,5654	,5640	,5330	,5324
appendicitis	,8208	,8303	,8589	,8403	,8403	,8398
australian	,8102	,8247	,8521	,8390	,8015	,8204
automobile	,7468	,8366	,7174	,6939	,9186	,8568
banana	,8806	,8806	,8951	,8919	,8770	,8770
bands	,6904	,6849	,6301	,6575	,7562	,7342
bupa	,6145	,6058	,6203	,7043	,5797	,6203
crx	,8194	,8162	,8546	,8592	,8101	,7994
german	,6830	,6850	,7060	,7110	,7050	,6790
ionosphere	,8774	,9030	,9002	,8831	,9002	,9115
mammographic	,7700	,7832	,8193	,8254	,7724	,7748
monk-2	,7548	1,0000	,7688	,9723	1,0000	1,0000
movement_libras	,8713	,8833	,7413	,7133	,8400	,8687
phoneme	,9108	,9108	,8249	,8222	,9134	,9134
saheart	,6625	,5756	,7273	,7144	,6105	,6537
sonar	,8562	,8563	,7496	,7948	,8562	,8757
spectfheart	,6964	,7154	,7527	,7867	,7377	,7196
titanic	,7344	,7344	,7877	,7808	,7344	,7344
wdbc	,9508	,9508	,9421	,9509	,9615	,9544
MEDIA	**,7834**	**,7978**	**,7717**	**,7834**	**,8080**	**,8010**

Tabla 5. 1º parte de test

	1 0 1 (FS)		1 1 0 (IS)		1 1 1 (IFS)		IS-REFS
	op=1(disimil.)	op = 2	op=1(disimil.)	op = 2	op=1(disimil.)	op = 2	op = 1
abalone	,2194	,1720	,2142	,2046	,2204	,2516	,2262
autos	,8642	,8500	,6887	,7233	,7338	,6977	,7161
balance	,8416	,7853	,8945	,8897	,8800	,8704	,8783
car	,9682	,9612	,9282	,9167	,9421	,9091	,8900
cleveland	,5084	,5288	,5996	,5795	,5794	,5493	,5761
contraceptive	,4806	,4820	,5261	,5452	,5431	,5607	,5105
dermatology	,9500	,9609	,9637	,9667	,9496	,9359	,9472
ecoli	,7649	,7539	,8202	,8082	,7789	,7692	,7934
glass	,7791	,7569	,7444	,7537	,7071	,6840	,7395
hayes-roth	,8495	,7956	,8418	,7742	,8341	,7879	,7813
iris	,9400	,9400	,9400	,9667	,9333	,9333	,9667
led7digit	,5163	,5089	,6505	,6296	,6313	,6024	,6396
letter	,8718	,8767	,6307	,6953	,6160	,6811	,7358
newthyroid	,9674	,9302	,9814	,9767	,9349	,9488	,9442
pageblocks	,9491	,9416	,9580	,9528	,9527	,9490	,9473
penbased	,9600	,9664	,8930	,9129	,8420	,9012	,9292
satimage	,8415	,8430	,8293	,8386	,8137	,8321	,8603
segment	,9831	,9788	,9450	,9450	,9485	,9468	,9407
shuttle	,9968	,9954	,9963	,9963	,9963	,9954	,9950
tae	,5946	,6749	,5601	,5684	,5219	,5030	,5742
thyroid	,9708	,9652	,9764	,9681	,9611	,9667	,9292
vehicle	,6868	,7234	,6703	,6916	,6608	,6785	,6573
vowel	,9838	,9919	,7980	,7848	,7111	,7960	,8778
wine	,9665	,9495	,9603	,9614	,9663	,9497	,9601
yeast	,5188	,5322	,5789	,5844	,5511	,5743	,5796
appendicitis	,8307	,8299	,8398	,8403	,8403	,8494	,8113
australian	,8073	,8175	,8522	,8493	,8449	,8551	,8464
automobile	,8729	,8357	,7468	,7436	,7170	,7222	,6948
banana	,8772	,8775	,8945	,8974	,8926	,8974	,8994
bands	,6822	,6767	,6849	,6795	,6384	,6521	,7014
bupa	,5739	,6696	,6899	,6464	,6841	,6754	,6667
crx	,7993	,8224	,8684	,8638	,8638	,8607	,8577
german	,7030	,6880	,7410	,7360	,7150	,7210	,7090
ionosphere	,9258	,9087	,9059	,8946	,9087	,8803	,9202
mammographic	,7652	,7735	,8374	,8399	,8411	,8362	,8375
monk-2	1,0000	1,0000	,9746	1,0000	,9953	1,0000	,9723
movement_libras	,8480	,8313	,6353	,6733	,6373	,6733	,7353
phoneme	,9123	,9095	,8331	,8357	,8279	,8327	,8316
saheart	,6277	,5866	,7057	,7036	,6926	,6991	,7209
sonar	,8033	,7936	,7742	,7746	,7071	,7507	,8224
spectfheart	,7598	,7754	,7903	,7751	,7940	,7867	,7901
titanic	,7344	,7344	,7860	,7877	,7803	,7803	,7865
wdbc	,9403	,9385	,9544	,9631	,9456	,9597	,9597
MEDIA	**,8008**	**,7985**	**,7931**	**,7939**	**,7799**	**,7839**	**,7944**

Tabla 6. 2º parte de test

	CW	FS-GGA	FS-SSGA	IS-ALLKNN	IS-CHC	IS-ENN	IS-RNG	IS-SSMA
abalone	,1571	,1895	,2022	,2368	,2364	,2529	,2504	,2368
autos	,7487	,7035	,5444	,5870	,1884	,5791	,1855	,1761
balance	,6574	,8029	,6239	,8784	,8816	,8735	,8783	,8848
car	,7548	,8258	,7159	,8808	,4961	,8756	,7193	,4161
cleveland	,3061	,5152	,5258	,5724	,5928	,5691	,5692	,5862
contraceptive	,4345	,4501	,4658	,4630	,4977	,4623	,4766	,5159
dermatology	,2984	,9583	,4914	,9691	,8997	,9578	,9186	,9155
ecoli	,7894	,7626	,7029	,8348	,7989	,8191	,8377	,8353
glass	,7082	,7332	,7376	,6686	,6502	,6869	,6867	,7149
hayes-roth	,3170	,4159	,4159	,4863	,5555	,5104	,3945	,6132
iris	,9467	,9533	,9467	,9533	,9600	,9600	,9600	,9667
led7digit	,4126	,3726	,0992	,3150	,5596	,4510	,5074	,4157
letter	,8315	,8822	,3206	,7591	,5341	,7649	,7809	,6774
newthyroid	,9674	,9674	,9535	,9581	,9442	,9535	,9581	,9581
pageblocks	,9419	,9435	,9434	,9399	,9400	,9399	,9417	,9382
penbased	,9737	,9627	,7173	,9618	,9119	,9682	,9664	,9356
satimage	,8786	,8789	,7698	,8680	,8479	,8726	,8805	,8495
segment	,9688	,9636	,9610	,9511	,9251	,9524	,9602	,9507
shuttle	,9936	,9876	,9936	,9904	,9940	,9899	,9904	,9936
tae	,4359	,4165	,3640	,5072	,4826	,4748	,5208	,5415
thyroid	,1606	,9791	,9722	,9278	,9264	,9292	,9334	,9306
vehicle	,6999	,7034	,6314	,6811	,6146	,6939	,6810	,6491
vowel	,9939	,9859	,9242	,9737	,7657	,9758	,9747	,8990
wine	,9490	,9609	,9099	,9546	,9660	,9546	,9604	,9496
yeast	,5323	,5115	,4137	,5749	,5971	,5655	,5817	,5938
appendicitis	,8208	,8303	,8494	,8779	,8494	,8970	,8775	,8684
australian	,7810	,8174	,8045	,8494	,8362	,8348	,8406	,8464
automobile	,7468	,7264	,4375	,5343	,2377	,5403	,1843	,2191
banana	,6826	,8809	,8809	,9045	,8977	,9062	,9009	,9023
bands	,5014	,7178	,5507	,6822	,6630	,6932	,6740	,6548
bupa	,6029	,5971	,5652	,6319	,5855	,6464	,6551	,6609
crx	,8210	,8377	,7963	,8592	,6525	,8607	,8103	,6556
german	,3120	,6840	,6930	,7100	,6920	,7170	,6950	,6450
ionosphere	,6610	,9115	,8092	,8461	,8832	,8546	,8631	,9060
mammographic	,7542	,7579	,7615	,7892	,8314	,8000	,8036	,8387
monk-2	,8379	,7779	,7779	,7615	,9768	,7686	,7802	,9584
movement_libras	,5713	,8567	,5313	,7160	,6113	,7400	,7660	,7280
phoneme	,9084	,9082	,8553	,8845	,8261	,8916	,9008	,8640
saheart	,6625	,5974	,6320	,6970	,6971	,7013	,6948	,6971
sonar	,6039	,8699	,6681	,7744	,7501	,8028	,8516	,7987
spectfheart	,2060	,7305	,7413	,7266	,7674	,7414	,7567	,7829
titanic	,6604	,6604	,6604	,6663	,7844	,7120	,7284	,7384
wdbc	,9526	,9595	,9193	,9632	,9631	,9596	,9579	,9579
MEDIA	**,6731**	**,7662**	**,6809**	**,7620**	**,7272**	**,7698**	**,7501**	**,7411**

Tabla 7. 3º parte de test

	0 1 0 (IS)	0 1 1 (IFS)	1 1 0 (IS)		1 1 1 (IFS)		IS-REFS
	op = 2	op = 2	op=1(disimil.)	op = 2	op=1(disimil.)	op = 2	op = 1
abalone	,8368	,8511	,7471	,8501	,9064	,9093	,8102
autos	,8806	,8853	,8997	,8978	,9119	,8949	,8774
balance	,9880	,9856	,9872	,9888	,9852	,9856	,9884
car	,9727	,9779	,9780	,9845	,9769	,9782	,9721
cleveland	,9074	,9756	,9806	,9798	,9815	,9747	,9065
contraceptive	,9732	,9747	,9840	,9881	,9812	,9863	,9701
dermatology	,9672	,9497	,9756	,9756	,9756	,9742	,9672
ecoli	,9554	,9621	,9725	,9688	,9762	,9792	,9547
glass	,9089	,8867	,9451	,9393	,9510	,9334	,8984
hayes-roth	,7354	,7766	,9091	,9053	,9167	,9053	,7921
iris	,9600	,9667	,9667	,9667	,9667	,9683	,9600
led7digit	,9606	,9548	,9664	,9685	,9678	,9726	,9535
letter	,8380	,8999	,8510	,8764	,8821	,8828	,8542
newthyroid	,9558	,9430	,9767	,9779	,9744	,9756	,9500
pageblocks	,9726	,9790	,9881	,9900	,9904	,9886	,9713
penbased	,9552	,9504	,9630	,9609	,9666	,9661	,9575
satimage	,9635	,9681	,9775	,9775	,9813	,9806	,9654
segment	,9728	,9817	,9869	,9887	,9868	,9886	,9753
shuttle	,9936	,9951	,9976	,9976	,9977	,9975	,9934
tae	,8989	,9107	,9188	,9089	,9405	,9206	,9040
thyroid	,9823	,9708	,9924	,9920	,9931	,9924	,9826
vehicle	,9368	,9439	,9699	,9722	,9770	,9708	,9294
vowel	,8003	,8465	,7760	,8439	,8644	,8389	,8025
wine	,9649	,9677	,9775	,9761	,9761	,9663	,9635
yeast	,9715	,9727	,9850	,9872	,9774	,9830	,9742
appendicitis	,8631	,8326	,9623	,9646	,9599	,9646	,8561
australian	,9772	,9830	,9902	,9909	,9938	,9924	,9808
automobile	,8790	,8903	,9025	,8946	,8979	,9009	,8868
banana	,9924	,9928	,9930	,9950	,9929	,9942	,9917
bands	,9479	,9390	,9836	,9829	,9849	,9795	,9493
bupa	,9536	,9710	,9841	,9790	,9819	,9797	,9601
crx	,9885	,9916	,9935	,9916	,9943	,9935	,9874
german	,9813	,9808	,9950	,9945	,9925	,9928	,9833
ionosphere	,9672	,9601	,9801	,9793	,9850	,9822	,9644
mammographic	,9831	,9883	,9937	,9934	,9943	,9931	,9825
monk-2	,9468	,9676	,9890	,9826	,9861	,9797	,9306
movement_libras	,8036	,8341	,8404	,8258	,7794	,8298	,8058
phoneme	,9928	,9933	,9881	,9926	,9824	,9891	,9929
saheart	,9832	,9881	,9892	,9892	,9897	,9881	,9848
sonar	,9146	,8870	,9688	,9796	,9820	,9844	,9002
spectfheart	,9363	,9841	,9869	,9860	,9897	,9888	,9298
titanic	,9956	,9967	,9972	,9968	,9977	,9976	,9959
wdbc	,9706	,9925	,9921	,9952	,9938	,9947	,9692
MEDIA	**,9379**	**,9453**	**,9582**	**,9622**	**,9647**	**,9637**	**,9378**

Tabla 8. Reducción de instancias

	0 0 1 (FS)	0 1 1 (IFS)	1 0 1 (FS)		1 1 1 (IFS)	
	op = 2	op = 2	op=1(disimil.)	op = 2	op=1(disimil.)	op = 2
abalone	,6250	,7500	,7250	,5500	,8000	,7500
autos	,8080	,8640	,8480	,8240	,8800	,8640
balance	,0000	,0000	,0000	,0000	,0000	,0000
car	,1667	,3000	,0000	,0000	,1667	,1667
cleveland	,5846	,7538	,6769	,6462	,7538	,8000
contraceptive	,4222	,6667	,4222	,4000	,5778	,6667
dermatology	,5941	,7941	,7588	,7471	,8176	,8176
ecoli	,2286	,5429	,2286	,1429	,6000	,5429
glass	,4667	,6444	,4000	,4667	,7111	,6444
hayes-roth	,2500	,2500	,2500	,2500	,2500	,2500
iris	,5500	,5500	,5000	,4500	,7500	,5500
led7digit	,0286	,0571	,0286	,0286	,0571	,1714
letter	,4375	,4500	,3875	,3500	,5125	,4750
newthyroid	,3200	,4800	,4800	,2400	,6400	,4800
pageblocks	,6600	,9000	,7400	,7000	,9000	,8800
penbased	,2125	,3750	,2125	,2750	,5125	,4000
satimage	,5611	,8667	,7056	,7556	,9278	,8889
segment	,6421	,7895	,7053	,7263	,8421	,8211
shuttle	,7778	,7778	,7778	,7556	,7778	,7778
tae	,2400	,2800	,3200	,1200	,3600	,3600
thyroid	,7238	,9048	,7905	,7905	,9143	,9048
vehicle	,5556	,6333	,6222	,6222	,8222	,6889
vowel	,3538	,4462	,3846	,3692	,4923	,5231
wine	,4769	,6769	,5692	,6000	,7538	,6923
yeast	,1500	,3750	,1500	,2000	,4000	,4000
appendicitis	,5429	,7714	,6857	,5714	,8000	,7429
australian	,5000	,8000	,6429	,5143	,8429	,8000
automobile	,8080	,8960	,8400	,8480	,8800	,8960
banana	,0000	,0000	,0000	,0000	,0000	,0000
bands	,5684	,8421	,7263	,6842	,8737	,8421
bupa	,4667	,6333	,4667	,3000	,6667	,6000
crx	,5733	,9067	,6133	,5467	,9333	,8800
german	,5400	,8400	,8100	,7900	,8800	,8600
ionosphere	,7333	,9030	,7939	,8242	,9152	,9091
mammographic	,4000	,6800	,2800	,2800	,4800	,4400
monk-2	,5000	,6667	,5000	,5000	,5333	,5000
movement_libras	,7222	,8578	,8400	,8844	,7533	,8622
phoneme	,0000	,3200	,0400	,0400	,1200	,1200
saheart	,4667	,7111	,7778	,5778	,7333	,6667
sonar	,6533	,8367	,7767	,8100	,9367	,9467
spectfheart	,7182	,9409	,9227	,8864	,9727	,9591
titanic	,0667	,3333	,0667	,0667	,2667	,2667
wdbc	,6600	,8733	,8467	,8400	,9000	,8933
MEDIA	**,4594**	**,6265**	**,5189**	**,4878**	**,6444**	**,6209**

Tabla 9. Reducción de características

	0 0 0	0 0 1 (FS)	0 1 0 (IS)	0 1 1 (IFS)	1 0 0	
	op = 2	op = 2	op = 2	op = 2	op=1(disimil.)	op = 2
abalone	,0000	9,8077	44,0373	20,0753	507,3769	196,7624
autos	,0000	4,9221	15,0337	5,5341	146,8496	107,6020
balance	,0000	14,7969	28,4977	25,0214	634,7260	175,3677
car	,0000	102,1086	186,1277	177,1597	4856,2456	2712,4402
cleveland	,0000	25,5912	10,9296	12,3945	414,9386	315,0883
contraceptive	,0000	174,4710	278,7045	138,7051	4970,8320	2526,9671
dermatology	,0000	61,5419	64,0083	32,6661	1017,0798	479,7530
ecoli	,0000	12,5276	22,2832	16,2643	354,9043	191,5462
glass	,0000	3,8427	9,3735	6,0509	191,7954	82,6022
hayes-roth	,0000	,9223	1,9182	1,2543	47,8179	21,9009
iris	,0000	,9019	3,4169	2,7426	39,0014	14,3812
led7digit	,0000	28,6441	30,0669	20,6148	512,0224	289,7101
letter	,0000	2400,0478	8022,0596	2169,2395	17520,7754	26516,9742
newthyroid	,0000	5,0479	4,3726	3,9658	179,2466	56,4509
pageblocks	,0000	30,3882	35,2533	20,4647	1431,5182	540,5057
penbased	,0000	367,0100	173,9796	139,3809	6521,0630	2192,6215
satimage	,0000	448,3638	412,9722	94,5824	12200,4854	8829,3580
segment	,0001	1372,3701	821,2018	384,0710	22231,5458	13537,8436
shuttle	,0000	253,4925	369,3175	154,1048	10956,5868	5413,5625
tae	,0000	2,8920	8,2188	7,6051	73,1905	37,7209
thyroid	,0000	90,3873	85,8477	40,1716	2716,6718	1536,8240
vehicle	,0000	184,1934	184,6907	94,9741	4929,6968	1657,9958
vowel	,0000	314,4147	412,4084	251,3974	4703,7129	1560,5427
wine	,0000	5,9700	7,2813	4,3148	23,0732	29,0124
yeast	,0000	144,0330	196,4721	172,7970	4936,9201	2320,7494
appendicitis	,0000	2,0715	1,2599	1,1705	56,2256	26,0051
australian	,0000	61,7753	68,9686	46,1077	3256,9205	2157,4592
automobile	,0000	6,8845	24,1131	9,9508	681,8463	129,9850
banana	,0001	953,9026	1180,9215	899,4705	37105,2413	19088,3556
bands	,0001	69,5917	64,4455	26,1513	1317,6771	763,3577
bupa	,0000	14,9226	27,8686	12,4408	669,3793	381,1384
crx	,0000	102,9428	73,1614	53,0185	2712,9269	1526,9080
german	,0000	225,7394	180,7393	101,0018	5059,9277	2697,2361
ionosphere	,0000	21,4857	41,3117	17,4694	1184,5495	499,6242
mammographic	,0000	28,2509	35,0740	30,2243	985,2631	634,4334
monk-2	,0000	10,0645	21,8918	11,6495	9,0955	1,4236
movement_libras	,0000	101,4035	272,5198	115,7750	3569,9399	1621,6520
phoneme	,0001	1511,9630	1113,7360	1034,1779	31795,6968	19698,1368
saheart	,0000	36,5859	29,8422	15,6771	692,9448	270,2852
sonar	,0000	17,6055	48,8767	14,5248	875,0564	372,5018
spectfheart	,0000	23,7857	27,3556	20,9028	1223,8515	445,8385
titanic	,0001	301,7542	408,0396	279,6771	7029,2146	2659,3995
wdbc	,0001	108,1977	81,9997	37,5489	2913,2572	1141,3071
MEDIA	**,0000**	**224,5957**	**351,8744**	**156,3370**	**4726,9091**	**2917,6588**

Tabla 10. 1º parte del tiempo de ejecución

	1 0 1 (FS)		1 1 0 (IS)		1 1 1 (IFS)		IS-REFS
	op=1(disimil.)	op = 2	op=1(disimil.)	op = 2	op=1(disimil.)	op = 2	op=1(disimil.)
abalone	253,1467	153,8032	226,6090	86,8633	57,2578	33,4802	113,6960
autos	32,8287	22,3855	43,6806	16,5462	20,6059	12,8221	36,6209
balance	725,5165	266,4028	43,4454	32,1254	46,8285	33,6793	33,8669
car	4349,6204	2428,9719	290,0153	119,5605	245,7184	130,8577	290,0931
cleveland	128,7292	89,7000	26,4050	15,8672	16,3368	13,3013	32,4942
contraceptive	4001,0819	2234,6676	374,2146	163,6182	219,4621	101,1941	363,3729
dermatology	527,6622	339,7815	80,0011	47,9508	35,9098	23,2348	77,4344
ecoli	313,3050	167,9569	33,8944	20,7830	23,1164	16,7712	31,9999
glass	136,1953	55,5344	27,7486	15,3582	17,0214	12,0372	27,0519
hayes-roth	39,8466	14,8946	16,7845	9,6313	14,1640	9,0398	7,6868
iris	30,5320	14,8083	10,7548	8,4133	10,2357	8,3174	7,0624
led7digit	419,9818	269,1245	42,5205	30,7033	42,2508	31,1149	45,8651
letter	7618,0388	6127,4751	1717,2462	949,2825	932,8859	570,3734	2205,2858
newthyroid	111,3482	64,5703	25,9411	13,3524	15,1204	12,0497	11,2867
pageblocks	502,6443	274,1000	59,2141	32,7879	31,5034	20,0809	56,2683
penbased	4889,2053	2021,0430	486,4332	189,4744	309,6033	144,6401	375,3666
satimage	4330,7166	2406,4176	423,3307	230,0560	116,3311	51,9650	300,8770
segment	6108,7265	3365,7141	688,7395	345,9016	236,2165	114,4656	1104,9349
shuttle	4210,2764	3310,1163	233,7422	156,1165	154,0057	90,7999	417,1480
tae	65,6479	37,6704	20,3068	12,7785	15,8249	11,3215	14,5789
thyroid	706,6560	422,3182	68,8185	44,7919	30,5725	26,3987	143,2791
vehicle	2166,2344	956,9749	419,2985	142,4844	132,3416	70,3964	320,8571
vowel	2781,2122	1289,9479	816,9433	313,0686	361,3155	157,3704	641,9448
wine	74,6666	38,6512	20,7569	12,1851	13,5038	9,3612	11,0214
yeast	4280,5538	2001,9954	356,7782	118,5345	276,5090	95,9513	286,6762
appendicitis	27,5117	17,1175	13,1002	7,8663	9,3017	7,3934	7,4527
australian	1259,2269	1083,1385	92,8649	52,6900	36,6561	35,8528	104,8167
automobile	75,1850	40,9805	61,4843	26,1370	26,3878	13,6156	55,2745
banana	31848,3305	19448,9245	808,9641	388,0785	813,9277	397,5476	677,4230
bands	600,0215	427,4950	79,8101	40,0096	28,4261	19,0437	54,2720
bupa	254,5181	205,7770	32,1094	20,1733	20,6440	15,8778	23,5723
crx	1113,0276	717,0562	83,5258	48,8986	27,3497	29,7393	86,3239
german	1285,5001	868,2358	127,3730	73,2749	55,2742	35,1322	243,7799
ionosphere	318,9405	144,8173	68,7597	31,5680	25,6925	14,9033	76,4751
mammographic	706,9613	394,1448	40,7818	29,9213	38,0889	30,8903	60,7798
monk-2	307,2800	141,3720	35,9680	24,0899	23,1186	17,1008	38,5620
movement_libras	1286,0267	315,7205	899,7967	312,7669	317,5363	91,6273	726,1824
phoneme	30211,4373	17070,2477	1164,9737	560,7880	1141,5886	557,6459	1003,1236
saheart	307,6342	204,1633	42,9278	21,6873	26,5359	19,0976	36,8505
sonar	266,0255	95,0032	68,2308	30,6531	25,0565	15,3874	117,6829
spectfheart	184,2741	108,3579	60,0533	26,7587	24,5359	14,0502	93,4272
titanic	6433,8817	2631,7854	218,0250	225,0149	230,5598	186,2466	154,1642
wdbc	629,4280	260,0160	87,1613	38,3740	33,8423	18,8610	235,1503
MEDIA	**2928,3624**	**1687,1949**	**245,1054**	**118,3020**	**146,0271**	**77,2334**	**250,0484**

Tabla 11. 2º parte del tiempo de ejecución

5.3. Test estadísticos

En las tablas anteriores se muestran los resultados obtenidos por cada método en cada dataset. También se muestra la media obtenida, pero con estos valores no podemos decir con certeza si un método es mejor que otro. Para ello se van a utilizar los test explicados en la sección 4.4.

La comparativa va a constar de 4 fases:

1. **Pesos y disimilitudes:** En esta fase, vamos a comparar cada variante de pesos con la correspondiente variante de disimilitudes. Primero comparamos con el test de Wilcoxon cada una de las variantes de pesos con la correspondiente de disimilitudes. Luego con la mejor de selección de instancias (entre pesos y disimilitudes), la comparamos con la de selección de instancias y disimilitudes en paralelo. En este punto tenemos 4 métodos ganadores. Por ultimo utilizamos el test de Friedman para determinar el mejor de los 4.
2. **KNN original:** En esta fase utilizamos el test de Friedman para determinar que método es mejor entre las variantes del KNN original. Se comparan KNN original, KNN original + selección de instancias, KNN original + selección de características y KNN original + selección de instancias + selección de características.
3. **Mejor de KEEL:** En esta fase, vamos enfrentar los métodos de KEEL para ver cuál es el mejor. Primero realizamos el test de Friedman para todos los métodos de selección de características (CW, FS-GGA, FS-SSGA). Luego hacemos lo mismo para los de selección de instancias (IS-AllKNN, IS-CHC, IS-ENN, IS-RNG, IS-SSMA). Por ultimo hacemos un test de wilcoxon entre el mejor de selección de características y el mejor de selección de instancias.
4. **Final:** Aquí ya tenemos 3 finalistas, 1 por cada fase. Por lo tanto realizamos un test de Friedman entre los finalistas y ese será el método elegido como mejor.

Pero aparte de estas comparativas, se va a comparar el algoritmo propuesto con el algoritmo KNN original, ya que uno de los objetivos era ver si mejorábamos o no. Para ello se va a utilizar el test de Wilcoxon en cada una de las variantes.

El la figura 7 se muestran todas las comparativas realizadas, con los resultados, de una forma más visual. Cada camino de colores llega hasta una comparativa donde solo sale el mejor. Los colores de estos caminos no significan nada, los caminos se han coloreado para que se pueda seguir mejor la evolución. En la última comparativa, al método ganador se le ha etiquetado con la palabra “GANADOR”.

Como se muestra en la figura 5, los valores contenidos en los recorridos hacen referencia a los rangos obtenidos en los test. El test de Wilcoxon si la comparativa es entre dos métodos y el de Friedman cuando son más.

En la figura 6, se muestran los valores fuera de los recorridos. Estos son los p_{valor}-es comentados en la sección 4.4. Ya se ha dicho que cuando estos valores son menores que 0.05, se dice que hay diferencias estadísticas. En este caso, cuando se cumple esta condición, el p_{valor} aparece en negrita.

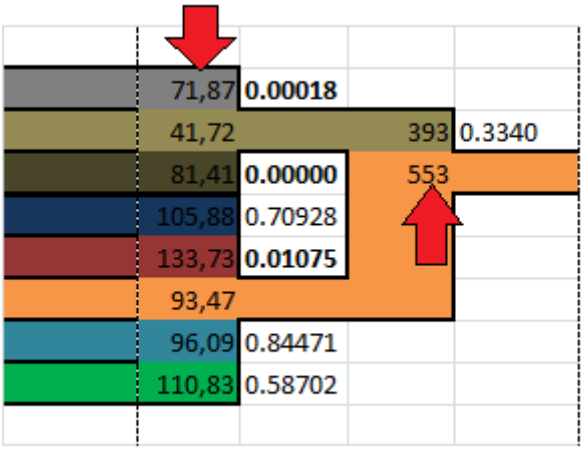

Figura 5. Diferentes de rangos

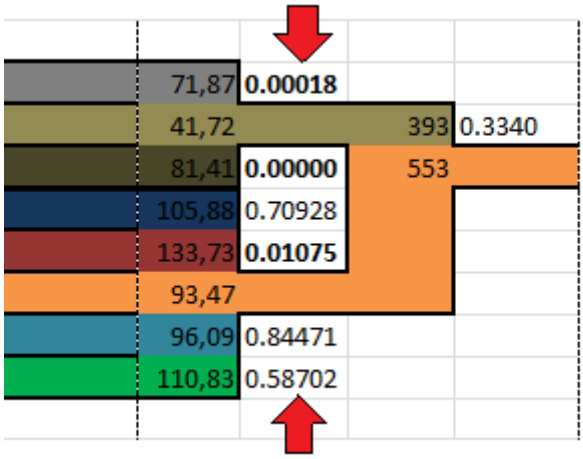

Figura 6. Diferentes p_{valor}-es

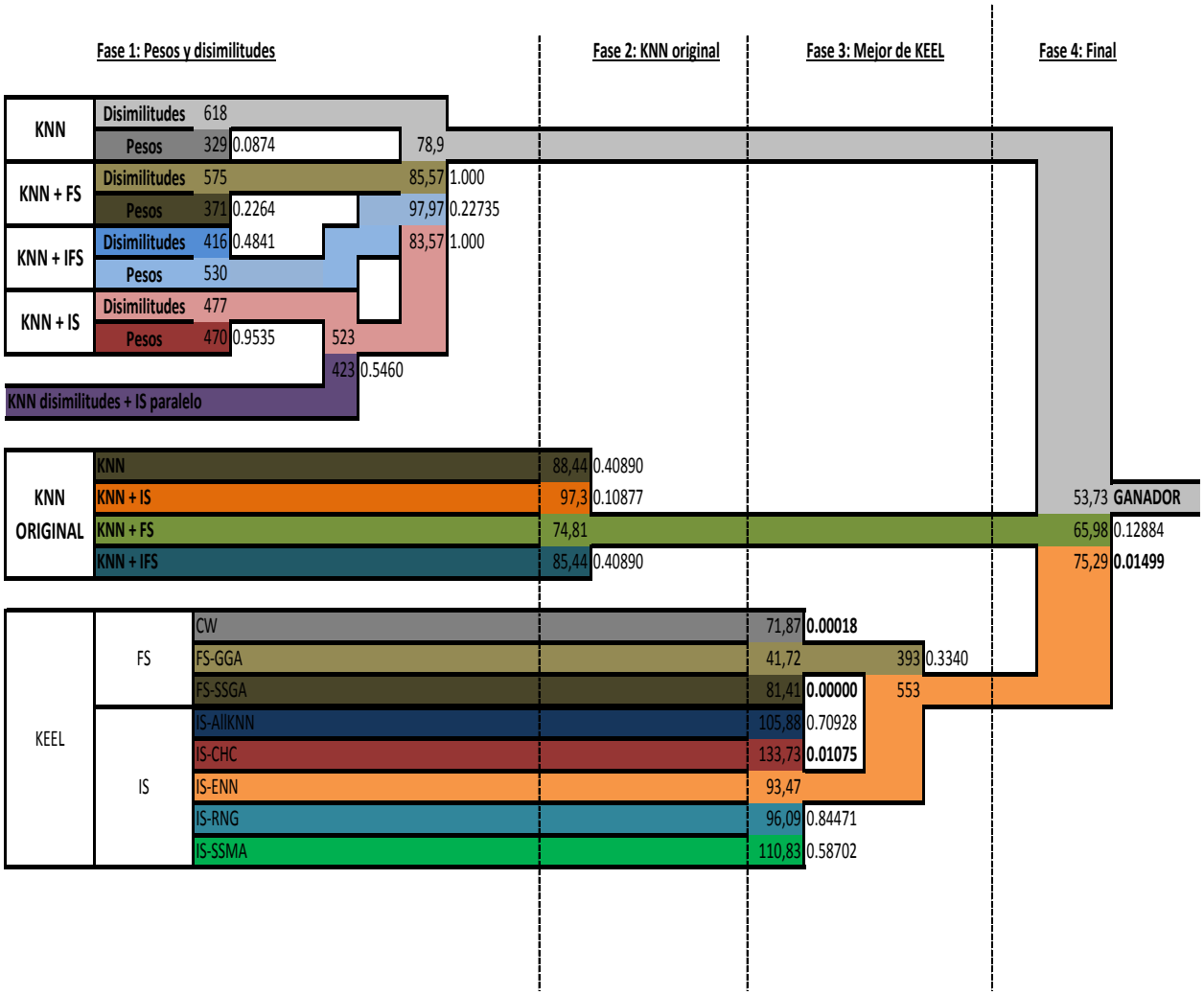

Figura 7. Todas las comparativas

Como se ha dicho antes, hay cuatro fases. A continuación se van a explicar los resultados de cada fase:

1. **Pesos y disimilitudes:** Como se ve en la figura 7, la opción de usar disimilitudes gana a todos los de pesos excepto cuando usamos KNN con selección de instancias y características. Pero ninguno de los vencedores es estadísticamente mejor que otros. Comparando el mejor de selección de instancias (que es el de disimilitudes) frente al paralelo, gana el de serie, pero como antes, no estadísticamente. Por ultimo cuando hacemos Friedman entre los ganadores, gana el KNN con disimilitudes sin ningún tipo de selección. Como antes sin diferencias estadísticas. Este pasa a la fase final.
2. **KNN original:** En este caso, hacemos un Friedman entre las 4 variantes del algoritmo KNN original. Como se ve, el que menos rangos tiene es KNN con

selección de características, por lo tanto es el ganador, pero sin diferencias estadísticas.

3. **Mejor de KEEL:** En este caso, primero hacemos Friedman entre todos los de selección de características. En la imagen se ve como el mejor es FS-GGA, y además es estadísticamente mejor que los otros dos métodos. Luego hacemos Friedman con los de selección de instancias, y el que menos rangos obtiene es IS-ENN. En este caso, solo podemos decir que IS-ENN es mejor estadísticamente que IS-CHC. Por último, entre los ganadores (FS-GGA y IS-ENN), hacemos un Wilcoxon. En la imagen se ve que el IS-ENN obtiene más rangos, por lo tanto es mejor, pero no estadísticamente.
4. **Final:** Para esta fase nos llegan los mejores de las fases anteriores. Tenemos los métodos: KNN con disimilitudes, KNN original con selección de características y IS-ENN. Haciendo un test de Friedman entre todos vemos que el que menos rangos tiene es el algoritmo propuesto (KNN con disimilitudes), por lo tanto es el ganador. Decir que este es estadísticamente mejor que el algoritmo IS-ENN.

También se ha dicho que se quiere comparar el algoritmo propuesto en todas sus variantes, contra el algoritmo KNN original.

A. **KNN disimilitudes vs KNN original:** El algoritmo KNN original obtiene 250 rangos y el algoritmo KNN disimilitudes obtiene 696. Por lo tanto el de disimilitudes es mejor. El p_{valor} es 0.0060 por consiguiente, es mejor estadísticamente.
B. **KNN disimilitudes + selección de características vs KNN original + selección de características:** El KNN original obtiene 433 rangos y el KNN con disimilitudes 513. El p_{valor} es 0.5959 por lo tanto el de disimilitudes es mejor pero no estadísticamente.
C. **KNN disimilitudes + selección de instancias vs KNN original + selección de instancias:** El KNN original obtiene 255 rangos y el de disimilitudes 690. El p_{valor} es 0.0086, por lo tanto el de disimilitudes es mejor estadísticamente.
D. **KNN disimilitudes + selección de instancias y características vs KNN original + selección de instancias y características:** El KNN original obtiene 449 rangos y el de disimilitudes 497. El p_{valor} es 0.8055, por lo tanto el de disimilitudes es mejor, pero no estadísticamente.

Se ve que el método propuesto es mejor que el original en todas las variantes, pero estadísticamente únicamente en dos de las cuatro.

6. Conclusiones y líneas futuras

En este proyecto hemos propuesto un sistema de clasificación basado en el algoritmo de los k vecinos más cercanos. Este sistema aprende ciertos parámetros que son utilizados para calcular la distancia entre los ejemplos. Además de la distancia, también se han implementado métodos de selección de instancias y/o características tanto para el algoritmo KNN original como para el propuesto.

Para comparar los resultados obtenidos hemos llevado a cabo una serie de test estadísticos y los hemos analizado en profundidad. En resumen las conclusiones obtenidas son las siguientes:

- El método que hemos propuesto obtiene mejores resultados que el método original en todas las variantes propuestas (selección de instancias y/o características) en los datasets utilizados.
- El método propuesto es estadísticamente mejor que el original, y la combinación de nuestro método con selección de instancias, también es estadísticamente mejor que el original con selección de instancias.
- El método propuesto, es estadísticamente mejor que el mejor de los métodos implementados en KEEL.
- El método propuesto, obtiene mejores resultados que el algoritmo KNN utilizando pesos por cada atributo.

En lo que a líneas futuras se refiere, este proyecto tiene varias ya que los resultados obtenidos son más que favorables:

- La primera y más clara de toda ellas es la de realizar el estudio con diferentes valores para el parámetro k $(k = 3, k = 5, \dots)$.
- La segunda es la de realizar el estudio con automorfismos con construcciones diferentes.
- Utilizar diferentes tipos de agregaciones.
- Utilizar diferentes funciones objetivo.
- Estudiar el comportamiento del nuevo método en el problema de las clases no balanceadas.

7. Bibliografía

[1] J. Derrac, S. García and F. Herrera, “IFS-CoCo: Instance and Feature selection based on cooperative coevolution with nearest neighbor rule”, Pattern Recognition, 2010, pag. 2082-2105

[2] M. Galar, A. Fernandez, E. Barrenechea, F. Herrera, “Empowering difficult classes with a similarity-based aggregation in multi-class classification problems”, Information Sciences, vol. 264, 2014, pag. 135-157.

[3] M. Steinbach and P. Tan, “kNN: k-Nearest Neighbours”, The Top Ten Algorithms in Data Mining (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series), 2009.

[4] H. Bustince, E. Barrenechea, M. Pagola y V. Mohedano, “Relación entre las funciones de disimilaridad restringida y las funciones de equivalencia restringida”. In XIII Congreso Español Sobre Tecnologías y Lógica Fuzzy, 41-46, Ciudad Real, España, 2006.

[5] S. García and F. Herrera, “Evolutionary Undersampling for Classification with Imbalanced Datasets: Proposals and Taxonomy”, Evolutionary Computation, vol. 17, no. 3, 2009, pag. 275-306.

[6] M. Galar, A. Fernández, E. Barrenechea, H. Bustince and F. Herrera, “A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches”, IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, vol. 42, no. 4, 2012.