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

Análisis de la eficacia del tuning local en un modelo de clasificación basado en reglas difusas de asociación (FARC-HD)



Grado en Ingeniería Informática

Trabajo Fin de Grado

Alumno: Oier Etxeberria Urrestarazu

Director: José Antonio Sanz

Pamplona, fecha de defensa (I don´t know yet men)



Agradecimientos

Patxi Mario Rodolfo el tercero.

Oier.

Resumen

Hoy en día el campo del *Machine Learning* o Aprendizaje Automático es un campo de investigación que está en auge. Dentro de este campo hay una gran variedad algoritmos y métodos capaces de extraer información. Dentro del subcampo de los algoritmos de clasificación nos encontramos con el FARC-HD.

FARC-HD es un algoritmo muy preciso y para ello en su etapa final aplica un algoritmo evolutivo para realizar un ajuste en las funciones de pertenencia y una selección de las mejores reglas. El ajuste de las funciones de pertenencia se hace de forma global para mantener la interpretabilidad del sistema y el objetivo del trabajo es analizar la eficacia de aplicar un ajuste local a expensas de perder la interpretabilidad del sistema.

Palabras clave

* Sistemas de clasificación
* Reglas de asociación difusas
* FARC-HD
* Tuning lateral
* Tuning local

Summary

Key words

Laburpena

Hitz Klabeak

Índice de Contenido

[1.- Introducción 9](#_Toc52548968)

[2.- Preliminares 10](#_Toc52548969)

[2.1.- Machine Learning 10](#_Toc52548970)

[2.2.- Problemas de Clasificación 10](#_Toc52548971)

[2.3.- Lógica Difusa 10](#_Toc52548972)

[2.3.1- Operadores 12](#_Toc52548973)

[2.3.2.- Implicación 13](#_Toc52548974)

[2.3.3.- Reglas Difusas 14](#_Toc52548975)

[2.4.- Tuning Local 15](#_Toc52548976)

[2.5.- Sistemas de Clasificación Basados en Reglas Difusas 15](#_Toc52548977)

[2.6.- Algoritmos Genéticos 15](#_Toc52548978)

[2.6.1.- Codificación 16](#_Toc52548979)

[2.6.2.- Cruce 16](#_Toc52548980)

[2.6.3.- Mutación 16](#_Toc52548981)

[2.6.4.- Selección 16](#_Toc52548982)

[2.7.- FARC-HD 16](#_Toc52548983)

[3.- FARC-HD-LOCAL 17](#_Toc52548984)

[4.- Marco Experimental 18](#_Toc52548985)

[4.1.- Datasets 18](#_Toc52548986)

[4.2.- Configuración de parámetros 19](#_Toc52548987)

[4.3.- Medidas de Rendimiento 19](#_Toc52548988)

[5.- Estudio Experimental 20](#_Toc52548989)

[Bibliografía 21](#_Toc52548990)

Índice de Ilustraciones

[Ilustración 1. Ejemplo lógica Clásica 11](#_Toc52545587)

[Ilustración 2. Ejemplo Lógica difusa. 11](#_Toc52545588)

[Ilustración 3. Variable difusa con sus etiquetas lingüísticas 12](#_Toc52545589)

[Ilustración 4. Esquema algoritmo evolutivo 15](#_Toc52545590)

Índice de Tablas

[Tabla 1. Descripción de las características de los dataset utilizados 12](#_Toc52289615)

[Tabla 2. Configuración de Parámetros 13](#_Toc52289616)

# 1.- Introducción

La inteligencia artificial es la ciencia que tiene como objetivo hacer que lo ordenadores puedan realizar tareas de humanos tal y como los haría un humano. Dentro de la ciencia de la inteligencia Artificial se encuentra el subcampo del *Machine Learning* o Aprendizaje Automático. Este subcampo se encarga de entrenar a la máquina para que aprenda por sí solo.

El aprendizaje automático tiene como objetivo aprender de una base de datos llena de ejemplos y posteriormente ser capaz de aplicar el aprendizaje en ejemplos desconocidos para la máquina.

Para realizar el aprendizaje podemos empezar de diferentes puntos de partida. Si en los ejemplos proporcionados para aprender sabemos la salida que debemos obtener, estamos ante lo que se llama aprendizaje supervisado. Por otro lado, si no lo sabemos, estamos ante aprendizaje no supervisado. Aparte de estos dos métodos existen más métodos, como el aprendizaje semisupervisado, por refuerzo, multi tarea, transducción etc. En nuestro caso utilizaremos el aprendizaje supervisado.

Este TFG está basado en el algoritmo de clasificación basado en reglas difusas de asociación FARC-HD (*Fuzzy Rule-Based Classification model for High-Dimensional problems*) [1].

El algoritmo tiene tres fases. En la primera fase, se extraen reglas difusas de asociación. En la segunda fase, se minimizan las reglas preseleccionándolas conforme su calidad. Por último, en la tercera fase, se aplica un algoritmo genético para seleccionar las mejores reglas y ajustar las funciones de pertenencia.

Este trabajo se centra en cambiar el ajuste a las funciones de pertenencia. En el FARC-HD se ajusta de forma global con el fin de mantener la interpretabilidad del sistema. Nosotros estudiaremos el efecto de ajustarnos de forma local para cada regla.

Al ajustarnos de forma local, perdemos la interpretabilidad del sistema, pero logramos ajustarnos mejor a cada regla, por lo tanto, deberíamos mejorar el rendimiento. Por otro lado, para ajustarnos de forma local necesitaremos una base de datos para cada regla, lo que supone un aumento en el tiempo de computación.

# 2.- Preliminares

En este apartado del proyecto explicaremos en detalle el marco teórico necesario para entender el funcionamiento del algoritmo empleado en el TFG. Para ello, explicaremos los conceptos básicos paso a paso.

## 2.1.- Machine Learning

El *Machine Learning* se define como el aprendizaje de las máquinas y es un subcampo dentro de la Inteligencia Artificial. Todos los sistemas del *Machine Learning* generan un modelo y este es el que se encarga de procesar toda la información y tomar decisiones. Este modelo trabaja obteniendo una entrada de datos, procesándolos y obteniendo una salida sobre estos.

## 2.2.- Problemas de Clasificación

Los problemas de clasificación son un subcampo del *Mahine Learning*. En este subcampo recibimos ejemplos y tratamos de clasificarlos en la clase a la que pertenecen. Para ello, partimos de un conjunto de ejemplos de los cuales sabemos a qué clase pertenecen. El objetivo es aprender los patrones generales que se repiten para pertenecer a cada clase y así clasificar correctamente ejemplos desconocidos.

## 2.3.- Lógica Difusa

La Lógica difusa la introdujo Lofti Zadeh [2] en 1965. En la lógica clásica, un elemento solo puede obtener dos valores: verdadero o falso. Es decir, un elemento puede pertenecer o no a un conjunto, pero el grado de pertenencia será absoluto. En la lógica difusa, por lo contrario, cada elemento tiene un grado de pertenencia con un valor real en el rango [0,1] para cada conjunto.

Vamos a explicar el ejemplo clásico que se suele utilizar para entender la lógica difusa. Vamos a determinar si un paciente tiene fiebre o no. Desde la perspectiva de la lógica clásica tendremos fiebre si la temperatura del paciente es mayor que 39º y sino no. Es decir, tendremos fiebre si la temperatura del paciente pertenece al conjunto de “*Fiebre Fuerte*”. Esta situación está representada en la Ilustración 1.

Diagrama

Descripción generada automáticamente

Ilustración 1. Ejemplo lógica Clásica

Por lo contrario, si representamos este caso con la lógica difusa, obtendríamos un valor real entre [0,1] para el conjunto de “*Fiebre Fuerte*”. Este caso esta representado en la Ilustración 2.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 2. Ejemplo Lógica difusa.

La lógica difusa nos permite representar mejor la realidad, ya que podemos representar conceptos como mucho frio, poco alto, muy gordo…

A la hora de representar conjuntos difusos, tendremos las *variables lingüísticas* como altura, peso, temperatura… y también tendremos *etiquetas lingüísticas* como bajo, medio, alto… Podemos representarlo como en la Ilustración 3.

Diagrama

Descripción generada automáticamente

Ilustración 3. Variable difusa con sus etiquetas lingüísticas

Como estamos tratando con conjuntos, con la lógica difusa también se necesitan operadores de unión, intersección y complementario. Para estas operaciones, existen diferentes métodos llamados *t-normas, t-conormas,* y *negaciones.*

### 2.3.1- Operadores

1. **T-normas**

**Definición:** *Una t-norma es una función T: que satisface:*

* 1. Cota:
  2. Monotonía:
  3. Conmutativa:
  4. Asociativa:

Estos son algunos de los operadores más utilizados para representar la intersección:

* *Mínimo:*
* *Producto:*
* *Lukasiewicz:*

1. **T-conormas**

**Definición:** *Una t-conorma es una función S: que satisface:*

* 1. *Cota:*
  2. *Monotonía:*
  3. *Conmutativa:*
  4. *Asociativa:*

Estos son algunos de los operadores más utilizados para representar la unión:

* *Máximo:*
* *Suma algebraica:*
* *Lukasiewicz:*

1. **Negaciones**

**Definición:**  es una negación difusa si y solo si:

Además, se dice que una negación es estricta si:

* 1. es continua

E involutiva si:

Cuando una negación es estricta e involutiva se dice que es una negación fuerte. Así, dado un conjunto A, su complementario es siendo *c* una negación fuerte.

### 2.3.2.- Implicación

En dos universos U y V una relación difusa es un conjunto dado por el producto cartesiano que se puede definir con una función de pertenencia.

(1)

Una implicación (1) se define como un caso especial de relación difusa, la cual es posible definir mediante una función de pertenencia . Y a partir de esa función podemos construir una matriz R que represente la Regla.

Los operadores de implicación son funciones que satisfacen las propiedades:

Estos son algunos de los operadores de implicación más utilizados:

* *Kleene-Dienes:*
* *Lukasiewicz:*
* *Mamdani:*
* *Zadeh:*
* *Larsen:*

### 2.3.3.- Reglas Difusas

Para construir una relación R que represente una regla, se pueden utilizar operadores de implicación o t-normas. Las t-normas en al practica suelen mejorar el rendimiento, pero no aprovechan parte de la teoría de la lógica clásica que con los operadores de implicación sí.

Para la regla (1) una opción para construir la regla puede ser así:

(2)

Existen distintos tipos de reglas difusas:

1. **Reglas difusas con una clase en el consecuente**

Si es y … y es entonces Y es

Donde , …, son las variables de entrada, , …, son las etiquetas lingüísticas de las variables e Y es la variable de salida que indica a cuál de las clases del conjunto de clases pertenece el ejemplo.

1. **Reglas difusas con una clase y un grado de certeza asociado**

Si es y … y es entonces Y es con grado

Donde , …, son las variables de entrada, , …, son las etiquetas lingüísticas de las variables e Y es la variable de salida que indica a cuál de las clases del conjunto de clases pertenece el ejemplo y es el grado de certeza de la regla .

1. **Reglas difusas con grados de certeza asociados a cada clase del consecuente**

Si es y … y es entonces

Donde , …, son las variables de entrada, , …, son las etiquetas lingüísticas de las variables y el consecuente es un vector de pesos en el que cada denota el grado de certeza de la regla para predecir la clase .

## 2.4.- Tuning Local

## 2.5.- Sistemas de Clasificación Basados en Reglas Difusas

## 2.6.- Algoritmos Genéticos

Dentro de los modelos de computación bio-inspirados se encuentran, los algoritmos evolutivos, las redes neuronales, los algoritmos basados en enjambres etc. Nosotros vamos a fijarnos en los algoritmos evolutivos.

Los algoritmos evolutivos tienen como objetivo resolver problemas de optimización o búsqueda y lo que los distingue es que el elemento clave de su diseño es algún mecanismo de evolución. Estos algoritmos intentan imitar el funcionamiento de los cromosomas en la vida real.

Los principales componentes de los algoritmos evolutivos son los siguientes:

* Población de individuos. Cada individuo representa una posible solución y se le llama cromosoma.
* Procedimiento de selección. Tiene algún mecanismo de selección basado en la aptitud de los individuos para resolver el problema.
* Procedimiento de transformación. En la evolución del programa se generan nuevos individuos a partir de los anteriores.

Existen distintos tipos de algoritmos evolutivos, pero en todos ellos el funcionamiento general es el siguiente (Ilustración 4):

1. Partimos de una población inicial de cromosomas. El número de cromosomas que conforman la población se mantiene fija.
2. Se evalúa la aptitud de cada cromosoma de la población.
3. Se seleccionan los cromosomas que se utilizaran para crear una nueva generación.
4. Se combinan los cromosomas seleccionados y se generan nuevos cromosomas. Los cromosomas pueden sufrir mutación.
5. Se genera la nueva población. Esta nueva población la completan los mejores cromosomas resultantes del conjunto de cromosomas de la generación anterior y la nueva.
6. Se repite el proceso de evaluación, selección y combinación hasta optimizar el resultado.

Diagrama

Descripción generada automáticamente

Ilustración 4. Esquema algoritmo evolutivo

Los algoritmos genéticos son un tipo de algoritmo evolutivo. Las principales características de un algoritmo genético son 3:

1. Los cromosomas se representan mediante codificación binaria.
2. Como método de selección suelen emplear el método de la ruleta con elitismo.
3. Como método de combinación emplean el cruce y la mutación.

A continuación, vamos a explicar más detalladamente en que se basan o como se realizan los apartados de codificación, cruce, mutación y selección.

### 2.6.1.- Codificación

La codificación es la forma en la que se representa la información en nuestro cromosoma.

### 2.6.2.- Cruce

### 2.6.3.- Mutación

### 2.6.4.- Selección

## 2.7.- FARC-HD

# 3.- FARC-HD-LOCAL

# 4.- Marco Experimental

En este apartado se presenta el contexto elegido para la realización de los experimentos. En primer lugar, describimos los conjuntos de datos (*datasets*) utilizados. Luego, mostramos los parámetros con los que se han ejecutado las pruebas. Por último, definimos las medidas de rendimiento empleadas para evaluar los resultados.

## 4.1.- Datasets

Hemos seleccionado nueve conjuntos de datos (*datasets*) que también se han utilizado para evaluar el rendimiento de FARC-HD original [1]. Estos conjuntos de datos contienen datos del mundo real obtenidos del repositorio de KEEL [5], disponibles públicamente desde el sitio web del proyecto[[1]](#footnote-1). En la tabla 1 resumimos las características principales de cada conjunto: número de ejemplos (#Ej.), número de atributos (#Atr.), y número de clases (#Clas.).

Tabla 1. Descripción de las características de los dataset utilizados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id.** | **Dataset** | **#Ej.** | **#Atr.** | **#Clas.** |
| clv | cleveland | 297 | 13 | 5 |
| eco | ecoli | 336 | 7 | 8 |
| iri | iris | 150 | 4 | 3 |
| mag | magic | 19020 | 10 | 2 |
| pag | pageblocks | 5472 | 10 | 5 |
| pen | penbased | 10992 | 16 | 10 |
| pm | pima | 768 | 8 | 2 |
| wine | wine | 178 | 13 | 3 |
| yst | yeast | 1484 | 8 | 10 |

Para evaluar el rendimiento de los algoritmos, se ha utilizado un modelo de validación cruzada de 5 particiones.

## 4.2.- Configuración de parámetros

En este apartado se muestran los parámetros con los que se ha realizado la experimentación. Dichos parámetros son los mismos que se utilizan en el FARC-HD [1] para realizar las pruebas. En el FARC-HD los parámetros que se utilizan son los recomendados por los autores para cada sección del algoritmo global [6]. Los parámetros que hemos utilizado se muestran en la tabla 2.

Tabla 2. Configuración de Parámetros

|  |
| --- |
| **Parámetros** |
| Número de etiquetas lingüisticas por variable: 5 |
| Soporte mínimo: 0.05 |
| Confianza mínima: 0.8 |
| Profundidad máxima Árbol: 3 |
| Parámetro K: 2 |
| Número máximo de evaluaciones: 20000 |
| Número de individuos: 50 |
| Parámetro α: 0.02 |
| Bits por gen: 30 |
| Inferencia: 1 (Combinación aditiva) |

## 4.3.- Medidas de Rendimiento

Para evaluar el rendimiento de las pruebas, se utiliza el grado de precisión o *accuracy rate*. Este método calcula el porcentaje de ejemplos clasificados correctamente. Por otro lado, utilizamos el tiempo para comparar el coste temporal de un ajuste local contra un ajuste global en el apartado genético. Por último, se estudia la convergencia del ajuste local mediante el grado de precisión mostrado en gráficos.

# 5.- Estudio Experimental

- Comparaciones

1. FARC-HD Global + suma pertenencia (FRM\_AC) VS lo mismo en local
2. FARC-HD Global + suma pertenencia (FRM\_WR) VS lo mismo en local
3. FARC-HD Global + FRM\_AC VS FARC-HD local + FRM\_WR

FARC-HD local + FRM\_WR -> semi-interpretable

- Gráficos convergencia

- Subir número máximo de evaluaciones?

# Bibliografía

1. J. Alcalá-Fdez, R. Alcalá y F. Herrera, «A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tunning.,» IEEE Transactions on Fuzzy Systems, vol. 19, nº 5, pp. 857-872, 2011.
2. L. A. Zadeh, «Fuzzy Sets,» Information and Control, vol. 8, nº 3, pp. 338-353, 1965
3. M. Mitchell, Introduction to Genetic Algorithms, 1998.
4. D. E. Goldberg, Genetic algorithms in search, optimization and machine learning, 1989.
5. J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez y F. Herrera, «KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework,» *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no, 2-3, pp. 255-287, 2011
6. J. Alcala-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernandez, and ´ F. Herrera, “KEEL: A software tool to assess evolutionary algorithms to data mining problems,” Soft Computing, vol. 13, no. 3, pp. 307–318, 2009

Cosas a revisar

* Traducir resumenes
* Poner bien números bibliografía
* Tablas con nombre abreviado?
* Referencia a tablas, ilustraciones, bibliografía..

1. <https://sci2s.ugr.es/keel/datasets.php> [↑](#footnote-ref-1)