Análisis del comportamiento de un algoritmo de aprendizaje de SBRD TSK-1 mediante búsqueda local variando la inicialización

Javier Cózar, Luis delaOssa, and José A. Gámez

Computing Systems Department - I^3A
University of Castilla-La Mancha - Spain
{javier.cozar,luis.delaossa,jose.gamez}@uclm.es

Resumen En un estudio anterior se diseñó un algoritmo basado en búsqueda local para derivar SBRDs TSK-1 a partir de un conjunto de datos. En este trabajo extenderemos este trabajo, analizando el comportamiento de dicho algoritmo variando el sistema inicial, factor que es determinante en cualquier algoritmo de búsqueda local. Finalmente, se realizará un análisis estadístico comparando cada una de las propuestas.

Keywords: Modelado Difuso, Búsqueda Local, TSK-1, Inicialización

1. Introducción

Los Sistemas Basados en Reglas Difusas (SBRDs) son modelos utilizados principalmente por su alto nivel de interpretabilidad. Este hecho permite que un experto humano sea capaz de extraer conocimiento útil a partir de estos modelos, o bien incorporar información a los mismos para mejorar su rendimiento. Estos sistemas están formados por reglas del tipo "Si-entonces", y han sido utilizados tanto en problemas de clasificación como en problemas de regresión [1,2,3].

Hay algoritmos capaces de aprender estos modelos a partir de un conjunto de datos, pero con un orden de complejidad exponencial en cuanto al número de variables. Algunos trabajos [3,4] se centran en reducir este coste computacional a costa de una reducción en la calidad de las soluciones. En [5] se utiliza esta misma estrategia combinada con un algoritmo de búsqueda local que parte del individuo vacío de reglas para la generación de SBRDs tipo TSK-1.

La elección del individuo inicial es un aspecto crucial para los algoritmos de búsqueda local, influyendo enormemente en la solución final. En este trabajo partiremos del algoritmo expuesto en [5], y realizaremos un estudio sobre diferentes inicializaciones con la intención de mejorar los resultados conseguidos.

La estructura de este trabajo se divide en 4 secciones, además de esta introducción. En la sección 2 describiremos los SBRDs tipo TSK-1. Posteriormente, en la sección 3 revisaremos el algoritmo expuesto en [5]. En la sección 4 se estudiará el comportamiento del algoritmo de búsqueda local variando la inicialización. Finalmente, en la sección 5 se extraerán las conclusiones.

2. Sistemas Basados en Reglas Difusas TSK-1

Los SBRDs estan formados por una base de conocimiento, que contiene la definición de los dominios de las variables y los conjuntos difusos asociados a las mismas, y un sistema de reglas difusas definidas sobre la base de conocimiento. El antecedente de una regla consta de uno o varios predicados, unidos mediante el operador de conjunción, de la forma X es F, donde X es una variable del problema y F es un conjunto difuso definido sobre el dominio de X. Una forma de incrementar la interpretabilidad de estos sistemas consiste en usar predicados del tipo X es A, donde A es una etiqueta lingüística y a su vez ésta se vincula con el conjunto difuso F. Estos sistemas se denominan sistemas basados en reglas difusas lingüísticas (SBRDLs) [6]. En los sistemas tipo TSK [7] el consecuente es una función polinómica de las variables de entrada del problema $P_s(X_1, \ldots, X_n)$. El uso de este tipo de consecuentes (al contrario que en sistemas tipo Mamdani [8], en los que el consecuente de las reglas son conjuntos difusos) supone una gran mejora en la precisión de estos sistemas, a costa de una pérdida en el nivel de interpretabilidad. El orden de un SBRD tipo TSK se refiere al grado del polinomio, por lo que los sistemas TSK-1 se refieren a sistemas en los que los consecuentes de las reglas son polinomios de grado 1.

Dado un conjunto de reglas $\mathcal{RB}$, la predicción global del sistema cuando la instancia $e_l = (x_1^l, \ldots, x_n^l, y^l)$ es procesada, es una media ponderada de las salidas inividuales generadas por cada regla $R_s \in \mathcal{RB}$ (ver ecuación 1).

$$\hat{y}^l = \frac{\sum_{R_s \in \mathcal{RB}} h_s^l P_s(x_1^l, \ldots, x_n^l)}{\sum_{R_s \in \mathcal{RB}} h_s^l} = \frac{\sum_{R_s \in \mathcal{RB}} h_s^l v_s^l}{\sum_{R_s \in \mathcal{RB}} h_s^l} \tag{1}$$

donde $h_s^l = T(A_1^s(x_1^l), \ldots, A_n^s(x_n^l))$ es el grado de cobertura de la instancia e_l con la regla R_s, T es una T-Norma[1], y $v_s^l = P_s(x_1^l, \ldots, x_n^l) = (a_{s_1} x_1 + \cdots + a_{s_n} x_n + b_s)$ es el valor del polinomio del consecuente de la regla R_s dada la instancia e_l.

3. Aprendizaje de SBRDs

En este trabajo nos centraremos en los algoritmos que derivan el sistema de reglas de un SBRDs fijada la base de conocimiento. En este caso, el método de aprendizaje parte de dos elementos:

- Un conjunto de datos $\mathcal{E} = \{e_1, \ldots, e_l, \ldots, e_N\}$, donde $e_l = (x_1^l, \ldots, x_n^l, y^l)$, x_i^l es el valor de la variable X_i en e_l e y^l es la salida de la misma[2].
- La base de conocimiento, que contiene la definición de las variables lingüísticas (sus dominios, las particiones difusas y, en el caso de SBRDLs, las etiquetas lingüísticas, para cada variable de entrada).

[1] En este trabajo usamos *min* como T-Norma.

[2] Por simplicidad consideramos una única variable de salida, pero este número podría ser mayor que uno.

En estos algoritmos, el proceso de derivación del conjunto de reglas se suele dividir en dos pasos. En primer lugar, se deriva un conjunto de reglas candidatas, que son reglas que potencialmente pueden pertenecer al sistema que modela el comportamiento subyacente en el conjunto de datos. Posteriormente se selecciona un subconjunto de reglas y se fija el consecuente de las mismas.

En [5], el proceso de generación de reglas candidatas se realiza a su vez en dos etapas. En primer lugar un algoritmo de descubrimiento de reglas asociativas deriva un conjunto de reglas candidatas R_s que superan un umbral de grado de cobertura, siendo éste la suma de los grados de cobertura de cada instancia del problema con R_s. Con el fin de reducir todavía más el tamaño de este conjunto de reglas, a continuación se aplica un proceso de selección, el cuál iterativamente escoge aquella regla que maximiza una métrica de calidad denominada mWRMSE(R_s), que básicamente premia por un lado el buen comportamiento a nivel individual con respecto a los datos, y por otro el grado de cobertura, pesando más aquellas instancias que todavía no han sido cubiertas por reglas previamente seleccionadas.

Posteriormente, se aplica un algoritmo basado en búsqueda local para seleccionar un subconjunto de las reglas candidatas y fijar sus consecuentes. La ventaja principal de utilizar un algoritmo de búsqueda local radica en la eficiencia computacional, ya que aprovecha el hecho de que un cambio en el consecuente de una regla R_s no afecta al error en la predicción del conjunto total de reglas, sino solo al error de aquellas reglas R_t tales que $\mathcal{E}^s \cap \mathcal{E}^t \neq \emptyset$, donde $\mathcal{E}^s$ y $\mathcal{E}^t$ es el conjunto de instancias cubiertas por las reglas R_s y R_t respectivamente. En este trabajo, la búsqueda del mejor conjunto de reglas parte del sistema vacío (cuando ninguna regla es disparada se utiliza una predicción por defecto consistente en el valor medio de la variable de salida). Por último, el proceso de búsqueda es guiado por la métrica de error RECM (Raíz del Error Cuadrático Medio).

4. Estudio experimental

En [5] se utilizaron cuatro métodos para seleccionar el subconjunto de reglas candidatas y fijar el consecuente de las mismas. Finalmente se concluyó que el método basado en búsqueda local, partiendo del sistema vacío, consiguió los mejores resultados de entre los 4 métodos propuestos (tanto en error como en simplicidad del sistema generado). Sin embargo, dado el reducido número de reglas de los sistemas construidos nos cuestionamos si una inicialización diferente podría significar un aumento en la capacidad de predicción de los sistemas con un leve incremento de la complejidad del modelo.

En este trabajo se estudiarán diferentes inicializaciones para el algoritmo de búsqueda local propuesto en [5] utilizando la misma configuración de parámetros propuesta por los autores. Para la experimentación utilizaremos un total de 18 conjuntos de datos obtenidos del repositorio KEEL[3]. De los 32 problemas de regresión disponibles se han seleccionado los que cumplen las siguientes características, orientadas a evitar tiempos de ejecución demasiado elevados: siendo

[3] http://http://www.keel.es/

n el número de variables de entrada y L el número de instancias, $6 \leq n \leq 40$ y $L \leq 15000$. El nombre de los problemas son: *machineCPU*, *dee*, *delta_elv*, *autoMPG8*, *ANACALT*, *concrete*, *abalone*, *stock*, *wizmir*, *wankara*, *forestFires*, *treasury*, *mortgage*, *baseball*, *compactiv*, *pole*, *puma32h*, *ailerons*.

Esta sección se divide en tres partes. En primer lugar detallaremos cada uno de los sistemas propuestos como puntos de partida. A continuación comprobaremos la calidad de cada uno de ellos, y finalmente se analizará el efecto de utilizar cada inicialización con la búsqueda local.

4.1. Individuos iniciales

En este estudio hemos decidido probar un total de 9 configuraciones para el sistema de reglas inicial, de tal manera que existan el doble de conjuntos de datos, cantidad que consideramos suficiente para permitir extraer conclusiones contrastadas. De estas 9 configuraciones, una (**Vacío**) será la utilizada en [5] (el sistema vacío de reglas), otras dos incluirán el conjunto total de reglas candidatas (**Completo**) y el resto contendrá el 25 % de las reglas candidatas. La decisión de fijar un único porcentaje de reglas ha sido tomada para evitar una excesiva cantidad de propuestas. Además, observando el número de reglas candidatas generado en los distintos problemas, el 25 % nos ha parecido un valor adecuado ya que significa un reducido número de reglas pero suficiente como para dotar de información a las soluciones iniciales. Los 6 sistemas con el 25 % de reglas candidatas se pueden dividir en tres grupos: selección aleatoria de las reglas (**Aleatorio**), selección informada basada en cobertura de ejemplos (**Cobertura**), es decir, se seleccionan las que mayor grado de cobertura tienen, y selección informada basada en el solapamiento entre reglas (**Solape**). En este caso, se escoge en cada iteración la regla que más generaliza (con mayor grado de cobertura), pero en este caso el grado de cobertura de cada instancia con la regla pesa positivamente si no es cubierta por otra regla previamente seleccionada y negativamente en caso contrario, multiplicando además el grado de cobertura por el número de reglas que cubren la instancia, de tal manera que cuanto más solapamiento exista mayor será la penalización.

Finalmente, para fijar los consecuentes de las reglas seleccionadas (salvo *Vacío* por ausencia de reglas), hemos utilizado la técnica de mínimos cuadrados teniendo en cuenta la cooperación entre reglas (mínimos cuadrados global, utilizado en [9]) y sin cooperación (mínimos cuadrados local, utilizado en [10]).

4.2. Análisis de los individuos iniciales

Con el fin de comprobar la calidad individual de cada una de las alternativas, en esta subsección estudiaremos cada una de ellas atendiendo tanto al error de training como de test. En la figura 1 podemos ver de forma gráfica el rendimiento de cada una de las estrategias. Como se puede observar, utilizar la técnica de mínimos cuadrados global conlleva en la mayoría de los casos un problema de sobreajuste (grandes diferencias entre el error de training y de test). En la figura 2a se muestra un diagrama de cajas y bigotes con los rankings medios en cuanto

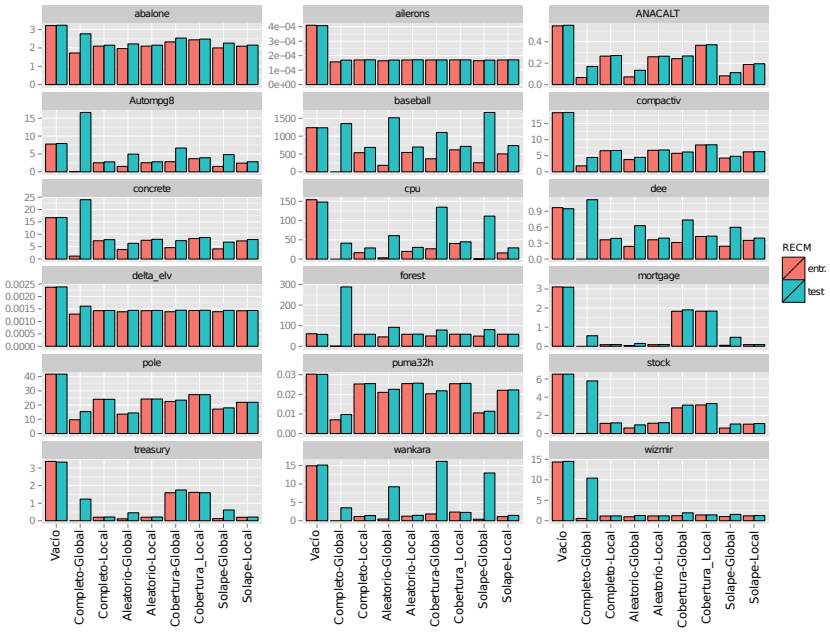
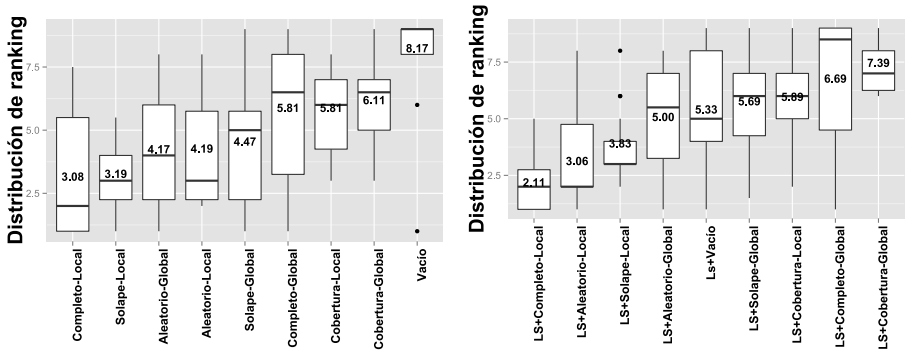


Figura 1: Calidad de los individuos iniciales (error de entrenamiento y test)

a error de test. Como se puede observar, Completo-Local es el que mejor ranking medio obtiene, aunque también presenta una gran varianza. La siguiente mejor estrategia es Solape-Local, consiguiendo un ranking medio similar al anterior. Sin embargo, en este caso la varianza es muchísimo menor. Posteriormente encontramos los individuos Aleatorio-Global y Aleatorio-Local. En este caso, los resultados obtenidos cuando se usa mínimos cuadrados global y local son similares. Aunque en media el global es mejor, el local parece ser ligeramente más estable (menor varianza) y la moda (ranking más frecuente) está en un puesto por debajo. Finalmente, la estrategia basada en la cobertura es la que peores resultados reporta. En nuestra opinión, y en base al análisis llevado a cabo, esto se debe a la forma en la que las reglas candidatas son generadas. En este proceso ya se seleccionan las reglas que tienen un mínimo grado de cobertura, por lo que al centrarse de nuevo en este mismo aspecto, esta estrategia no aporta información suficientemente discriminativa sobre la calidad de las reglas.

4.3. Análisis de la búsqueda local con diferentes inicializaciones

En esta sección analizaremos los modelos construidos tras aplicar la búsqueda local. Con el fin de estudiar en profundidad los resultados obtenidos, se han generado un total de cuatro figuras. En la primera (figura 3), se muestran los errores de entrenamiento y test obtenidos por cada inicialización y posterior búsqueda local para cada base de datos. En la figura 4 se muestran, tanto para entrenamiento como para test, la diferencia entre los errores obtenidos para las



(a) Sobre soluciones iniciales (b) Tras aplicar búsqueda local

Figura 2: Distribución de rankings en base al error de test

soluciones iniciales y tras aplicarles la búsqueda local. Finalmente, en las figuras 5 y 6 se muestran el número de reglas y el número de sistemas evaluados por la búsqueda local, respectivamente.

En primer lugar, si nos centramos en la figura 4, evidentemente para el caso del sistema vacío de reglas se obtienen los mejores decrementos de error, ya que son los individuos con menos información introducida y peor calidad. Sin embargo, la búsqueda local es capaz de evolucionar dicho sistema de tal manera que con pocas reglas (y pocos sistemas evaluados) se consiguen resultados competentes en comparación al resto de propuestas (ver figura 3). Otra cosa que se observa es que los individuos que utilizan la técnica de mínimos cuadrados global se estancan prematuramente en un óptimo local, ya que la búsqueda local finaliza habiendo evaluado muy pocos sistemas (ver figura 6).

Si atendemos a las gráficas 5 y 6, sorprende especialmente la estrategia LS+Completo-Local, ya que aunque parte del sistema completo de reglas, finalmente construye modelos con pocas reglas, equiparable a los modelos construidos por las estrategias que inluyen tan solo el 25 % de las reglas candidatas. En cambio, el número de sistemas evaluados es enorme, lo que se traduce en un alto coste computacional.

Finalmente, si comparamos las tres estrategias que utilizan el 25 % de las reglas candidatas entre sí, en el caso de usar mínimos cuadrados local (el global hace que la búsqueda no modifique en gran medida el individuo inicial), podemos ver que la estrategia LS+Cobertura es la peor de las tres. Como ya se vió en la subsección 4.2, ésta generaba las soluciones iniciales con peor calidad (tras el sistema vacío), y por la misma razón era de esperar que también tuviese un mal comportamiento tras aplicar la búsqueda local. Por el contrario, LS+Aleatorio y LS+Solape funcionan bien y se comportan de manera similar. Es especialmente sorprendente que LS+Aleatorio+Local consiga tan buenos resultados basando la información del individuo inicial en el azar. Esto, en nuestra opinión, es debido de nuevo a la forma en la que se generan las regla candidatas. Puesto que cada regla candidata debe superar unos umbrales de tal manera que tenga un mínimo

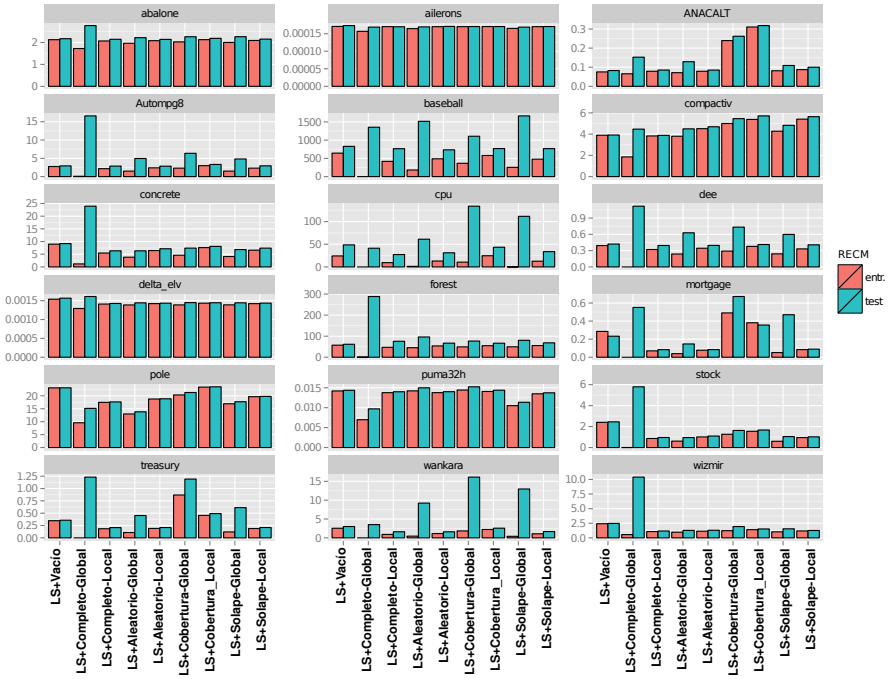


Figura 3: Error de entrenamiento y de test obtenido tras la búsqueda local

grado de cobertura y factor de calidad, el algoritmo de búsqueda local premia en mayor grado la diversidad que el grado de información de los individuos iniciales.

Por último, realizaremos una comparación estadística[4] entre las estrategias propuestas (dos a dos) a nivel de error de test, utilizando un nivel de confianza $\alpha = 0{,}05$. Para ello, en primer lugar aplicaremos un test de Friedman [11] para determinar si todas las propuestas son similares o por el contrario algún par de métodos es estadísticamente diferente. En la figura 2b podemos ver la distribución de rankings en cuanto a error de test. En ella se observa que, en media, la estrategia que mejor ranking obtiene es LS+Completo-Local, seguida de LS+Aleatorio-Local y LS+Solape-Local. El resto de enfoques parece estar un escalón por debajo en cuanto a error de test. Finalmente, el p-valor obtenido por el test de Friedman es $2{,}49 \cdot 10^{-9} \leq 0{,}05$ por lo que existe diferencia estadísticamente significativa entre ellos. A continuación procedemos con un test post-hoc por el método de Shaffer [12]. En la tabla 1 se muestran los p-valores obtenidos para cada par de estrategias. Como se puede observar, no existe diferencia estadísticamente significativa entre LS+Completo-Local (que era el que mejor ranking medio consiguió), LS+Aleatorio-Local y LS+Solape-Local. Sin embargo, LS+Completo-Local requiere muchísimo esfuerzo computacional, como se observa en la figura 6, ya que partiendo del sistema completo de reglas

[4] Este estudio, así como las gráficas generadas, ha sido realizado utilizando el paquete de R exreport: http://exreport.jarias.es/

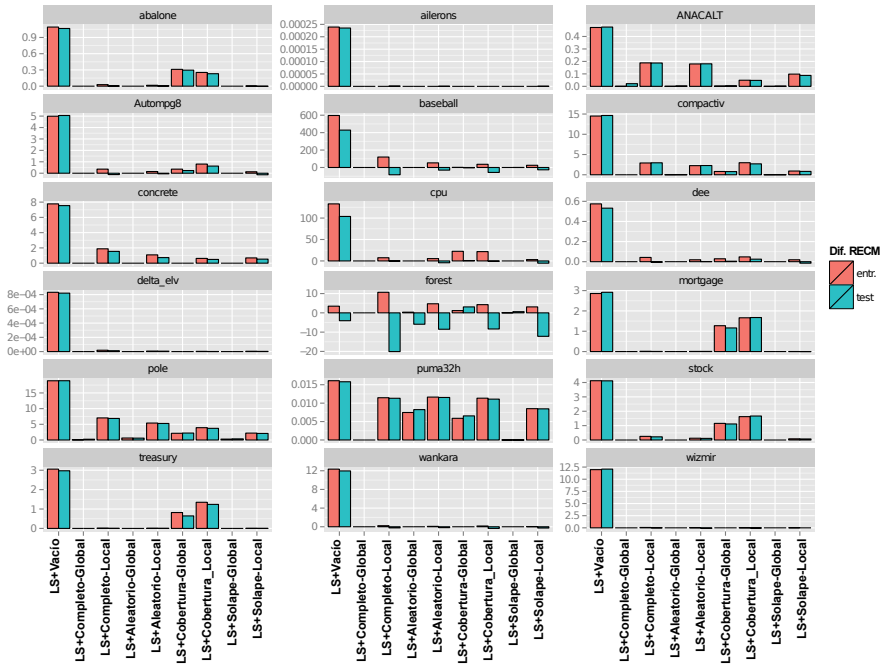


Figura 4: Diferencia de error de entrenamiento y test entre las soluciones iniciales y los sistemas obtenidos tras aplicar la búsqueda local

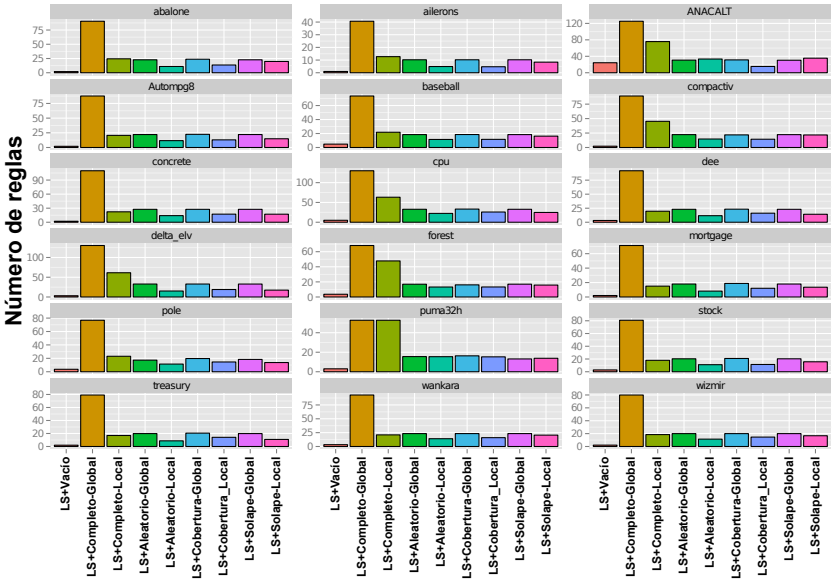


Figura 5: Error de entrenamiento y de test obtenido tras la búsqueda local

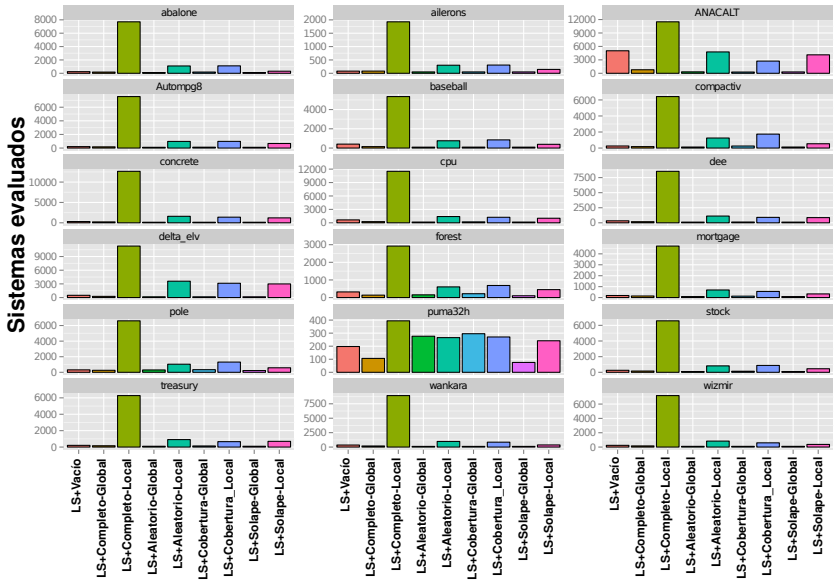


Figura 6: Error de entrenamiento y de test obtenido tras aplicar la búsqueda local

emplea muchas iteraciones para reducir el número de reglas del sistema. Respecto a los dos restantes, LS+Aleatorio-Local siempre necesita un mayor número de evaluaciones por tratarse de una solución inicial menos informada (ocurre en los 18 conjuntos de datos utilizados), por lo que necesita más iteraciones para conducir el sistema hacia un óptimo local. Sin embargo, el número de reglas es ligeramente menor en el caso de LS+Aleatorio-Local frente a LS+Solape-Local (solo en el problema puma32h LS+Solape-Local alcanzó las 13.8 reglas, mientras que LS+Aleatorio-Local 15.43), debido en nuestra opinión a la mayor diversidad introducida en estos sistemas iniciales de reglas.

Cuadro 1: Test post-hoc por el método de Shaffer

method	LS+A+L	LS+S-L	LS+A+G	LS+V	LS+S-G	LS+Cob-L	LS+Com-G	LS+Cob-G
LS+Completo-Local	**1.00e+00**	**1.00e+00**	4.35e-02	1.21e-02	2.68e-03	1.15e-03	1.80e-05	2.67e-07
LS+Aleatorio+Local	-	**1.00e+00**	**6.63e-01**	**2.90e-01**	**9.61e-02**	4.97e-02	2.15e-03	7.00e-05
LS+Solape-Local	-	-	**1.00e+00**	**1.00e+00**	**7.88e-01**	**5.35e-01**	4.65e-02	2.95e-03
LS+Aleatorio+Global	-	-	-	**1.00e+00**	**1.00e+00**	**1.00e+00**	**1.00e+00**	**2.13e-01**
LS+Vacío	-	-	-	-	**1.00e+00**	**1.00e+00**	**1.00e+00**	**5.35e-01**
LS+Solape-Global	-	-	-	-	-	**1.00e+00**	**1.00e+00**	**1.00e+00**
LS+Cobertura-Local	-	-	-	-	-	-	**1.00e+00**	**1.00e+00**
LS+Completo-Global	-	-	-	-	-	-	-	**1.00e+00**

5. Conclusiones

Este trabajo parte de la definición del algoritmo expuesto en [5]. En el mismo se probaron diferentes algoritmos para la generación de SBRDs tipo TSK-1, concluyendo que los mejores resultados eran los obtenidos por el enfoque basado en búsqueda local partiendo del sistema vacío de reglas. Sin embargo, el número de reglas contenido en los modelos generados por este algoritmo era muy reducido, por lo que en este trabajo estudiamos el comportamiento del mismo variando el punto de partida. Para ello, se han probado un total de 8 soluciones iniciales (además del vacío). Dos de ellas parten del sistema completo de reglas y el resto del 25 %. Para la asignación de los consecuentes a cada una de las reglas se han empleado dos alternativas: mínimos cuadrados global y local.

Para la experimentación se han utilizado un total de 18 problemas obtenidos del repositorio KEEL. En primer lugar se ha estudiado cada uno de los individuos iniciales en cuanto a capacidad de predicción. En esta fase se observó que la estrategia de mínimos cuadrados global, en general conlleva un gran problema de sobreajuste. Respecto a la estrategia basada en la cobertura, fue la que peores resultados consiguió (al margen del sistema vacío de reglas).

Finalmente, se estudió el efecto de usar cada una de las inicializaciones propuestas junto con la búsqueda local. Para el análisis estadístico, se aplicó un test de Friedman para determinar si había o no diferencia estadísticamente significativa entre las propuestas, seguido de un test post-hoc por el método de Shaffer. El nivel de confianza que se usó fue de $\alpha = 0{,}05$. El algoritmo LS+Completo-Local es el que menor ranking medió consiguió, aunque también es el que mayor coste computacional requiere, ya que parte del sistema completo de reglas y utiliza muchas iteraciones para eliminar reglas. Sin embargo, no existe diferencia significativa entre esta estrategia, LS+Aleatorio-Local y LS+Solape-Local, requiriendo muchísimo menos esfuerzo computacional estas dos últimas. En relación a ellos, por un lado LS+Aleatorio-Local efectua un mayor número de evaluaciones de sistemas ya que es un individuo menos informado. Por el contrario, LS+Solape-Local genera sistemas ligeramente más complejos (con un mayor número de reglas).

Agradecimientos. Este estudio ha sido parcialmente financiado por la JCCM bajo el proyecto PEII-2014-049-P. Javier Cózar también ha sido financiado por el MECD a través de la beca FPU12/05102.

Referencias

1. Cordón, O., Herrera, F.: A proposal for improving the accuracy of linguistic modeling. IEEE Transactions on Fuzzy Systems **8**(3) (2000) 335–344
2. Nozaki, K., Ishibuchi, H., Tanaka, H.: A simple but powerful heuristic method for generating fuzzy rules from numerical data. Fuzzy Sets and Systems **86** (1997) 251–270

3. Alcala-Fdez, J., Alcala, R., Herrera, F.: A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. Fuzzy Systems, IEEE Transactions on **19**(5) (2011) 857–872
4. Cózar, J., de la Ossa, L., Gámez, J.: TSK-0 fuzzy rule-based systems for high-dimensional problems using the apriori principle for rule generation. In: RSCTC. (2014) 270–279
5. Cózar, J., delaOssa, L., Gámez, J.A.: Generación de reglas difusas tipo tsk-1 basada en el principio apriori derivando el sistema de reglas mediante búsqueda local. In: Actas del X Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2015). (2014) 57–71
6. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning. Information Science **8** (1975) 199–249
7. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications for modeling and control. IEEE Transactions on Systems, Man, and Cybernetics **15**(1) (1985) 116–132
8. Mamdani, E.H.: Applications of fuzzy algorithm for control a simple dynamic plant. In: Proceedings of the IEEE 121(12). (1974) 1585–1588
9. Cózar, J., delaOssa, L., Gámez, J.A.: Un algoritmo grasp para el aprendizaje de sistemas basados en reglas difusas lingüísticas. In: Actas de la XV Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2013). (2013) 99–108
10. Herrera, F., Villar, P.: Un algoritmo genético para aprendizaje de un sistema basado en reglas difusas tipo takagi sugeno. In: Actas del V Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados. (2007) 543–548
11. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics (1940) 86–92
12. Shaffer, J.P.: Modified sequentially rejective multiple test procedures. Journal of the American Statistical Association **81**(395) (1986) 826–831