

Un modelo de inducción de reglas difusas mixtas aplicado a un problema real de fertilización asistida

Luis Arévalo, Estudiante,
Roberto Bonvallet, Estudiante,
Pablo Garrido, Estudiante,
Guillermo Oyarzún, Estudiante.

Resumen—Este documento da cuenta del proceso de la creación de un modelo de reglas para el problema de la fertilización asistida. Para ello, se presenta la forma en que se procesó la información de la base de datos y la aplicación del algoritmo de reglas difusas mixtas, propuesto en [2], considerando una correcta sintonización de los parámetros y una comparación de los resultados con el algoritmo de clustering PNC2 [1].

Términos Claves—Reproducción asistida, inducción de reglas, pnc2, mixed fuzzy rule, parallel coordinates.

I. INTRODUCCIÓN

A la hora de realizar reproducción asistida, intervienen muchas variables como la calidad del embrión, la tasa de fecundación y la calidad de la transferencia, entre otras. Al existir tantas variables involucradas, es muy difícil poder determinar a priori en qué grado influye cada una de ellas en el éxito del proceso, si es que lo hacen. Es por esto que una buena técnica para explicar las características de un fenómeno de manera intuitiva y natural es la inducción de reglas, que permite generar un modelo para poder predecir qué características debe tener el proceso para que sea exitoso, dado que existen muchas variables que influyen en las diferentes fases.

En la primera parte de este paper se detalla cuál es específicamente el problema a resolver, indicando la estructura y naturaleza de los datos, además de una explicación de por qué se necesita un modelo para explicar el proceso. Luego, en la sección 3 se describe los algoritmos utilizados para obtener un modelo predictivo e interpretativo: PNC2 y mFRL, junto con un sistema de visualización gráfica de reglas [5]. Con los algoritmos ya señalados, se procede a realizar ciertas pruebas que permitan acercarse a un modelo más explicativo; todo esto clarificado en la sección 4. Una vez que se realizan los experimentos necesarios, se interpretan los resultados obtenidos, los cuales son mostrados en la sección 5. Finalmente, en la sección 6, se dan a conocer las conclusiones del modelo generado, con sus respectivas restricciones, ventajas, desventajas y trabajo futuro.

L. Arévalo, R. Bonvallet, P. Garrido y G. Oyarzún son estudiantes de Ingeniería Civil Informática en la Universidad Técnica Federico Santa María, en Valparaíso, Chile. Email: {larevalo, rbonvall, pgarrido, goyarzun}@inf.utfsm.cl.

II. PROBLEMA

A. Descripción del problema

El problema abordado corresponde a una problemática actual que enfrenta la Clínica las Condes en el área de medicina reproductiva; específicamente referido a la *reproducción asistida*. El conflicto se presenta cuando la pareja desea tener un hijo y por diversas razones no puede hacerlo de forma natural. Para solucionar dicho problema, la clínica realiza principalmente dos métodos de reproducción asistida dependiendo de las incapacidades que tiene un integrante de la pareja (o eventualmente ambos): ICSI e IVF¹

En el proceso de reproducción, la clínica almacena todos los datos de relevancia en una base de datos necesaria para mantener un registro del paciente. El proceso, debido a la cantidad de variables involucradas, presenta cierto grado de éxito dependiendo de factores como: edad de la mujer, calidad de los embriones, características fisiológicas de la mujer, entre otros.

Debido a que no existe 100% de éxito en todos los tratamientos, la clínica necesita de un modelo que permita entender qué variables son cruciales en el éxito de la implantación y posterior fecundación del embrión en la mujer.

B. Estructura y naturaleza de los datos

De momento, la clínica ha provisto de una base de datos que contiene registros para casos de 0% implantación y 100% implantación. Ambos casos presentan leves diferencias en cuanto a la cantidad y similaridad de los atributos debido a que las características del proceso mismo varían según el grado de éxito o fracaso (por ejemplo: no se podría registrar información del parto en el caso de 0% implantación por causa del fracaso en el proceso).

En el caso de 0% implantación, los registros contienen 79 características (o variables), de los cuales 23 son variables categóricas, 4 continuas, 43 enteras y 10 compuestas². A su vez, esta parte presenta 1337 registros. Del total de registros, 29 de ellos presentaron en el proceso un clivaje³ a las 72 horas;

¹La explicación en extenso del problema se encuentra en [6].

²Estas variables describen tanto cuantitativa como cualitativamente un atributo, por ejemplo, *calidad de embriones* describe tanto la cantidad de nucléolos como la forma de estos. De esta forma, la variable compuesta se compone de una variable entera más una categórica o una variable continua más una categórica.

³Concepto descrito en [6].

por otro lado, a las 48 horas, 218 tuvieron clivaje. A su vez, se registran otras características del proceso como: simetría de pronúcleos, morfología de los ovocitos y disparidad en los pronúcleos. Dichas características completan los 1090 registros restantes y se enfocan en describir específicamente las propiedades mencionadas anteriormente; sin embargo, contienen información completa del proceso.

En el caso de 100% implantación, los registros contienen 86 variables, de las cuales 28 son variables categóricas, 7 continuas, 42 enteras y 9 compuestas. En este caso se tiene una menor cantidad de registros, registrando sólo 858. Del total de datos, 111 corresponden a un proceso que tuvo clivaje a las 48 horas, 19 tuvo clivaje a las 72 horas y los 299 restantes no presentaron clivaje en el medio externo. Adicionalmente, se tienen registros de la transferencia de los embriones, siendo 130 transferidos sin haber sido congelados y 299 congelados antes de ser transferidos. Al igual que para el caso de 0% implantación, estos últimos registros enfatizan un análisis en la transferencia pero aún registrando de forma completa el proceso.

En cuanto a las clases de salida, la información contenida en la base de datos hace referencia a las dos clases que fueron anteriormente mencionadas: 100% implantación y 0% implantación.

III. ALGORITMOS UTILIZADOS

A. Mixed Fuzzy Rule Algorithm (mFRL)

El algoritmo de reglas difusas mixtas [2] crea un conjunto de reglas caracterizadas por dos subconjuntos del espacio de características: una región de evidencia y una región de generalización, denominadas respectivamente núcleo y soporte de la regla. Cada regla asocia a los puntos cubiertos por su soporte una clase de salida. Además, cada regla tiene asociados un peso (número de ejemplos cubiertos) y un ancla (ejemplo que gatilló la creación de la regla).

En cada iteración (época) del algoritmo, se recorre secuencialmente los ejemplos de entrenamiento y se evalúa si son cubiertos por el conjunto de reglas creadas; según sea el caso, se realizan las siguientes operaciones:

- *covered*: si el ejemplo está correctamente cubierto por una regla, i.e. cae en su soporte y es de la clase predicha por la regla, se expande el núcleo de modo que cubra al ejemplo y se incrementa el peso de la regla.
- *commit*: si el ejemplo no es clasificado correctamente por ninguna de las reglas actuales, se crea una nueva regla cuyo núcleo comprende sólo este ejemplo. El soporte cubre todo el espacio, el peso de la regla es inicializado en 1 y el ancla es fijada en el ejemplo actual.
- *shrink*: en cualquiera de los casos anteriores, se encoge la cobertura de todas las reglas de clase distinta a la del ejemplo actual, para evitar que éste sea incorrectamente clasificado. Para el encogimiento, se elige una dimensión y se elimina del núcleo o del soporte, según corresponda, los valores del dominio que cubren al ejemplo en esa dimensión. Como criterio, se escoge la dimensión a lo largo de la cual el encogimiento resulte en la menor pérdida de volumen posible del soporte.

En [4], el autor propone varias heurísticas para determinar la pérdida de volumen al encoger una regla. Todas ellas son medidas de la reducción que significa eliminar valores del dominio de una variable:

- *domain-based shrink*⁴: mide la proporción del dominio que es encogida.
- *rule-based shrink*: mide la proporción del soporte que es encogida.
- *anchor-based shrink*: utiliza la distancia del ejemplo al ancla para ponderar la distancia encogida.
- *area-based shrink*: utiliza el tamaño de la sección del soporte, acotada por el núcleo, en que se encuentra el ejemplo,

Estas heurísticas son propuestas para variables continuas, pero pueden ser adaptadas para variables nominales, excepto las heurísticas anchor- y area-based, pues no existe noción de distancia entre los valores de los dominios categóricos.

El algoritmo de aprendizaje descrito anteriormente puede ser visualizado a continuación:

Algorithm 1 Pseudocódigo mFRL

```

while las reglas sufren cambios do
  for cada ejemplo de entrenamiento do
    if el ejemplo está correctamente cubierto then
      covered()
    else
      commit()
    end if
    shrink()
  end for
  if cambió el sistema de reglas then
    for cada regla do
      core ← (anchor)
      w ← 0
    end for
  end if
end while

```

Una vez creado el sistema de reglas, la clasificación se realiza evaluando el grado de membresía del patrón de entrada para cada regla, y asignándole la clase de la regla en la que tiene mayor membresía. La membresía de un punto en una regla se calcula evaluando en cada dimensión funciones trapezoidales con valor 1 en el núcleo, 0 fuera del soporte y transición lineal dentro del soporte. La membresía de la regla se obtiene a partir de estas evaluaciones según algún criterio; en este caso, se comparó dos criterios: el máximo y el mínimo de las membresías por dimensión.

B. PNC2

Para comparar los resultados del algoritmo anterior, se utilizó PNC2, propuesto e implementado por Lars Haendel [1]. PNC2 es un algoritmo de clustering aglomerativo y jerárquico que genera reglas del estilo if-then. El hecho de

⁴Esta heurística no es mencionada en [4]; se asignó este nombre a la heurística propuesta en FRL.

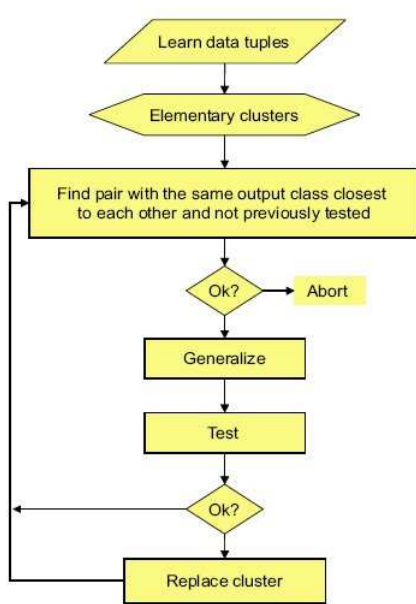


Fig. 1. Algoritmo PNC2.

que sea aglomerativo significa que considera inicialmente a cada ejemplo de entrenamiento como un cluster en sí, para luego iterativamente intentar fusionar clusters parecidos bajo ciertos criterios de similitud. Al igual que mFRL, las reglas (clusters) son representadas en el espacio como un hipercubo, y se define como criterio de similitud la distancia entre los hipercubos. Esta medida puede generar problemas cuando 2 clusters se encuentran juntos y ambos representan clases distintas. Para suplir esta falencia el algoritmo realiza un test antes de fusionar 2 clusters, comprobando si pertenecen o no a la misma clase. Otra de las características que posee este algoritmo es que permite trabajar con datos mezclados sin necesidad de crear modelos separados para los distintos tipos de atributos.

El esquema de generalización y formación de clusters de este algoritmo a partir de la data de puede ser visto en la figura 1.

C. Coordenadas Paralelas

Para poder visualizar de manera más intuitiva las reglas generadas, fue necesario crear un sistema de visualización gráfica que permitiera ver las n -dimensiones de la regla en forma más legible. Para ello, se ocuparon las coordenadas paralelas [5], que son capaces de visualizar n dimensiones en un sistema bidimensional. En este sistema, cada eje vertical (ordenada) representa un atributo (dimensión) de la regla que puede ser continuo o categórico.

La ventaja de la visualización gráfica es que permite diferenciar reglas que cubren vastas regiones en el espacio de aquellas reglas que cubren una región reducida o bien un ejemplo específico. De esta forma, se puede evaluar el grado de generalización y calidad de la regla.

En la figura 2 se muestra un ejemplo de 3 reglas generadas para el problema de la Iris [5], en donde existen 6 atributos y 3

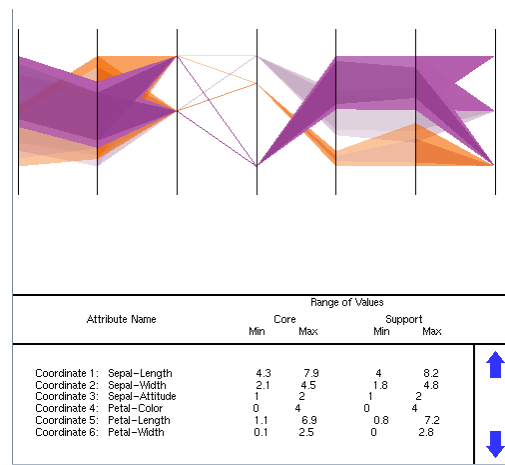


Fig. 2. Conjunto de reglas para la data de la Iris.

clases. Allí se puede apreciar que entre dos variables continuas (por ejemplo, Sepal-length y Sepal-width) se genera una región poligonal con 4 vértices, entre una variable categórica y una continua (Petal-Color y Petal-Length) se genera una región poligonal con tres vértices y entre dos variables categóricas (Sepal-Attitude y Petal-Color) se genera una línea que une un valor del dominio de una dimensión con la otra.

Además del manejo de diferentes tipos de variables, las coordenadas paralelas permiten visualizar las regiones de soporte y núcleo, las que se encuentran diferenciadas por la intensidad en el color que representa a la regla. Para el caso de la figura 2, la región de soporte es el color más claro y la región núcleo el color más opaco.

Finalmente, cabe destacar que la visualización es aún más necesaria cuando la cantidad de dimensiones es grande.

IV. EXPERIMENTOS

A. Preprocesamiento de los datos

Para aplicar los algoritmos explicados en la sección anterior, fue necesario realizar un preprocesamiento a las bases de datos provistas por los médicos. Lo primero que se realizó fue fusionar los registros de las tablas de 100% y 0% de implantación en una única tabla, distinguiendo cada uno de los casos mediante un atributo diferenciador. Si bien la mayoría de los atributos eran compartidos por ambas tablas de origen — sólo cerca de 8 atributos no coincidieron porque correspondían a atributos del parto —, aquellos atributos que no coincidieron fueron borrados ya que, por ejemplo, la información sobre el parto era de poca relevancia para lo que se intentaba modelar.

En una primera instancia la base de datos parecía no poseer valores nulos ya que todos los campos poseían información, pero, una vez que se comenzó a trabajar, se observó un patrón dentro de ciertos campos: cerca del 70% de los registros tenía dichos campos marcados como “información pendiente”. Tal patrón llevó a tomar la medida de eliminar esos atributos de la base de datos debido a que podían crearse reglas del tipo

if ‘no posee información’ then...

las cuales obviamente no significan nada. Al momento de aplicar el modelo sobre los datos preprocesados, se produjeron

reglas que gatillaron la eliminación de otros atributos de la data debido a que las reglas generadas sólo necesitaban de un atributo para realizar la clasificación, lo cual pareció muy extraño. Luego de una discusión, se llegó al acuerdo de eliminarlas, al considerar que no se conocía a ciencia cierta el proceso de registro de información y era posible que los números arrojados significasen que era información pendiente. Otro arreglo que se hizo fue separar una variable en dos o más, ya que existían campos con información compuesta, es decir, representaban 2 atributos y habían sido concatenados en un string, por ejemplo, para el endometrio el campo era: "9.5TRI", donde el número representaba el grosor y el texto representaba la textura. Este cambio permitió obtener información más ordenada y específica del problema.

Una vez terminada la etapa de preprocesamiento, la base de datos quedó configurada de la siguiente forma: 42 atributos (14 categóricos y 28 continuos) y 1352 registros.

B. Experimentos realizados

Una vez implementado el algoritmo mFRL, se comparó la influencia de la elección de la heurística de membresía (norma difusa) utilizada para la clasificación: al utilizar la norma mínimo, basta con que una dimensión del patrón por clasificar caiga fuera del soporte para que la membresía en la regla sea cero, lo que produce muchos "empates a cero" en puntos que están en zonas del espacio de características alejadas de las regiones de cobertura de las reglas. En el caso de la norma máximo, ocurre algo similar: basta con que uno de los atributos caiga dentro del núcleo para asignar membresía máxima al patrón en esa regla, incluso cuando en las otras dimensiones los valores pueden no estar ni siquiera cubiertos. Ante este resultado, se optó por utilizar la norma mínimo en los experimentos, pues se prefirió que ocurran empates en las zonas no cubiertas por las reglas (i.e. las zonas de donde el modelo no sabe mucho) a que ocurran asignaciones máximas en las zonas que el modelo mejor conoce.

También se analizó la influencia de las heurísticas de volumen sobre el algoritmo y su significado particular, ya que no existían buenas explicaciones en [4]. De las heurísticas propuestas para realizar la compresión de volumen para el caso continuo, se optó por utilizar la que consideraba la pérdida de volumen desde el soporte y no desde el núcleo, debido a que la esencia del algoritmo es minimizar la pérdida de volumen con la idea de maximizar la cobertura. Considerando lo anterior, La heurística que mejor representaba este concepto y que poseía mejores resultados fue la propuesta. Para el tratamiento de las variables nominales se desarrolló una heurística de volumen, ya que en los papers abordados no existían métodos explícitos para tratar este tipo de variables. Para ello, se consideró la idea original *greedy* de minimizar la pérdida de cobertura manteniendo la máxima cobertura posible. Así, se desarrolló un método que intentaba mantener siempre el *anchor* de la regla, es decir, el ejemplo que gatilló la creación de la misma. De esta forma, a la hora de reducir el soporte o el núcleo se obvian las dimensiones que pertenecen al *anchor* y, en caso de no poder reducir una regla, la mantiene así hasta la próxima iteración. De este modo, nunca quedan núcleos ni soportes vacíos.

```

if (Edad > 14  $\wedge$  Edad < 34)  $\wedge$  (Folículos > 15  $\wedge$  Folículos < 60)  $\wedge$  Tasa de recuperación < 70%  $\wedge$  Meiosis I < 1  $\wedge$  (Meiosis II > 5  $\wedge$  Meiosis II < 30)  $\wedge$  (Inseminados > 5  $\wedge$  Inseminados < 25)  $\wedge$  (2PN > 2  $\wedge$  2PN < 4)  $\wedge$  Tasa de Fecundación < 43%  $\wedge$  (No Fecundados > 2  $\wedge$  No Fecundados < 25)  $\wedge$  (Total Inseminados > 7  $\wedge$  Total Inseminados < 25)  $\wedge$  Fuente Ovocitos = 48 hrs.  $\wedge$  Cantidad Transferida < 3  $\wedge$  Calidad Transferida = Sin dificultad
then
    0% Implantación
end if

```

```

if Grosor Endometrio < 10.5  $\wedge$  Folículos < 6  $\wedge$  Ovocitos = 3  $\wedge$  Meiosis I = 0  $\wedge$  Meiosis II < 3  $\wedge$  E = 0  $\wedge$  Diferidos < 2  $\wedge$  2PN Dif = 0  $\wedge$  (No Fecundados > 3  $\wedge$  No Fecundados < 25)  $\wedge$  Citoplasma = (IVF  $\vee$  Normal  $\vee$  Oscuro)  $\wedge$  Calidad de Transferencia = (sin dificultad  $\vee$  sin dificultad con sedación)
then
    0% Implantación
end if

```

Fig. 3. Reglas generadas por PNC2.

Otra de las mejoras realizadas al algoritmo se llevó a cabo en la sección de la asignación de funciones de membresía para cada atributo, debido a que para atributos nominales se perdía la esencia de función difusa, es decir, no existía una suavidad dentro de los grados de membresía y estos cambiaban bruscamente de 1 a 0. Para resolver esta falencia, se creó un nivel intermedio dentro de esta función de membresía, con el valor de $1/(\text{tamaño del soporte})$; de esta forma, un atributo que poseía gran parte del soporte tenía una mejor ponderación que aquel que tenía una pequeña parte.

Dentro de la experimentación con los datos, se entrenó al algoritmo PNC2 con todos los datos para luego analizar su rendimiento a través de un muestreo. Para el caso del mFRL se realizó un entrenamiento y clasificación con la técnica de validación cruzada con 10 submuestras.

En la siguiente sección se presenta en profundidad los resultados obtenidos por ambos algoritmos acorde tanto a los cambios como a las mejoras hechas al algoritmo propuesto.

V. RESULTADOS

A. Reglas

Como se mencionó anteriormente, se ocupó el algoritmo PNC2 como base de comparación para ver el desempeño de mFRL. Considerando los datos obtenidos después del preprocesamiento, a continuación se presenta algunas de las reglas obtenidas.

PNC2: las siguientes reglas son aquellas que abarcaron la mayor cantidad de datos de entrenamiento, llegando a cubrir un 10% de los datos. Este algoritmo generó un total de 42 reglas. Las dos reglas con mayor cobertura se presentan en la figura 3.

mFRL: Este algoritmo generó 10 reglas. Las dos reglas con mayor cobertura se presentan en la figura 4.

Cabe destacar que las reglas correspondientes a los casos de 100% implantación abarcaban regiones más

if (Tasa de Recuperación > 0.6% \wedge Tasa de Recuperación < 100%) \wedge (Tasa de Fecundación > 0.57% \wedge < Tasa de Fecundación < 100%) **then**
 100% Implantación
end if

if (Tasa de Recuperación > 0.83% \wedge Tasa de Recuperación < 100%) \wedge (Tasa de Fecundación > 0.57 \wedge Tasa de Fecundación < 100%) \wedge (Total Inseminados > 1 \wedge Total Inseminados < 25) **then**
 100% Implantación
end if

Fig. 4. Reglas generadas por mFRL.

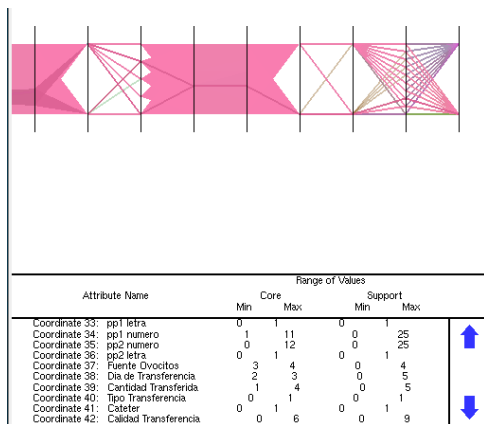


Fig. 5. Regla 1

amplias en los núcleos que los casos de 0% implantación. Esta fue una de las razones por la cual las reglas de 100% tuvieron mayor cobertura en el entrenamiento.

B. Visualización de las reglas

Se mencionó con anterioridad que mFRL generó 10 reglas. Parte de cada una de las reglas se puede apreciar en las figuras 5, 6, 7 y 8 ⁵. En estas reglas, la región más opaca es la región núcleo mientras que la más clara es la región de soporte.

Mirando en detalle las reglas, casi todas tienen el soporte del tamaño del dominio, pese a ello, la región correspondiente al núcleo difiere abarcando un sólo ejemplo en las reglas 5 y 6 y una región extensa para las reglas 7 y 8.

De las reglas puestas, las reglas 5 y 6 describen casos de 0% implantación mientras que las reglas 7 y 8 describen casos de 100% implantación. Esto indica que a pesar de que ambas tienen un soporte similar, las reglas correspondientes al caso de 100% implantación tienen más probabilidad de clasificar a ejemplos que son realmente de esa clase debido a que, en el caso de la norma de membresía *min*, asegura la clasificación donde hay más evidencia. Por lo tanto, a pesar de que existen

⁵Se tomó parte de la regla para ver más claramente el significado de la misma, sin embargo, el patrón a describir es similar en toda la regla. Por otro lado, sólo se tomaron 4 reglas representativas.

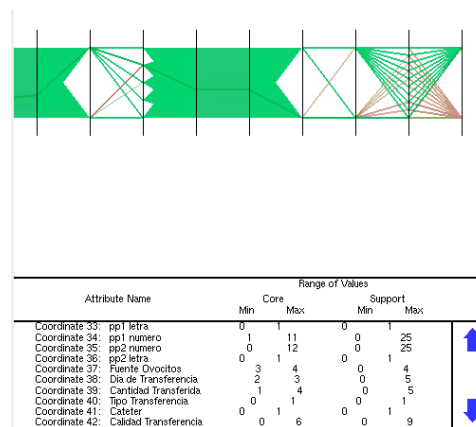


Fig. 6. Regla 2

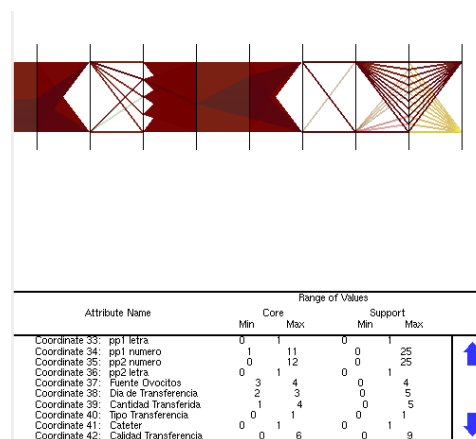


Fig. 7. Regla 6

varias reglas, hay algunas que nunca se ocupan porque la función de membresía las ve malas y sencillamente no las ocupa.

En el caso de las reglas obtenidas por PNC2, no se realizó un análisis gráfico debido a que su tratamiento era distinto y carecía de regiones de soporte y núcleo. Por otra parte, los resultados obtenidos por este algoritmo, eran claramente explicados en el programa por lo que no se vió la necesidad de ver la coherencia de las reglas de manera gráfica.

C. Clasificación

Para el algoritmo PNC2 se obtuvo los siguientes resultados después de realizar la clasificación:

Número de Reglas	42
Error Promedio	1%
# Reglas que explican mayor 10%	4

Para el algoritmo mFRL, los resultados de la validación cruzada fueron los siguientes:

	Domain-based	Anchor-based
Mal clasificados	2	3
Empate	1214	1161
Bien clasificados	136	188

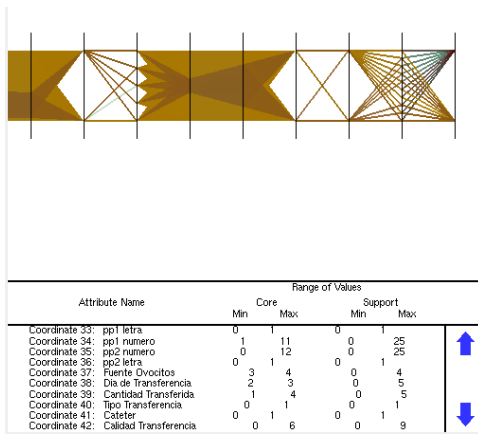


Fig. 8. Regla 7

La fila *empate* representa el número de patrones para los cuales hubo reglas de clases distintas con la misma membresía, y que por consiguiente no pudieron ser clasificados.

VI. CONCLUSIONES

A pesar de que el algoritmo PNC2 no fue visto en profundidad en cuanto a su desempeño en la clasificación y calidad de las reglas obtenidas, fue una buena base de comparación para saber si el algoritmo propuesto estaba arrojando resultados coherentes y si el bajo desempeño era producto de la naturaleza de los datos o del algoritmo mismo. Es cierto que sólo se ocuparon dos algoritmos para generar un modelo predictor, pero ayudó a ver que la eficiencia del mismo radicaba fuertemente en la naturaleza y la falta de información en la completación de la base de datos por parte de los médicos, lo que llevó a hacer muchos ajustes que quizás afectaron fuertemente la integridad y relación de los datos.

No está de más mencionar que el algoritmo propuesto para generar el modelo de inducción de reglas aplicado al problema de la clínica es un algoritmo simple y fácil de entender, y que es muy prometedor para trabajar con variables de todo tipo (específicamente con variables categóricas que se presentan frecuentemente en la vida real). Pese a ello, se tuvo que hacer muchas mejoras o cambios al algoritmo debido a la falta de detalle en cuanto a algunas heurísticas para evaluar volúmenes o coberturas (sobre todo en el caso del tratamiento de variables categóricas). Esto hace pensar que el trabajo realizado previamente en [2] y [4] necesita ser ahondado en cuanto al tratamiento de variables categóricas, ya que fue originalmente pensado para ello. Por otro lado, falta un análisis o demostración de la convergencia del algoritmo porque en el presente trabajo no se pudo demostrar que el algoritmo convergía pasado 5 épocas como se mencionaba en [2].

Uno de los aspectos importantes que faltó ver fue la jerarquización de las reglas que es abordada en detalle en [3]. Este queda como trabajo futuro debido principalmente a que permite generalizar las reglas haciéndolas menos específicas y más explicativas por sí mismas, ayudando a la interpretabilidad del modelo, el cual, fue uno de los aspectos más relevantes a evaluar durante el trabajo realizado.

- [1] Lars Haendels. *The PNC2 Cluster Algorithm. An integrated learning algorithm for rule induction*. Dortmund, Germany 2003.
- [2] Michael R. Berthold. *Mixed Fuzzy Rule Formation*. Abril 2002.
- [3] Thomas R. Gabriel, Michael R. Berthold. *Constructing Hierarchical Rule Systems*. California, USA.
- [4] Thomas R. Gabriel, Michael R. Berthold. *Influence of fuzzy norms and other heuristics on "Mixed fuzzy rule formation"*. October 2003.
- [5] Michael R. Berthold, Thomas R. Gabriel, A. Simona Pintilie. *Exploring Hierarchical Rule Systems in Parallel Coordinates*. IDA 2005, LNCS 3646, pp. 97-108.
- [6] Taller de Análisis Inteligente de Datos. *Estado del Arte—Inducción de Reglas*. Agosto, 2005.
- [7] Taller de Análisis Inteligente de Datos. *Planificación del Proyecto—Inducción de Reglas*. Octubre, 2005.