Extracción de Reglas utilizando estrategias adaptativas

Por

Lic. Laura Lanzarini

Directores

Ing. Armando De Giusti
Dr. José Angel Olivas Varela

Facultad de Informática
Universidad Nacional de La Plata

Tesis presentada para obtener el grado de
Doctor en Ciencias Informáticas

Marzo 2017

Resumen

En la actualidad, son numerosas las áreas interesadas en extraer conocimiento útil y novedoso a partir de información almacenada. La tecnología actual permite registrar todo tipo de procesos, en variados formatos y almacenarlo en forma local o subirlo a la nube con suma facilidad. Los datos se encuentran disponibles y contienen el registro de todo lo ocurrido. Esa información es producto de decisiones que fueron tomadas en distintos instantes de tiempo. Analizar los hechos pasados permite comprender los criterios utilizados y asociarlos con los resultados obtenidos ya sea que hayan sido positivos o negativos.

La Minería de Datos, una de las etapas más importantes del proceso de Extracción de Conocimiento o KDD (por su nombre en inglés *Knowledge Discovery in Databases*), cuenta con un conjunto de técnicas capaces de modelizar y resumir estos datos históricos, facilitando su comprensión y ayudando a la toma de decisiones. Su objetivo es generar una representación alternativa de la información que deje de manifiesto las relaciones existentes en ellos. Luego, a partir de su análisis e interpretación se podrá comprender por medio de la razón la naturaleza, cualidades y relaciones de los datos históricos, es decir, se podrá obtener conocimiento.

El proceso de KDD ha sido descripto por varios autores con distinto nivel de detalle. El consenso general reconoce al menos tres etapas: la primera tiene que ver con la manera en que se recolecta y prepara la información con la que se va a trabajar, la segunda se refiere a la construcción del modelo e incluye las técnicas de Minería de Datos y la última consiste del análisis e interpretación del modelo obtenido y eventualmente su comunicación a quienes deben tomar decisiones.

Esta tesis incluye un detalle de cómo llevar a cabo este proceso ejemplificando cada etapa a través de situaciones concretas. En la mayoría de los casos obtener resultados satisfactorios implica revisar muchas de las acciones realizadas incluso desde el inicio del proceso. Ya desde el momento en que la información fue recolectada se tomó la decisión de descartar seguramente ciertos aspectos del problema. No puede representarse o modelizarse lo que no se ha registrado. La información no puede generarse automáticamente; sólo puede transformarse para extraer a partir de ella las relaciones de interés. La calidad de la

información recolectada no sólo depende de la precisión empleada en la digitalización sino en su capacidad para describir el problema a resolver.

Cada técnica de Minería de Datos a utilizar tiene sus propias características. No todas pueden operar con datos faltantes o con información en cualquier formato. Es preciso saber primero que tipo de problema se quiere resolver y en base a eso elegir la técnica a aplicar. Una vez decidido esto, se conocerá la manera en que deben ser preprocesados los datos para obtener una vista minable adecuada.

Hay dos tipos de problemas que pueden ser resueltos: descriptivo y predictivo. El primero tiene que ver con hallar una representación que explique las características relevantes de los distintos grupos presentes en los datos. En esta dirección, en el marco de las investigaciones de esta tesis, se han resueltos varios casos reales en el área de la educación analizando la información académica de los alumnos que estudian carreras en Informática en tres Universidades Nacionales distintas: la Universidad Nacional de La Plata, la Universidad Tecnológica Nacional Facultad Regional La Plata y la Universidad Nacional de Río Negro. Más allá de las particularidades de cada uno, el tema común a todos los casos se relaciona con el desgranamiento académico y la deserción universitaria. El primer caso analizado fue la información de los alumnos de la carrera Licenciatura en Sistemas de la Sede Atlántida de la UNRN (Formia and Lanzarini, 2013). En esa oportunidad, se utilizó un método basado en proyecciones para seleccionar las características de los estudiantes y se aplicó un método de agrupamiento para determinar el perfil de los alumnos que abandonaban la carrera. Como conclusión relevante se detectó que existía una relación inversa entre el desempeño académico de los alumnos y la cantidad de horas que trabajaban y que la cantidad de alumnos con necesidades económicas era significativa. En este caso la recomendación fue arbitrar los mecanismos para ofrecer a los alumnos, según su perfil, becas de comida, transporte, laborales o de estudio con el objetivo de reducir sus necesidades e incrementar su dedicación a la carrera. Los resultados obtenidos para los alumnos de la UNLP fueron totalmente diferentes (Lanzarini et al., 2015a,b). Para este caso se representó el avance académico generando 5 atributos nuevos con la cantidad acumulada de asignaturas aprobadas por año. Luego a través de distintas visualizaciones se comprobó que los alumnos con mejores condiciones económicas tenían un desempeño académico entre malo y regular mientras que los que trabajan o manifestaban tener intenciones de hacerlo mostraban un mayor compromiso con sus estudios y aunque demoraban unos años más, tenían mayor chance de finalizarlos. También se observó que los alumnos que lograban mantener su ritmo académico durante el segundo año de estudio terminaban la carrera. Es entre primer y segundo año que se define la continuidad del alumno en la Facultad de Informática. En este caso la recomendación fue reforzar las tutorías a los alumnos entre el segundo cuatrimestre del primer año y todo el segundo año. Ese es el período de mayor vulnerabilidad para los estudiantes y donde parecen tener la mayor cantidad de dificultades

en sus estudios. Con respecto los alumnos de la UTN-FRLP los resultados fueron similares pero la población de alumnos tiene mayor edad y la proporción de alumnos que trabajan es mayor por lo que la relación entre trabajo y avance académico se acentúa (Baldino and Lanzarini, 2016).

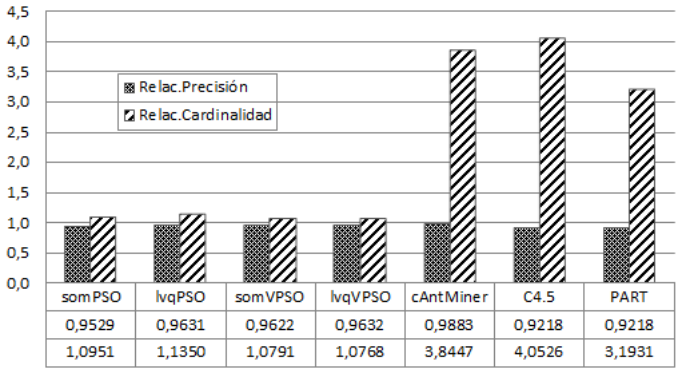
Contar con perfiles generales que describan las relaciones importantes es sumamente útil para comprender el estado de situación y actuar en consecuencia. Sin embargo, cuando se necesita responder a nuevas situaciones que no estuvieron presentes al momento de generar el modelo, se está frente a un problema predictivo. Su resolución requiere técnicas tales como: árboles de decisión, redes neuronales y reglas de clasificación. Estas últimas constituyen el tema central de esta investigación.

Esta tesis presenta una nueva técnica de Minería de Datos capaz de construir, a partir de la información disponible, un conjunto de reglas de clasificación con tres características principales: precisión adecuada, baja cardinalidad y facilidad de interpretación (Lanzarini et al., 2015c,d). Esto último está dado por el uso de un número reducido de atributos en la conformación del antecedente. Esta característica, sumada a la baja cardinalidad del conjunto de reglas permite distinguir patrones sumamente útiles a la hora de comprender las relaciones entre los datos y tomar decisiones.

El método propuesto hace uso de una variante original de la técnica de optimización basada en cúmulos de partículas PSO (Particle Swarm Optimization), definida por la autora de esta tesis en (Lanzarini et al., 2008), que incorpora la capacidad de operar con un cúmulo de tamaño variable haciendo uso del concepto de edad de una partícula. Esto hace innecesario tener que elegir a priori la cantidad de partículas a utilizar. Cuando el cúmulo es de tamaño fijo, pocas partículas no pueden realizar una búsqueda exhaustiva adecuada mientras que un número excesivo representará un tiempo de cómputo mayor.

En lo que se refiere al problema de extracción de reglas, el método propuesto utiliza el cúmulo para operar con una clase a la vez por lo que cada partícula evoluciona sólo el antecedente de una regla diferente. Para poder operar con atributos cualitativos y cuantitativos, el movimiento de la partícula se encuentra dividido en dos partes: una binaria y otra continua. La primera indica cuáles serán los atributos a utilizar mientras que la segunda determina los intervalos que controlan los atributos numéricos.

Para reducir el tiempo de la búsqueda se propone utilizar un agrupamiento de los datos de entrada a través de una red neuronal competitiva supervisada. Una vez entrenada, los centroides permitirán conocer cuáles son los lugares más prometedores del espacio de búsqueda facilitando la inicialización del cúmulo. También se propone un criterio para simplificar las reglas a medida que las partículas realizan la búsqueda. Esto se debe a que pequeñas modificaciones realizadas sobre los atributos nominales generan grandes cambios en el valor de aptitud de la partícula provocando que una regla que se estaba formando



somPSO	lvqPSO	somVPSO	lvqVPSO	cAntMiner	C4.5	PART
0,9529	0,9631	0,9622	0,9632	0,9883	0,9218	0,9218
1,0951	1,1350	1,0791	1,0768	3,8447	4,0526	3,1931

Figura 1: Resultados obtenidos sobre 13 Bases de Datos de repositorio

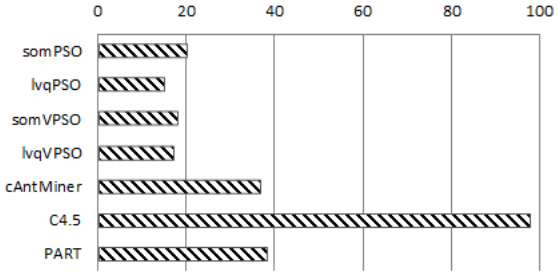


Figura 2: Simplicidad del modelo obtenido medida a partir de la cantidad de comparaciones totales realizadas en cada caso

adecuadamente pase abruptamente a tener un valor de aptitud nulo. Para evitar estas situaciones es preciso controlar los movimientos realizados.

El método propuesto opera iterativamente buscando cubrir los ejemplos de la clase mayoritaria. Cada vez que se utiliza PSO las partículas competirán entre ellas para formar la mejor regla de la clase seleccionada. Al hallarla, se retirarán del conjunto de entrada los ejemplos correctamente cubiertos y se volverá a repetir el proceso con los restantes hasta hallar el conjunto de reglas completo.

La figura 1 ilustra un análisis comparativo de los resultados obtenidos al aplicar cuatro variantes del método propuesto a 13 bases de datos del repositorio UCI (Bache and Lichman, 2013). La inicialización del cúmulo de partículas se realizó de dos formas distintas: con una red SOM y con una red LVQ. Cuando se trabajó con poblaciones de tamaño fijo se utilizaron redes neuronales de 30 neuronas competitivas. Para la red SOM se utilizó una grilla de 6x5 con 4 vecinas como máximo por neurona. Estas combinaciones aparecen mencionadas en la figura 1 como somPSO y lvqPSO. Las versiones con población variable, denominadas somVPSO y lvqVPSO, utilizan la estrategia de edad para modificar

la cantidad de individuos y pueden comenzar con una población menor ya que agregarán o quitarán partículas durante el proceso de búsqueda según consideren adecuado. Los resultados obtenidos fueron comparados con los arrojados por los siguientes métodos existentes en la literatura: PART (Frank and Witten, 1998a), cAntMinerPB (Medland et al., 2012) y C4.5 (Quinlan, 1993).

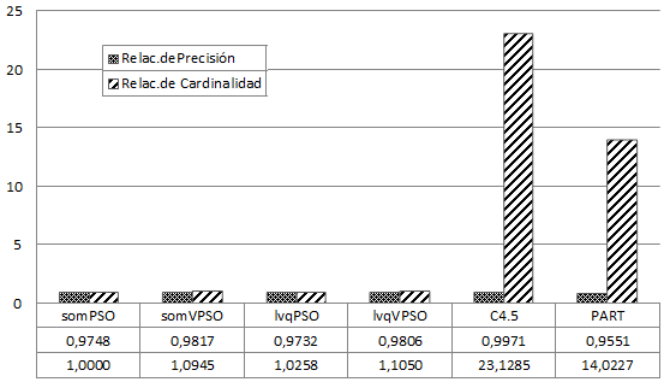
Dado que se trata de conjuntos de datos con distintas características, se han normalizado los resultados obtenidos en lo referido a precisión, cardinalidad del conjunto de reglas y longitud promedio por antecedente de una regla. En la figura 1 se denomina "Relac.Precisión" al promedio de los cocientes entre la precisión obtenida por el método y la precisión de la mejor solución para cada base de datos. Aplicando el mismo razonamiento, "Relac.Cardinalidad" es el promedio de los cocientes entre las cantidades promedio de reglas que posee la solución hallada por el método y la solución con cardinalidad mínima para cada base. Como puede observarse en la figura 1, las cuatro variantes del método propuesto tienen una cardinalidad parecida y son las de menor tamaño. En particular, lvqVPSO es, en promedio, un 7% superior a la solución de menor tamaño mientras que los restantes incrementan este valor entre un 284% y un 300%. Es decir que existe una gran diferencia entre la cardinalidad de la solución del método propuesto y los restantes métodos analizados. Con respecto a la precisión cAntMiner es el método que ofrece los mejores resultados siendo un 2% superior al método propuesto pero como se dijo anteriormente, para lograr este 2% de mejora en la precisión utiliza en promedio un conjunto de reglas cuya cardinalidad casi cuadruplica la del método propuesto.

La simplicidad del modelo obtenido por cada método puede verse en la figura 2 y fue estimada calculando la cantidad promedio de comparaciones que utiliza cada uno, es decir, el producto entre la cardinalidad del conjunto de reglas y la longitud promedio del antecedente en cada caso. Allí se observa que las variantes del método propuesto requieren menos del 50% de las utilizadas por cAntMiner, aproximadamente el 15% de las necesarias para C4.5 y el 35% de las empleadas por PART.

También se lo ha utilizado en dos casos reales de empresas financieras que otorgan préstamos para consumo y dos bases de datos financieros de crédito al consumidor del repositorio UCI Machine Learning Repository (Bache and Lichman, 2013). Una de las bases de datos reales proviene de una importante entidad de ahorro y crédito de Ecuador con más de 20 años de trayectoria en el mercado interno. La otra base de datos real pertenece a la Cooperativa de Ahorro y Crédito, una institución de ahorro mutuo de Ecuador.

Los préstamos para consumo son operaciones con montos muy inferiores a los préstamos hipotecarios y requieren tomar decisiones rápidas ya que generalmente son acordados con los clientes a través de un servicio en línea.

La figura 3 resume los resultados obtenidos para estos cuatro conjuntos de datos. Los resultados de la aplicación de las cuatro variantes del método propuesto han sido



somPSO	somVPSO	lvqPSO	lvqVPSO	C4.5	PART
0,9748	0,9817	0,9732	0,9806	0,9971	0,9551
1,0000	1,0945	1,0258	1,1050	23,1285	14,0227

Figura 3: Resultados obtenidos sobre dos casos reales de empresas financieras que otorgan préstamos para consumo y dos bases de datos financieros de crédito al consumidor de repositorio.

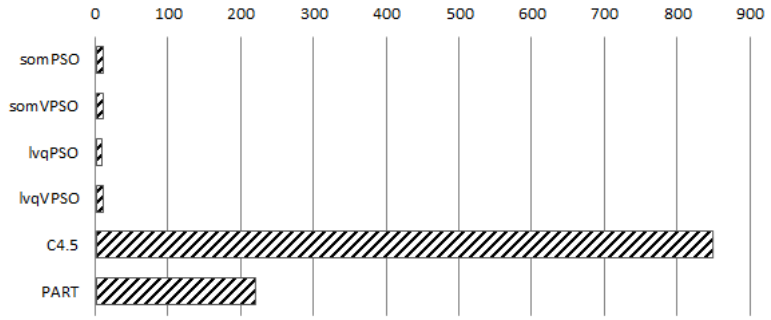


Figura 4: Simplicidad del modelo.

comparados con los métodos C4.5 y PART. En esta oportunidad no se ha incluido cAntMiner por el excesivo tiempo requerido por este método para brindar el conjunto de reglas.

Nuevamente se observa que todas las variantes del método propuesto arrojan los conjuntos de reglas con la menor cardinalidad. Se recuerda que el valor 1 en “Relac.Cardinalidad” representa que se ha obtenido el conjunto con la menor cantidad de reglas. Los otros métodos son 23 o 14 veces más grandes. La versión lvqPSO es la de mejor desempeño entre las variantes propuestas con una precisión promedio un 1% inferior a C4.5 pero si se observa la simplicidad graficada en la figura 4, equivalente a la cantidad de condiciones promedio que involucra el modelo completo puede verse que es 80 veces menor requiriendo un promedio de 10 preguntas para decidir si otorga o no el préstamo contra las más de 800 que plantea C4.5. El 1% de precisión es aceptable en este tipo de situaciones ya que el volumen de operaciones compensará las decisiones incorrectas. Sin embargo, tomar una decisión en base a un cuestionario corto que no tendrá mas de 10 preguntas incrementa notablemente las posibilidades de concretarlo. Ningún cliente soportará en línea un cuestionario tan

extenso como el que proponen los otros métodos. Con base en estos resultados se considera que el objetivo originalmente planteado ha sido cumplido.

Finalmente, se han incluido como anexos algunos trabajos relacionados resueltos durante el desarrollo de esta tesis con partes del modelo propuesto.

Palabras Claves: Minería de Datos, Reglas de Clasificación, Estrategias Adaptativas, Optimización mediante Cúmulo de Partículas, Mapas Auto-Organizativos.

Agradecimientos

Esta tesis es el resultado de varios años de trabajo. Son muchas las personas que de una forma u otra han colaborado para que pudiera terminarla pero sin duda el Ing. Armando De Giusti es el responsable de que, a lo largo de todo este tiempo, nunca perdiera de vista la necesidad de finalizarla. Sin su guía no hubiera podido lograrlo.

También quiero agradecer al Dr. José Angel Olivas Varela por acompañarme en el desarrollo de esta tesis aportando ideas y colaborando en la revisión de este documento.

Quiero expresar mi reconocimiento a todos los que han trabajado conmigo y a los que después de varios años aún lo siguen haciendo; docentes-investigadores de la UNLP y de otras universidades, becarios y tesistas de grado y postgrado, junto a quienes he investigado y resuelto tantos problemas.

Finalmente, pero no por eso menos importante, quiero agradecer a mi familia por la infinita paciencia que tienen para conmigo y en especial a mi marido, el amor de mi vida, quien me acompaña haciendo suyo cada nuevo proyecto que emprendo.

A todos, sinceramente, muchas gracias.

TABLA DE CONTENIDOS

INDICE DE TABLAS

INDICE DE FIGURAS

FIGURE **Page**

CAPITULO 1

INTRODUCCIÓN

Disponer de herramientas capaces de representar de una manera descriptiva las decisiones tomadas resultan sumamente útiles para explicar lo hecho y elegir el camino a seguir ante nuevas situaciones. Cuanto más simple de comprender sea dicha representación, más fácil será decidir cómo y cuándo utilizarla.

Esta tesis hace foco en la obtención automática de proposiciones condicionales o reglas con el objetivo de construir una representación con capacidad para explicar la manera en que se encuentra organizada la información histórica e identificar las razones por las cuales se tomaron ciertas decisiones.

Se comenzará describiendo brevemente los motivos por los que se ha estudiado el tema de extracción de reglas para luego exponer los objetivos y aportes de la tesis. A modo de resumen se indicará el listado de publicaciones que sustentan este trabajo y que serán explicadas a lo largo de todo este documento. Por último se describe el contenido y la finalidad de cada capítulo.

1.1 Motivación

En la actualidad, son numerosas las áreas interesadas en extraer conocimiento útil y novedoso a partir de información almacenada. La tecnología ha facilitado la recolección de información y permitido la generación de grandes repositorios de datos. Originalmente, se aplicaron soluciones estructuradas para su procesamiento y se utilizaron distintos mecanismos de consulta para buscar relaciones. Luego, este enfoque fue reemplazado por

una alternativa más prometedora basada en extraer las relaciones subyacentes sin disponer de una hipótesis previa. Así, a fines del siglo XX, comenzaron a surgir publicaciones tales como (Piatetsky-Shapiro and Frawley, 1991) y (Fayyad et al., 1996b) donde se destacó la importancia de contar con mecanismos para extraer automáticamente patrones o relaciones a partir de la información disponible dando lugar al conocido Proceso de Extracción de Conocimiento o *KDD* (sigla correspondiente a su nombre en inglés *Knowledge Discovery in Databases*).

En este contexto, la Minería de Datos, una de las etapas del proceso de KDD, ha recibido la atención de numerosos sectores al extremo tal de que su nombre sea considerado sinónimo del proceso completo. Esto se debe a que la Minería de Datos reúne al conjunto de técnicas capaces de modelizar y resumir datos históricos, facilitando su comprensión y ayudando a la toma de decisiones. Se trata de un área de investigación que ha recibido la atención de distintos sectores. Empresarios y académicos, por motivos muy diferentes han contribuido al desarrollo de distintas técnicas capaces de analizar la información disponible con el objetivo de extraer relaciones o patrones nuevos.

En resumen, quienes recién se inician en el tema tienden a pensar que las técnicas de Minería de Datos son las únicas responsables de los resultados que se puedan obtener sin advertir que en realidad no son otra cosa que una parte de un gran proceso y que como tal debe ser trabajado en su totalidad para obtener la solución buscada.

Ante un problema concreto la primera pregunta que debe hacerse es qué tipo de conocimiento se pretende extraer. La respuesta a esta pregunta tiene dos opciones: o se espera tener la capacidad de describir e identificar las características más importantes de la información recolectada o bien se espera extraer el conocimiento suficiente como para poder responder a nuevas situaciones imitando el comportamiento realizado en el pasado. En el primer caso se dice que el tipo de conocimiento a extraer es descriptivo mientras que en el segundo caso es predictivo.

El tema de esta tesis se enmarca en la resolución de problemas predictivos. Se busca construir a partir de información histórica una herramienta informática capaz de aprender el criterio utilizado para responder en situaciones pasadas. Existen distintas alternativas para cumplir con este objetivo pero si se piensa en quienes finalmente deban hacer uso de esta herramienta resulta fundamental tener la posibilidad de explicar la respuesta sugerida. Esta no es una característica que siempre se verifica. Por ejemplo, soluciones basadas en árboles y/o reglas tienen mayor capacidad explicativa que las redes neuronales o las máquinas de vectores de soporte. Si bien existen distintos modelos, aquellos que posean la capacidad de explicarse a si mismos, serán los elegidos por quienes deben tomar decisiones.

La capacidad explicativa del modelo pasa por resumir e identificar adecuadamente las características más relevantes. Existen técnicas capaces de organizar la información de

manera jerárquica, como por ejemplo, en forma de árbol. Son estructuras, que aunque en ocasiones resulten excesivamente grandes, permiten ver en su parte superior las características más importantes del problema. En ese sentido, poseen cierta capacidad descriptiva y puede reducirse su tamaño a través de distintos criterios de poda estableciendo una relación de compromiso entre una correcta cobertura de la información y el tamaño del modelo.

Por otro lado, a diferencia de los árboles cuyas técnicas de obtención se basan en dividir continuamente a los datos disponibles de la mejor manera posible, las reglas plantean dar una correcta cobertura a la mayoría de los ejemplos brindados para una cota de error dada. En ambos casos, el objetivo es obtener un modelo sencillo que permita explicar las decisiones tomadas y a la vez posea una tasa de acierto aceptable.

Cuando se habla de reglas, se trata de proposiciones expresadas en forma condicional; es decir que su estructura es de la forma:

```
SI (ocurre esta situación) ENTONCES (también se verifica esta otra)
```

son las preferidas a la hora de caracterizar esa enorme cantidad de datos históricos que fueron guardados automáticamente.

Lamentablemente, la mayoría de los métodos existentes, cubre la información histórica utilizando un conjunto de reglas generalmente extenso y complejo que, pese a tener la forma `SI-ENTONCES`, se torna prácticamente ilegible. Este es el problema que se busca resolver con este trabajo.

1.2 Objetivos

Esta tesis presenta una nueva técnica de Minería de Datos capaz de construir, a partir de la información disponible, un conjunto de reglas de clasificación que posea tres características fundamentales: precisión aceptable, cardinalidad baja y simplicidad en la definición de las reglas. Esto último está dado por el uso de un número reducido de atributos en la construcción del antecedente. Esta característica, sumada a la baja cardinalidad del conjunto de reglas permite distinguir patrones sumamente útiles a la hora de comprender las relaciones entre los datos. Estas cualidades hacen que el modelo obtenido posea una gran capacidad descriptiva resultando sumamente útil para la toma de decisiones.

El método propuesto ha sido aplicado sobre varios conjuntos de datos, tanto de repositorio como reales, demostrando que ambas características se verifican aunque en algunos casos su precisión sea ligeramente superada por otros métodos existentes. Esto tiene que ver con la presión realizada por mantener la simplicidad del modelo no permitiendo la generación de reglas con poca cobertura.

1.3 Contribuciones

- La descripción del proceso de *KDD* se realiza desde un punto de vista aplicado. A diferencia de la literatura convencional donde sólo se describen las etapas, se muestra la resolución de varios problemas reales que ejemplifican la manera de obtener una solución concreta. La extracción de conocimiento se plantea como un objetivo global cuyas partes deben ser revisadas durante su desarrollo para lograr los resultados esperados.

- Se describen alternativas originales de la técnica PSO. Esta técnica, cuyo uso se ejemplifica en distintas áreas, se utiliza en la literatura como una metaheurística poblacional con tamaño de población fija. La cantidad de partículas a utilizar no es un tema menor ya que un bajo número limita la capacidad exploratoria del método mientras que un número excesivo incrementa innecesariamente el tiempo de cómputo. En esta tesis se introduce una variante original para resolver este problema a partir del concepto de edad de la partícula, un valor numérico que acota su tiempo de permanencia en el cúmulo. Este concepto es totalmente nuevo ya que en PSO las partículas sólo se desplazan por el espacio de búsqueda sin eliminar ni agregar nuevos individuos. Además, las partículas del cúmulo, durante la adaptación, van reduciendo su inercia siendo atraídas por su propia mejor versión y por la que haya hallado la mejor solución hasta el momento. Esto genera una oscilación cerca del óptimo que en ocasiones dificulta la convergencia. En esta tesis también se propone una alternativa original para resolver este problema. Ambas modificaciones fueron utilizadas para dar lugar al método propuesto en esta tesis.

- El aporte central de esta investigación es la definición y presentación de un nuevo método capaz de obtener un conjunto de reglas de clasificación de cardinalidad baja que sea fácil de comprender. Se espera que pueda ser utilizado tanto como modelo predictivo como descriptivo. Por tal motivo, se busca que la longitud de las reglas sea sumamente reducida. Para lograrlo se hará uso de las variantes mencionadas anteriormente.

- Se incluye en el documento la comparación de los resultados obtenidos con el método propuesto y con otros existentes en la literatura al ser aplicado a bases de datos de repositorio. También se han analizado dos casos reales de empresas financieras que otorgan préstamos para consumo. Se trata de volúmenes grandes de operaciones con montos reducidos. En estos casos la velocidad en la toma de decisiones es fundamental y el método propuesto es una solución sumamente adecuada para estos casos.

- Se incluyen en forma de anexos algunos casos reales resueltos durante el desarrollo de esta investigación utilizando partes del modelo final.

1.4 Publicaciones

Esta tesis está respaldada por las siguientes publicaciones:

1.4.1 Publicaciones en Revistas

- **Simplifying Credit Scoring Rules using LVQ+PSO**. Laura Lanzarini, Augusto Villa Monte, Aurelio Fernández, Patricia Jimbo Santana. Kybernetes: The International Journal of Systems & Cybernetics. Publisher: Emerald Group Publishing Limited. vol. 46, nro 1., pp.8-16, doi. 10.1108/K-06-2016-0158, ISSN 0368-492X, 2017.

- **SOM+PSO: A Novel Method to Obtain Classification Rules**. Laura Lanzarini, Augusto Villa Monte, Franco Ronchetti. Journal of Computer Science & Technology. Vol 15. No 1. pp.15 - 22. ISSN 1666-6038. Abril 2015.

- **Selección de atributos representativos del avance académico de los alumnos universitarios usando técnicas de visualización. Un caso de estudio** Laura Lanzarini, María Emilia Charnelli, Guillermo Baldino, Javier Díaz. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología (TE&ET). Red de Universidades Nacionales con Carreras de Informática (RedUNCI). 2015 nro. 15. pp. 42-50- ISSN 1850-9959. Junio 2015.

- **Caracterización de la deserción universitaria en la UNRN utilizando Minería de Datos. Un caso de estudio**. Sonia Formia, Laura Lanzarini, Waldo Hasperué. Revista TE & ET; no. 11, pp. 92-98. ISSN 1850-9959. Diciembre de 2013.

- **E-Mail Processing with Fuzzy SOMs and Association Rules**. Laura Lanzarini, Augusto Villa Monte, César Estrebou. Journal of Computer Science and Technology. Vol 11, nro. 1. pp. 41-46. ISSN 1666-6038. Abril 2011.

1.4.2 Capítulos de libro

- **Obtaining classification rules using lvqPSO**. Laura Lanzarini, Augusto Villa Monte, Germán Aquino, Armando De Giusti. Advances in Swarm and Computational Intelligence. Lecture Notes in Computer Science. Vol 6433, pp. 183-193, doi. 10.1007/978-3-319-20466-6_20, ISSN 0302-9743. Springer-Verlag Berlin Heidelberg. Junio 2015.

- **Obtaining Classification Rules Using LVQ+PSO: an application to Credit Risk**. Laura Lanzarini, Augusto Villa Monte, Aurelio Fernández, Patricia Jimbo Santana. Scientific Methods for the Treatment of Uncertainty in Social Sciences.

Advances in Intelligent Systems and Computing. Springer-Verlag Berlin Heidelberg. vol. 377. pp 383-391, doi. DOI: 10.1007/978-3-319-19704-3_31. ISSN 2194-5357. Junio 2015.

- **A new Binary PSO with velocity control**. Laura Lanzarini, Javier López, Juan Maulini, Armando De Giusti. Advances in Swarm Intelligence. Lecture Notes in Computer Science. Vol.6728, pp.111-119, doi. 10.1007/978-3-642-21515-5_14. ISSN 0302-9743. Springer Berlin / Heidelberg. Junio 2011

- **Particle Swarm Optimization with Variable Population Size**. Laura Lanzarini, Victoria Leza, Armando De Giusti. Artificial Intelligence and Soft Computing. Lecture Notes in Computer Science. Vol 5097. pp. 438-449, doi 10.1007/978-3-540-69731-2_43. Springer Berlin / Heidelberg. ISSN 0302-9743. Junio 2008

- **Face Recognition using SIFT descriptors and Binary PSO with velocity control**. Maulini, Lanzarini. Publicado en el Libro Computer Science & Technology Series – XVII Argentine Congress of Computer Science Selected Papers. ISBN 978-950-34-0885-8. pp. 43-53. EDULP. 2012.

1.4.3 Congresos con referato

- **An exploratory analysis of methods for extracting credit risk rules** Patricia Jimbo, Augusto Villa Monte, Enzo Rucci, Laura Lanzarini, Aurelio Bariviera. XVII Workshop Agentes y Sistemas Inteligentes (WASI). XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016). pp. 834-841. 2016.

- **Academic Performance of University Students and its Relation with Employment**. Laura Lanzarini, María Emilia Charnelli, Javier Díaz. XLI Conferencia Latinoamericana en Informática (CLEI 2015) - XXIII Simposio Iberoamericano de Educación Superior en Computación. pp. 1-6. doi: DOI: 10.1109/CLEI.2015.7360017. 2015.

- **Face recognition based on fuzzy probabilistic SOM**. Lanzarini, Ronchetti, Estrebou, Lens, Fernández. 2013 IFSA World Congress - NAFIPS Annual Meeting. IEEE Catalog Nro.: CFP13750-USB. pp. 310-314. doi. 10.1109/IFSA-NAFIPS.2013.6608418. ISBN: 978-1-4799-0347-4. Edmonton, Canadá. 2013.

1.5 Organización de la tesis

Esta tesis está organizada de la siguiente forma:

- Este capítulo 1 brinda una introducción al tema de esta tesis justificando la importancia de contar con modelos con una gran capacidad descriptiva que ayuden a identificar y comprender los patrones que se encuentran presentes en los datos como herramienta para la toma de decisiones. Se define el objetivo de esta investigación así como las publicaciones científicas que lo respaldan. Finalmente se detalla la organización del documento.

- En el capítulo 2 se analiza el estado del arte en lo que se refiere a los distintos métodos de extracción de reglas de clasificación. También se incluye un breve resumen del camino recorrido en temas relacionados con la selección de atributos. Este es un aspecto importante en el proceso de construcción de reglas ya que reduce la búsqueda de las condiciones adecuadas que conformarán los antecedentes de las reglas. En este punto se hará una mención especial a las técnicas de agrupamiento o técnicas de clustering. Este tema es fundamental para determinar el punto de partida utilizado por el método propuesto para construir las reglas de clasificación. También se analizará la aplicación de distintas estrategias de agrupamiento en la resolución de problemas concretos.

- El capítulo 3 introduce las Técnicas de Optimización haciendo énfasis en la optimización por cúmulo de partículas o PSO (Particle Swarm Optimization). Aquí se describe el trabajo realizado en este tema y se presentan las variantes originales diseñadas para resolver varios aspectos no deseados del método original.

- El capítulo 4 contiene el aporte central de esta tesis ya que es el que contiene la descripción del método propuesto. Se trata de un nuevo método adaptativo capaz de extraer un conjunto de reglas reducido con pocas condiciones en sus antecedentes y con una tasa de acierto aceptable. También se detallan los resultados obtenidos y se los compara contra otros métodos descriptos en el capítulo 2.

- Finalmente el capítulo 5 contiene las conclusiones finales y algunas líneas de trabajo futuras.

CAPITULO 2

EXTRACCIÓN DE REGLAS

Justificar las decisiones tomadas es útil desde todo punto de vista. Conocer los motivos que permiten elegir el camino a seguir resulta fundamental ya sea para conseguir consenso o para explicar una inversión importante, aunque esta última no sea precisamente económica.

El avance de la tecnología ha facilitado el registro digital de todo tipo de operaciones. Independientemente del formato utilizado, se han documentado numerosas acciones pasadas que reflejan con aciertos y fracasos el camino transitado.

Este capítulo presenta una mirada general sobre la manera en que puede obtenerse conocimiento a partir del análisis de situaciones históricas. Si hay algo de lo que se dispone hoy en día, es de información. Comprender la manera en que se ha trabajado en el pasado permite identificar los criterios utilizados evidenciando los patrones que regulan las reglas de negocio de nuestro proceso o problema a resolver.

Este capítulo contiene una descripción del marco en el que se desarrolla el método propuesto en esta tesis, ilustrado a través del análisis de casos concretos y algunos trabajos relacionados con los que luego se comparará el desempeño del método propuesto.

2.1 Motivación

El vertiginoso avance de la tecnología ha permitido la generación de enormes repositorios de información digital asociada a procesos de muy diversa índole. Lo que originalmente

comenzó con el registro de transacciones comerciales se ha extendido a otras áreas como la salud, la educación y la gestión pública, entre otros. La informática ha penetrado transversalmente en casi la totalidad de los procesos actuales. Esto ha llevado a disponer de registros detallados de situaciones previas que documentan las decisiones tomadas y los resultados obtenidos oportunamente.

En los últimos años, ha crecido considerablemente el interés por extraer conocimiento a partir de la información disponible de forma automática dando lugar a lo que se conoce como "Proceso de Extracción de Conocimiento a partir de Bases de Datos" o *KDD* (del inglés *Knowledge Discovery from Databases*).

Según (Fayyad et al., 1996a), *"KDD es el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia comprensibles a partir de los datos"*.

La respuesta obtenida a través de este proceso es lo que da valor al almacenamiento de información y será de gran ayuda a la hora de tomar decisiones.

La Minería de Datos es una parte del proceso de Extracción de Conocimiento y reúne al conjunto de técnicas que permiten obtener los patrones mencionados previamente. Sin embargo, para extraer el conocimiento esperado es preciso analizar la información antes de aplicar una determinada técnica y luego de obtener los patrones es preciso analizarlos para determinar su utilidad y en caso de ser necesario, tal vez sea preciso revisar uno, dos o todos los pasos previos.

Fases del proceso de Extracción de Conocimiento

El proceso de *KDD*, está muy lejos de poder ser mecanizado. Comienza con el análisis del problema y la comprensión, por parte del encargado de obtener el nuevo conocimiento, de cual es la información con la que se va a trabajar y cual es el tipo de respuesta que se espera obtener. Esto permitirá determinar la importancia que los datos tienen en relación a las reglas de negocio y condicionará la forma en que serán tratados así como cuales serán las técnicas a emplear para procesarlos. No es cierto que una vez que se tienen los datos, puede aplicárseles directamente una técnica para llegar al resultado esperado. La realidad está muy lejos de esto y es precisamente allí donde el conocimiento del experto en Minería de Datos cobra valor.

Una vez que se ha entendido el problema y se conoce el significado de la información que se dispone, para aplicar el proceso de *KDD* deben realizarse las distintas etapas que lo componen tal como se encuentra ilustrado en la figura 2.1. Estas etapas si bien pueden ejecutarse en forma secuencial, habitualmente se realizan varias veces dependiendo de los resultados que se vayan obteniendo.

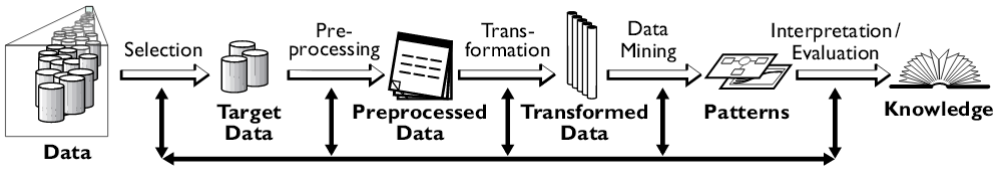


Figura 2.1: Etapas del Proceso de Extracción de Conocimiento (Fayyad et al., 1996a)

Tal como se dijo desde un principio, es posible que luego de recorrer todas las etapas los resultados obtenidos no sean los esperados y por lo tanto sea preciso revisar hacia atrás para ver cómo se llegó a esos resultados no deseados. Tal vez, la información no estuvo correctamente representada (etapa de limpieza y transformación) o la técnica elegida no fue la adecuada (etapa de minería de datos).

Es importante ver que uno de los problemas más grandes que se presenta al querer extraer conocimiento a partir de los datos es lograr comprender qué es lo que se quiere hacer y cómo debe representarse la información para poder lograrlo. Se trata de un proceso iterativo que comienza con la definición del problema a resolver y luego representa y modeliza la información disponible. A lo largo del proceso es común tener que revisar lo realizado siendo necesario rehacer una, algunas o todas las etapas hasta alcanzar la respuesta buscada. Puede encontrarse una descripción muy clara de las partes que componen el proceso de extracción de conocimiento en (Molina Felix, 2001). Incluso, en dicho trabajo se presentan ejemplos de aplicación concretos.

A continuación se hará una breve mención sobre cada parte de este famoso proceso de extracción de conocimiento con el objetivo de compartir algunas experiencias:

- **Recolección de datos**

  La realidad indica que muchos de los procesos de extracción de conocimiento que se han llevado a cabo han tenido que operar con información que no fue recolectada para ese fin. Esto se debe a que en el momento en que dicha información fue registrada no se conocía su utilidad ni la manera de procesarla. Eso llevó a que muchos registros no fueran normalizados provocando que un mismo proceso recolectado de distintas fuentes tuviera diferente representación. Tal fue el caso del registro de la información correspondiente al Programa de Sanidad Escolar (ProSanE) impulsado por el Ministerio de Salud de la Nación. El objetivo de ese programa era evaluar el estado de salud de niños y niñas en edad escolar, realizar el seguimiento de atención de los problemas de salud que se hubieran detectado e implementar acciones de promoción y prevención de la salud en las escuelas. Fueron varias las Universidades Nacionales que colaboraron con el Ministerio para analizar la información y en particular, la

autora de esta tesis realizó el análisis estadístico de la información correspondiente a varios establecimientos educativos de los Municipios de La Plata, Berisso y Ensenada. Los resultados obtenidos se encuentran detallados en (Lanzarini, 2011). En esa oportunidad el mayor desafío fue unificar la representación de la información ya que los distintos profesionales médicos que participaron en el registro utilizaron una aplicación informática de formato libre para cargar los datos. Como resultado, fue posible encontrar que algunos médicos indicaron la agudeza visual utilizando valores numéricos comprendidos en el rango 0 y 1 mientras que otros lo hacían con valores entre 0 y 10 e incluso en formato de texto del tipo "N/10" siendo N un número entre 0 y 10. Algo similar ocurría con la altura que en ocasiones se encontraba registrada en centímetros y otra en metros. Todas estas inconsistencias son propias de la falta de controles durante el registro y deben ser consideradas al momento de integrar la información.

- **Selección de Datos**

  Es importante llegar a reconocer cuál de toda la información registrada es la efectivamente relevante al problema que se busca resolver y cuál no. Por lo general, se trabaja con información estructurada en forma de tupla, como si fuera el resultado de una extensa vista de una base de datos con un elevado número de características o atributos (columnas) y un extenso número de ejemplos (filas). Ambos aspectos son contraproducentes para poder operar con los datos. Si se dispone de un elevado número de atributos es posible que el modelo a construir resulte extremadamente complejo y difícil de interpretar perjudicando su comprensión y su posterior aplicación en el proceso de toma de decisiones. En cuanto a la cantidad de ejemplos, lo importante no es que sean muchos sino que sean representativos de la situación a resolver por lo que identificar los casos adecuados permitirá reducir el tiempo de cálculo a la vez que simplificará la obtención de la respuesta buscada.

  En lo que se refiere a la selección de las características hay distintos enfoques.

  Cuando se trata de seleccionar atributos existen básicamente dos tipos de técnicas: filtros o métodos envolventes (en inglés wrappers). En el primer caso se busca calificar cada atributo por separado a través de una métrica para luego utilizar únicamente los que superen un umbral predefinido. En el segundo caso, se toman subconjuntos de atributos y se los utiliza en la solución del problema para luego medir el desempeño de dicha solución. Finalmente se elige el subconjunto con mejor resultado. En (Charnelli et al., 2015) puede verse un ejemplo de ambos enfoques. En esa oportunidad se utilizó un filtro basado en la métrica chi-cuadrado y un método envolvente basado en un árbol para analizar la información proveniente de una encuesta realizada por la Fundación Sadosky a varios alumnos de colegios secundarios acerca de si estudiarían

o no una carrera relacionada con informática. Como resultado se logró establecer que sólo el 10% de la información relevada era relevante a la hora de obtener un perfil general. Este resultado no es un tema menor porque no sólo reduce el tamaño de la información a almacenar sino que facilita la realización de futuras encuestas y agiliza la obtención de los perfiles buscados al reducir el tiempo de cálculo.

Un enfoque más sencillo suele ser el uso de técnicas de visualización. Por ejemplo en (Lanzarini et al., 2015a) se ejemplifica la manera de identificar, a través de técnicas de visualización, las características más relevantes en lo que se refiere al rendimiento académico de los alumnos de la Facultad de Informática de la Universidad Nacional de La Plata. Su aplicación a la información correspondiente a alumnos regulares y no regulares de la UNLP permitió establecer relaciones interesantes acerca del desempeño académico de los alumnos. Por ejemplo pudo verse que se produce un punto de inflexión en el segundo año y a partir de ese momento una gran cantidad de alumnos detienen su progreso en la carrera. También se identificaron los atributos más representativos para la construcción de un modelo de clasificación que permite describir y caracterizar a los alumnos según su condición de regularidad.

En cambio, en (Lanzarini et al., 2015b) se describe el proceso de identificación de las características más relevantes a través de técnicas de Minería de Datos. A diferencia del caso anterior, aquí se utilizaron medidas de correlación entre los atributos medidos y la respuesta esperada para poder identificar cuáles eran los más relevantes. Nuevamente en esta oportunidad se ha trabajado con la información del rendimiento académico de los alumnos de la Facultad de Informática de la Universidad Nacional de La Plata. A partir de los modelos obtenidos se puede afirmar que el hecho de que el alumno trabaje no implica que su rendimiento académico disminuya y que los alumnos jóvenes que demoran algunos años en ingresar a la Facultad presentan mejor rendimiento si manifiestan interés por conseguir un trabajo. Esto permite establecer una relación positiva entre el trabajo y el desempeño académico de los estudiantes contrariamente a lo que hace varias décadas se pensaba.

En cualquier caso, la identificación adecuada de los atributos o características a utilizar es un aspecto crucial a la hora de construir un modelo.

- **Limpieza y Transformación**

  A partir de esta etapa, el tipo de modelo a utilizar para resolver el problema planteado juega un papel fundamental.

  Las tareas de limpieza tienen que ver con la manera en que se completan los datos faltantes y se identifican los valores fuera de rango. Sin embargo, no todos los modelos se ven afectados por estos problemas. Por ejemplo, existen métodos de construcción de árboles de clasificación que pueden operar perfectamente utilizando

sólo la información disponible y partiendo el espacio de los ejemplos de entrada mediante una técnica supervisada.

La transformación de atributos, por su parte, involucra todas las modificaciones de deben efectuarse sobre los datos existentes ya sea para incrementar su capacidad descriptiva o por exigencia del modelo donde van a ser utilizados. Por ejemplo, las fechas específicas suelen ser poco relevantes a la hora de buscar relaciones generales. A menos que un día específico resulte un hito importante, suele ser aconsejable utilizar sólo el año o el mes o si se trata de un día hábil o no. Esto dependerá del problema a resolver.

En lo que se refiere a las restricciones impuestas por los modelos, las operaciones de numerización y discretización suelen ser las requeridas. Por ejemplo, si se desea resolver un problema de predicción utilizando una red neuronal, la información de entrada deberá ser numérica. En caso de disponer de información cualitativa, la misma deberá ser numerizada.

También en ocasiones se generan atributos nuevos ya sea para aportar mayor información o para simplificar el modelo final. Por ejemplo en (Lanzarini et al., 2015a), donde el objetivo era analizar el desempeño académico de los alumnos de la Facultad de Informática de la UNLP, se generó un nuevo atributo con la proporción de finales aprobados por cada alumno por año durante los primeros 5 años de la carrera. De esta forma se logró resumir la información registrada para cada uno de los exámenes finales rendidos por los alumnos en un atributo específico de su avance académico y a la vez se redujo el tiempo de obtención del modelo.

En todos los casos se tiene en mente el proceso a realizar y se actúa en consecuencia.

- **Construcción del Modelo**

  Aquí se encuentran todas las técnicas de la Minería de Datos cuyo resultado se expresa a través de la obtención del modelo buscado. La elección depende del tipo de problema a resolver.

  Cuando se busca un modelo predictivo, es decir, un modelo que ante un nuevo ejemplo permita dar una respuesta, las soluciones no lineales resultan muy convenientes. Tal es el caso de la redes neuronales. En cambio, si lo que se busca es explicar como está organizada la información disponible, es decir, se busca describir los datos con los que se está trabajando, las técnicas de agrupamiento suelen ser las elegidas.

  Sin embargo, existen soluciones que pueden cumplir ambas funciones permitiendo tener un panorama más amplio. Tal es el caso de las reglas de asociación y en especial las reglas de clasificación. Se hará una mención especial de este tipo de modelo en la sección siguiente.

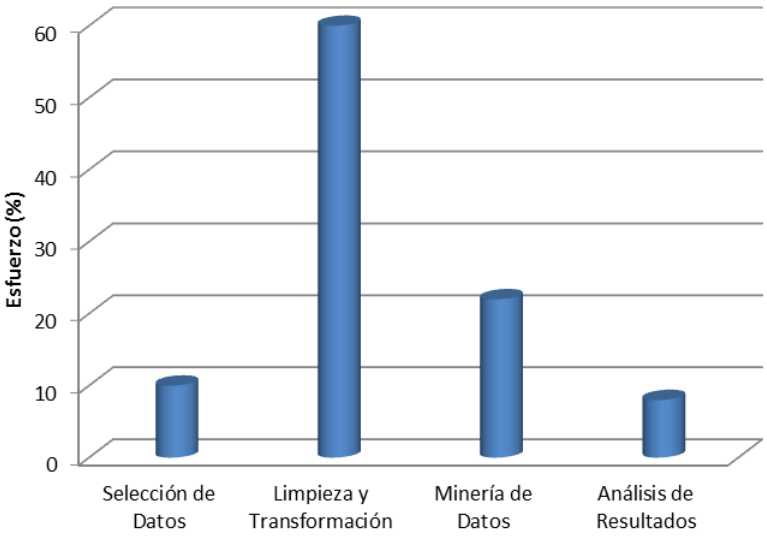


Figura 2.2: Esfuerzo requerido por cada una de las etapas del proceso de *KDD*

- **Interpretación y Evaluación de los resultados**

  Si se trata de un problema de predicción, La interpretación del modelo obtenido como resultado del paso anterior permitirá explicar las decisiones tomadas para dar una respuesta. En cambio, si se trata de un problema descriptivo resulta fundamental que sea sencillo de manera que su simple inspección permita identificar las relaciones más importantes presentes en los datos.

  Por lo general, hay una relación de compromiso entre la simplicidad y la precisión de un modelo de predicción.

La figura 2.2 muestra el esfuerzo que requiere cada una de las etapas mencionadas previamente. Según (López et al., 2013) el mayor trabajo se encuentra en el procesamiento previo que debe realizarse sobre los datos antes de comenzar a aplicar las técnicas de Minería de Datos. Allí reside gran parte del éxito de todo el proceso de *KDD*.

2.2 Reglas

Una regla es una proposición condicional. Su forma es

(2.1) *"SI (condición A) ENTONCES (resultado B)"*

siendo, por lo general, tanto A como B, conjunciones de expresiones de la forma (nom_atributo = valor$_i$) en caso de que el atributo sea cualitativo mientras que si es

cuantitativo debe indicar el rango a utilizar ya sea a través de un intervalo o a través una cota superior o inferior.

Las técnicas de extracción de reglas permiten obtener a partir de la información disponible un conjunto de este tipo de proposiciones dando así lugar a un modelo que puede tener uno de dos objetivos: establecer relaciones que describan los ejemplos existentes o clasificar los ejemplos a través del conjunto de reglas.

En el primer caso las reglas que se obtienen son reglas de asociación mientras que en el segundo caso las reglas serán de clasificación. Por ejemplo, si a partir de información referida al desempeño académico de ciertos alumnos se obtiene una regla de la forma:

```
Si (Resultado de la prueba A = DESAP) y (Asistencia = BAJA) ENTONCES

(Resultado de la prueba B = DESAP)
```

y para dicha regla se comprueba que la cardinalidad del conjunto de ejemplos que la verifica es representativa y la precisión de la regla es aceptable para el problema a resolver, debe considerarse que el resultado de la prueba A y la cantidad de clases a las que ha asistido el alumno condicionan el resultado de la prueba B.

Sin embargo, esta regla puede considerarse tanto como una regla de asociación como una de clasificación. En el primer caso el énfasis del conocimiento adquirido estará puesto en la incidencia del resultado de la prueba A y la asistencia del alumno en el resultado de la prueba B. Por otro lado también puede verse como una regla de clasificación ya que si se considera cierta, en caso de estar frente a un alumno que ha desaprobado la prueba A y que tiene una asistencia baja pero aún no ha rendido la prueba B puede predecirse el resultado de la evaluación faltante.

El ejemplo anterior permite apreciar las ventajas de contar con afirmaciones condicionales extraídas en forma automática las cuales pueden resumirse de la siguiente forma (Wang and Fu, 2006):

- Su interpretación es inmediata e incluyen una justificación de la proposición que expresan. Esto es un aspecto que no todos los modelos descriptivos poseen y es muy valorado por quienes deben utilizarlas convirtiéndolas en un modelo muy buscado.

- Permiten acotar el dominio de trabajo ya que ítems irrelevantes o redundantes tienden a estar ausentes en las reglas extraídas. A futuro, puede reducirse trabajo, costos e incluso espacio de almacenamiento de la información eliminando la información poco representativa. Se destaca nuevamente que ítems y atributos no son sinónimos siendo los primeros sumamente útiles para detectar valores de los atributos poco utilizados o poco representativos.

- A través de ciertas métricas es posible seleccionar las reglas adecuadas que permitan resumir correctamente la información disponible determinando las relaciones más representativas de una forma clara.

- Permiten identificar los pares de la forma (atributo, valor) más relevantes dentro de la información disponible. Muchos atributos pueden intervenir en la toma de decisiones. Sin embargo, algunos pueden tener una participación más activa en comparación con otros. No todos los modelos permiten establecer estas relaciones. Los procesos de extracción de reglas proporcionan una solución a este problema.

Reglas de Asociación

Una regla de asociación es una expresión como la mencionada en 2.1 donde la única restricción que posee es que las proposiciones atómicas que conforman el antecedente no se encuentren presentes en el consecuente. Nótese que no se trata sólo del uso del mismo atributo sino que lo que se pide que no coincida es (nom_atributo = valor_i) si el atributo es cualitativo o (nom_atributo $\in [limite_{inf}$, $\text{limite}_{sup}]$) si es cuantitativo. Esto implica que para disponer de los intervalos correspondientes a los valores numéricos debe realizarse previamente una tarea de identificación de los rangos adecuados lo cual redunda en la discretización de dichos atributos. Por tal motivo, cuando se busca obtener reglas de asociación se asume que se trabaja con variables cualitativas.

A estos pares de la forma (atributo, valor) o (atributo, intervalo) se los denomina*ítems*.

La obtención automática de reglas de asociación a través de técnicas de minería de datos es un problema aún no resuelto en el que se viene trabajando desde hace más de 20 años. Buscando dar una definición más formal de una regla de asociación, se utilizará la notación indicada en (Agrawal et al., 1993a), en reconocimiento a unos de los artículos más referenciados en la temática.

Sea $I = \{I_1, I_2, \ldots, I_n\}$ el conjunto completo de ítems presentes en la información disponible. Puede verse cada ejemplo del problema como una transacción T formada por algunos de estos ítems, es decir que, $T \subset I$. Sea D el conjunto completo de transacciones o ejemplos del problema a resolver. Una regla de asociación es una expresión de la forma $A \Rightarrow C$ con $A,\ C \subset I,\ A \cap C = \emptyset$, donde A y C son llamados antecedente y consecuente de la regla respectivamente.

Cuando se trata de extraer reglas de asociación, el problema de base es obtener los "conjuntos de ítems frecuentes". Se trata de conjuntos formados por pares de la forma (atributo,valor) cuyos elementos aparecen con una frecuencia mayor o igual a un umbral especificado de antemano; de allí que se los denomine "frecuentes". El conjunto I mencionado previamente está formado por todos los ítems.

Luego, con los elementos de cada conjunto de ítems frecuentes identificado, se forman las reglas de manera aleatoria y se miden con distintas métricas. Es un proceso computacionalmente costoso porque consiste en formar todas las combinaciones posibles para luego determinar si cumplen con las métricas de calidad indicadas a priori, de allí la importancia de seleccionar los ítems adecuados para realizar la construcción ya que de esa forma se garantiza que al menos la regla a obtener contará con un número mínimo de ejemplos que la respalde. De todas formas, eso no garantiza su calidad.

Hay varias alternativas para obtener reglas de asociación (Aggarwal, 2015). Un algoritmo "clásico" es el algoritmo "A Priori", presentado por primera vez en (Agrawal and Srikant, 1994) junto con el algoritmo "AprioriTID". Ambos algoritmos superaron con creces a los anteriormente conocidos, definidos en (Agrawal et al., 1993b) y (Houtsma and Swami, 1993) y se convirtieron en la base de muchos algoritmos posteriores así como un punto de referencia para los algoritmos actuales.

Los algoritmos de la familia Apriori fueron los primeros en mostrar algunas estrategias para identificar los conjuntos de ítems frecuentes a utilizar en la construcción de las reglas. Si bien se trata de algoritmos fáciles de implementar resultan costosos en tiempo ya que realizan múltiples pasadas sobre la base de datos. Como forma de resolver esto se desarrollaron variantes que reducen el tiempo de cómputo que se requiere para obtener los conjuntos de ítems frecuentes (Bayardo, 1998; Inokuchi et al., 2000; Wu et al., 2009) y actualmente sigue siendo motivo de investigación (Saravana Kumar and Manicka Chezian, 2012; Vasoya, 2014; Singh et al., 2015).

Las métricas más utilizadas para medir la calidad de una regla de asociación son el soporte y la confianza. Ambas se basan en el mismo concepto de soporte de un subconjunto de ítems. Dado un conjunto de ítems I, el soporte de un subconjunto de ítems $I_0 \subseteq I$ se define como la proporción de ejemplos de entrada que los contienen. Es decir que para cada transacción T de la base de datos D el soporte de I_0 se calcula de la siguiente forma:

$$supp(I_0) = \frac{|T \in D \mid I_0 \subseteq T|}{|D|}$$

siendo $|D|$ la cantidad total de ejemplos o cardinalidad de D. También puede decirse que el soporte de I_0 es la probabilidad de que aparezca en una transacción de D. Luego, el soporte de una regla de asociación $A \Rightarrow C$ en D es

$$Sop(A \Rightarrow C) = supp(A \cup C) \tag{2.2}$$

y su confianza es

$$Conf(A \Rightarrow C) = \frac{supp(A \cup C)}{supp(A)} = \frac{Sop(A \Rightarrow C)}{supp(A)} \tag{2.3}$$

El soporte es la proporción de casos en los que la regla se cumple. La confianza es la probabilidad condicional de C con respecto a A. Generalmente, se considerarán "recomendables" las reglas que superen ciertos umbrales de soporte y confianza predefinidos. Si una

regla cumple con un soporte mínimo garantiza su representación por un cierto número de casos mientras que si posee cierta confianza se tendrá alguna medida de su precisión.

La confianza de una regla es un valor entre 0 y 1 donde 1 indica que la regla es perfecta y 0 que no se cumple en ningún caso. Algunos autores observan que no cumple con todas las propiedades que se espera de una medida de precisión. Según (Piatetsky-Shapiro, 1991) cualquier medida de precisión ACC debería verificar las siguientes tres propiedades:

- $ACC(A \Rightarrow C) = 0$ cuando $Sop(A \Rightarrow C) = supp(A)supp(C)$. Esta propiedad indica que cualquier medida de precisión debería testear independencia.
- $ACC(A \Rightarrow C) = 0$ debería incrementarse monótonamente con $Sop(A \Rightarrow C)$ cuando el resto de los parámetros permanecen sin cambios.
- $ACC(A \Rightarrow C) = 0$ debería decrecer monótonamente junto con $supp(A)$ (o $supp(C)$) cuando el resto de los parámetros permanecen sin cambios.

Puede verse que la confianza sólo cumple con la segunda. Como solución a esto existen otras métricas de calidad de una regla que pueden consultarse en la literatura (Azevedo and Jorge, 2007).

Reglas de Clasificación

Las reglas de clasificación en cierto sentido pueden considerarse un caso particular de las reglas de asociación ya que son proposiciones condicionales con intersección vacía entre las condiciones del antecedente y el consecuente. La diferencia entre estos dos tipos de reglas radica en que cuando se trata de reglas de clasificación, todas las reglas que componen el modelo tienen un único atributo en su consecuente, el mismo para todas las reglas, el que contiene la etiqueta de la clase.

También se diferencian en la manera en que se obtienen ya que utilizan una técnica supervisada a partir de ejemplos etiquetados. Es preciso conocer a qué clase corresponden los ejemplos de muestra para poder construir el modelo.

Los métodos de obtención de reglas de clasificación pueden dividirse en dos categorías:

- Directos: son aquellos capaces de extraer las reglas directamente a partir de los datos. A esta categoría corresponden los métodos tales como ZeroR, OneR, PRISM, entre otros (Witten et al., 2011).
- Indirectos: construyen un modelo con la información disponible y luego a partir de él extraen las reglas. Existen métodos capaces de obtener conjuntos de reglas a partir de árboles o redes neuronales. En esta categoría se ubica el método propuesto en esta tesis. Algunos trabajos relacionados se mencionan a continuación.

2.3 Trabajos Relacionados

Esta sección hace un breve recorrido por los métodos habitualmente utilizados cuando se busca extraer reglas de clasificación. Algunos de ellos han sido usados para medir el desempeño del método propuesto en esta tesis.

2.3.1 Arboles

La construcción de árboles de clasificación se basa en algoritmos del estilo "divide y vencerás". Utiliza un aprendizaje supervisado para identificar la manera de separar recursivamente los ejemplos hasta lograr subconjuntos suficientemente homogéneos. Existen distintos criterios de división la mayoría de ellos basados en medidas de entropía siendo la tasa de ganancia o *gain ratio* una de las de mejor desempeño. El método C4.5 definido en (Quinlan, 1993) es sin duda el más utilizado ya que puede operar con atributos cualitativos y cuantitativos. También incluye un criterio de poda que permite simplificar el árbol según el nivel de precisión indicado. La mejora de este método, denominada C5.0, ha sido definida por el mismo autor y analizada en varias oportunidades (Quinlan, 2015). Puede verse en (RR, 2012) los resultados de su aplicación en tres bases de datos. También (Pandya and Pandya, 2015) realiza una análisis comparativo de ambas versiones.

Existen métodos de obtención de reglas de clasificación basados en árboles que van desde la mera simplicación de sus ramas (Ding An et al., 2001) hasta otros que construyen árboles de manera parcial debido al alto costo que tiene el armado de la estructura completa. Este último es el caso del método PART definido en (Frank and Witten, 1998b) que utiliza árboles que comienzan a construirse de misma forma que lo hace C4.5 pero controlando el error que se produce con cada nueva división. La construcción se detiene usando como criterio el error que se produce al insertar un nuevo atributo en una rama. Para ello mide el error cometido en cada paso a través del límite superior del intervalo de confianza para el error promedio que puede construir a partir de los ejemplos antes y después de considerar un atributo. C4.5 usa algo similar para podar el árbol. A partir de la estructura formada de manera parcial selecciona la rama que dará lugar a la mejor regla obtenida para el conjunto de ejemplos de partida. Luego para poder obtener las siguientes, suprime los ejemplos correctamente cubiertos por la regla hallada y repite el proceso hasta lograr una cobertura adecuada. De cada subárbol elige la rama con mayor tasa de acierto y la utiliza como regla. Como resultado obtiene una lista de reglas de clasificación en lugar de un conjunto de reglas. Es decir que las reglas obtenidas deben ser evaluadas en orden, utilizando la primera que cumpla con el ejemplo a clasificar. Este es uno de los métodos que se utiliza como comparación del método propuesto en esta tesis ya que la forma de construcción de las reglas es similar.

El método PART por basarse en la construcción parcial de un árbol analiza los atributos en forma independiente a la hora de agregarlos a la estructura. Otra forma de construir cada regla es trabajar sobre el antecedente completo y hacer modificaciones sobre los atributos que lo componen en forma simultánea. De esta manera trabaja el método propuesto en esta tesis.

2.3.2 Redes Neuronales

Cuando el objetivo es obtener un modelo basado en reglas trabajando sobre todos los atributos al mismo tiempo, las técnicas de agrupamiento resultan sumamente útiles. Estas técnicas operan sin supervisión y dan como resultado grupos de ejemplos con características similares. Resultan una herramienta muy útil tanto para detectar los patrones interesantes dentro del conjunto como para determinar relaciones entre los atributos que los definen. Por este motivo, resultan de especial interés aquellas estrategias que basan su funcionamiento en una medida de parecido o similitud.

Por ejemplo en (Lanzarini et al., 2011b) se presentó un método que permite identificar los conjuntos de items frecuentes a partir de una red neuronal SOM difusa (ver anexo A). En ese artículo, el objetivo fue identificar las relaciones más relevantes entre los términos presentes en 2995 mails correspondiente al Proyecto de Tutorías realizadas en el marco del Proyecto PACENI durante el período abril a noviembre de 2009. PACENI es la denominación adoptada por el "Proyecto de Apoyo para el mejoramiento de la enseñanza en el primer año de las carreras de grado de Ciencias Exactas y Naturales, Ciencias Económicas e Informática" y es uno de los elementos que componen el Programa de Calidad Universitaria y que, a su vez, integra una de las facetas a las que se dedica la Secretaría de Políticas Universitarias, ámbito dependiente del Ministerio de Educación. Es importante tener presente que, la extracción de información a partir de e-mails requiere de algunas consideraciones especiales por tratarse, en general, de textos cortos con una redacción bastante abreviada. En estas condiciones, métricas tales como la longitud del texto o la frecuencia con la que una palabra aparece dentro de él pierden relevancia. Para el procesamiento se utilizó un diccionario construido automáticamente a partir de la reducción de cada palabra a su raíz (stemming) y su posterior selección. Utilizando el diccionario se representó cada mail como un vector numérico y se los utilizó para entrenar una red neuronal FSOM (Fuzzy Self Organizing Map). A partir de los pesos de cada neurona de la red entrenada y teniendo en cuenta los grados de pertenencia de los ejemplos correspondientes, se identificaron las combinaciones de términos más habituales. Finalmente, utilizando métricas de reglas de asociación se estableció la relevancia de cada combinación.

Las redes neuronales competitivas han sido utilizadas en reiteradas ocasiones como

punto de partida para obtener reglas de clasificación (Pateritsas et al., 2007; Chihli and Huang, 2010). Sin embargo el resultado del entrenamiento varía en función del tamaño de la red y los valores de los pesos iniciales. Por lo tanto, cuando no se dispone a priori de un conocimiento adecuado del espacio de los datos de entrada, resulta complejo establecer una medida adecuada que lleve al resultado óptimo. En tales casos, es preciso contar con mecanismos capaces de comparar distintos agrupamientos con el objeto de establecer similitudes y diferencias.

Una herramienta muy utilizada para resolver problemas de clustering es la técnica conocida como acumulación de evidencia (Fred and Jain, 2005). Su objetivo principal es comparar el resultado producido por distintas estrategias de agrupamiento a través de una matriz de co-asociación que permite medir el grado de similitud entre los patrones de entrada. De esta manera, es posible aplicar una misma estrategia de agrupamiento con distintos parámetros y comparar los resultados obtenidos a fin de establecer de una manera óptima la ubicación de los grupos más homogéneos.

Existen varios estudios sobre la acumulación de evidencia y todos coinciden en lo difícil que resulta extraer los clusters cuando no se encuentran debidamente separados (Fred and Jain, 2005; Vega Pons and Ruiz Schulcloper, 2011). Si bien los resultados obtenidos utilizando “cluster ensembles” son robustos y confiables, queda la tarea de establecer una representación para el conocimiento obtenido.

En esta línea, en (Hasperué and Lanzarini, 2007) se propuso una estrategia para obtener reglas de clasificación a partir de una matriz de co-asociación de los datos de entrada cuyos valores surgen del entrenamiento de una red neuronal competitiva dinámica utilizando el método AVGSOM definido en (Hasperué and Lanzarini, 2005). El método de extracción de reglas propuesto y utilizado en este artículo es una versión mejorada de (Hasperué and Lanzarini, 2006) ya que propone una forma no supervisada para establecer los agrupamientos iniciales a través de una matriz de co-asociación y modificó la separación de las reglas mejorando su precisión. Una variante de esta estrategia que permite obtener reglas difusas fue presentada en (Hasperué and Lanzarini, 2008). Sin embargo, ambas tienen la limitación de trabajar sólo sobre atributos numéricos.

2.3.3 Técnicas de optimización

Si se piensa en la regla como una forma adecuada de relacionar items, puede plantearse el proceso de extracción de un conjunto de reglas de clasificación como un problema de optimización. Desde esta perspectiva, el objetivo es seleccionar los items adecuados para formar las reglas en cuestión. La literatura muestra diversas soluciones basadas en meta-heurísticas poblacionales tales como algoritmos genéticos (AG), optimización basada en cúmulo de partículas (PSO) y colonias de hormigas (ACO). Incluso hay soluciones que las

combinan.

El funcionamiento de todas estas técnicas se basa en la mejora sucesiva de los elementos de la población. La representación más utilizada es la que relaciona a cada uno de estos elementos con una regla específica y dado que la población evoluciona en busca de un único objetivo la pérdida de diversidad es inevitable. Por este motivo, la generación de una lista de reglas es el resultado habitual. Es decir que no se llegará a un conjunto de reglas independientes sino que obtendrá una lista que deberá ser inspeccionada en un orden dado para poder ser aplicada. Esta también es una característica de la que no está exento el método propuesto y que será descripta con más detalle en el capítulo 4.

En lo que se refiere a la manera en que se realiza la búsqueda, debe observarse que AG y PSO tienen buen desempeño cuando se trabaja con atributos numéricos mientras que ACO resuelve mejor las situaciones donde aparecen atributos cualitativos. Para operar con los dos tipos de atributos todos estos métodos deben buscar la manera de reducir la brecha.

Revisando la literatura se encuentra que aunque hay muchos artículos publicados, los resultados obtenidos dejan ver que la extracción automática de reglas de clasificación utilizando técnicas de optimización es un tema aún no resuelto. En (Al-Magaleh and Shahbazkia, 2012) se propone el uso de un AG convencional para extraer reglas pero se publican los resultados sobre cuatro bases del repositorio UCI y se lo compara con C4.5 y DTGA, un método de dos pasadas que primero construye un árbol con C4.5 y luego usa AG para obtener reglas eficientes a partir de él. Un enfoque similar ya había sido utilizado en (Fidelis et al., 2000) donde dentro de un cromosoma de longitud fija se evoluciona también la decisión de cuales serán los atributos que compondrán la regla. En todos los casos,la manera de operar con atributos nominales es a través de una numerización binaria. Esta representación incrementa rápidamente la dimensión del espacio de entrada haciendo más compleja la evaluación de la performance de las soluciones en evolución. Existen trabajos que proponen buscar representaciones alternativas como (De Jong and Spears, 1991) que sugiere descubrir conceptos en lugar de sólo combinar atributos para formar reglas. El enfoque es interesante pero no hay evidencia de resultados concretos y satisfactorios en esta dirección. En la mayoría de los casos la representación binaria sigue siendo la solución más popular.

En lo que se refiere a PSO existen distintas soluciones. Algunas son totalmente genéricas y proponen utilizar cúmulos de partículas, de tamaño fijo, inicializadas en forma aleatoria, que se desplazan utilizando el algoritmo convencional como (Wang et al., 2006) hasta otros que focalizan en la resolución de un problema particular como (Gandhi et al., 2010) extrayendo reglas de clasificación para resolver un problema concreto de datos de cáncer. En todos los casos las partículas contienen una única regla y se aplica algún criterio de nicho para no perder diversidad.

En (Holden and Freitas, 2008) se detalla la manera de combinar PSO y ACO para

extraer reglas de clasificación. En este trabajo la propuesta es combinar lo mejor de estas dos técnicas: la capacidad de operar con atributos numéricos de PSO con la habilidad de manejar a través del valor de feromona la probabilidad de incluir un atributo nominal en la construcción del antecedente. Luego de obtener el conjunto de reglas se aplica una técnica de simplificación para reducir su longitud.

Una de las mejores soluciones halladas y que muestra resultados similares al método propuesto en esta tesis, se denomina cAnt-MinerPB y su descripción fue presentada en (Medland et al., 2012). Se trata de una de las últimas extensiones del método Ant-Miner definido en (Parpinelli et al., 2002). En Ant-Miner la búsqueda se optimiza para encontrar la mejor regla individual en cada paso de la cobertura secuencial y opera sobre atributos nominales. Luego en (Liu et al., 2002) se presenta una heurística basada en densidad cuyo cálculo tiene un bajo costo computacional y en (Liu et al., 2003) se introduce una estrategia diferente para la actualización de los niveles de feromonas y se cambia la regla de transición de estados. En comparación con el trabajo de (Parpinelli et al., 2002) esta versión promete mejorar la exactitud de las listas de reglas. La incorporación de atributos numéricos fue definida en (Otero et al., 2008) a través del método cAntMiner que propuso utilizar el concepto de entropía para discretizarlos.

En cAnt-MinerPB descripto en (Medland et al., 2012) la mejora se realiza de dos maneras: buscando dinámicamente la función de calidad de la regla que se utiliza mientras se están podando las reglas y mejorando la función de calidad de lista de reglas que guía la búsqueda. Los autores afirman que el cambio de la función de calidad de la regla tiene poco efecto sobre el rendimiento global, pero que al mejorar la función de calidad de la lista de reglas es posible influir positivamente en las listas de reglas descubiertas. Los resultados obtenidos son buenos aunque el tiempo de ejecución no escala adecuadamente en relación al crecimiento del juego de datos. Cuando se utilizó la versión disponible en (MYRA, 2011) para operar con datos reales los tiempos requeridos para realizar las pruebas resultaron prohibitivos. Sin embargo, el funcionamiento del algoritmo para datos de repositorio fueron satisfactorios. Este tema será retomado en el capítulo 4.

2.4 Conclusiones

Obtener conocimiento nuevo a partir de información almacenada no es una tarea simple.

En este capítulo se han explicado brevemente cada una de las etapas que conforman el proceso de extracción de conocimiento utilizando la definición dada en (Fayyad et al., 1996a) aunque no es la única. Haciendo una revisión más amplia de la literatura, se observa que existe un acuerdo general por un proceso que inicia con la obtención y preparación de la información, continua con la generación del modelo y finaliza con el análisis e interpretación

de los resultados. Por tal motivo, alcanza con la elección de las etapas indicadas.

Más allá del detalle dado a los distintos pasos del proceso, el valor de esta introducción radica en la mirada global que se propone de la tarea a realizar. Por lo general, quienes se inician en la Minería de Datos consideran al proceso como un secuencia de pasos que no deben volver a ser transitados hasta no haber llegado al final y comprobado que los resultados no son los esperados. El error en el procedimiento es no detectar tempranamente el tipo de problema a resolver y seleccionar el modelo adecuado para hacerlo. Esto implica un amplio conocimiento de las opciones existentes y del efecto que las transformaciones que se realizan en las primeras etapas tienen sobre su funcionamiento. Una vez construido el modelo las métricas de performance son más acotadas y las herramientas que ayudan a su interpretación dependerán de las preferencias del especialista y de quienes sean los encargados de la toma de decisiones. Mirar la tarea como un todo es fundamental para lograr el éxito.

De todos los modelos que pueden trabajarse dentro de la Minería de Datos, la segunda parte de este capítulo ha estado dedicada a explicar los conceptos básicos de las reglas de clasificación, las métricas generalmente utilizadas para medir su desempeño así como los métodos que habitualmente se utilizan para obtenerlas.

El foco estuvo puesto en los métodos C4.5, PART y cAntMiner por ser los utilizados para medir el desempeño del método propuesto el cual se describirá con detalle en los capítulos siguientes.

CAPITULO 3

OPTIMIZACIÓN MEDIANTE CÚMULO DE PARTÍCULAS

Cuando se debe resolver un problema se busca hacerlo de la mejor manera posible. Sin embargo, hallar la mejor solución posible existente o solución óptima no siempre es una tarea sencilla. Todo depende de la complejidad del espacio de búsqueda.

Si el problema no es complejo existen técnicas para hallar la solución, como la búsqueda del óptimo utilizando información del gradiente o a través de estrategias partitivas. A medida que la complejidad del espacio de búsqueda aumenta, el costo computacional de estas técnicas se incrementa notablemente haciendo que su aplicación sea inviable. En estos casos debe considerarse la posibilidad de buscar una solución subóptima aceptable en un tiempo razonable. Se trata de una aproximación al óptimo buscado que cumple con el umbral de error establecido.

Cuando la solución exacta es difícil de obtener, las estrategias de búsqueda aproximadas basadas en distintas heurísticas han demostrado ser sumamente efectivas dando lugar a distintas técnicas de optimización iterativas. Entre estas técnicas se distinguen las basadas en una única solución y las que utilizan grupos de soluciones para aproximar el óptimo.

En el primer caso, la estrategia habitual es perfeccionar sucesivamente la solución actual haciendo uso de cualquier mejora inmediata. El concepto de gradiente suele ser el empleado en estos casos. Sin embargo, al hacer esto, es posible que la solución quede atrapada en un óptimo local (Skiena, 2010). Esta estrategia basada en la información obtenida a partir del gradiente puede ser mejorada si se permite que la solución empeore durante un cierto tiempo con la expectativa de mejorar en futuras iteraciones (Kirkpatrick et al., 1983)(Glover, 1989). Para evitar llegar a situaciones sin retorno una opción es

recordar la mejor solución hallada hasta el momento permitiendo eventualmente retroceder para retomar un camino de búsqueda alternativo.

En el segundo caso, cuando la estrategia se basa en mejorar un conjunto de soluciones generalmente denominado población, existen alternativas basadas en distintos procesos biológicos. En biología, adaptación es un proceso realizado por un organismo que ha evolucionado durante un período con el objetivo de acomodarse a las condiciones de su entorno (Futuyma, 2006).

La naturaleza ejemplifica a diario el éxito de los procesos adaptativos. Los seres humanos los han imitado en numerosos contextos y disciplinas con el único objetivo de "mejorar" su desempeño a la hora de resolver un problema. Por lo tanto no es de extrañar que, explícita o implícitamente, se haya utilizado durante la construcción de modelos matemáticos complejos ciertos comportamientos biológicos y se haya hecho un uso abundante de metáforas y términos procedentes de la genética, etología, e incluso de la etnología o la psicología para justificar decisiones y/o acciones (Clerc, 2013).

La inteligencia de cúmulo basa la búsqueda de la mejor solución en la "inteligencia colectiva". Aquí el objetivo es utilizar el conocimiento adquirido por los individuos de la población para mejorar el desempeño del grupo. Según el enfoque utilizado, podrán utilizarse individuos existentes para generar nuevos como lo hacen los Algoritmos Genéticos definiendo políticas para controlar el tamaño de la población (Lanzarini et al., 2000) o permitir que los individuos se desplacen en el espacio de soluciones observando el comportamiento de su entorno y el suyo propio. Esto último corresponde a la "inteligencia de cúmulo" que se discute con detalle en este capítulo por tratarse de la técnica de optimización más utilizada a lo largo de esta tesis para alcanzar el objetivo originalmente propuesto. La optimización realizada de esta forma se denomina "optimización mediante cúmulo de partículas" o PSO por su nombre en inglés *Particle Swarm Optimization*.

Las secciones siguientes describen brevemente las dos técnicas más conocidas que permiten obtener una solución óptima aproximada a través de una población de tamaño fijo cuyos individuos tienen la capacidad de desplazarse por el espacio de soluciones recordando su mejor desempeño y observando su entorno: PSO continuo y PSO binario. Luego se presentan algunas alternativas originales que mejoran sus respectivos desempeños.

3.1 PSO Continuo

La optimización mediante cúmulo de partículas fue propuesta en (Kennedy and Eberhart, 1995). Al igual que otras técnicas de optimización su funcionamiento tiene una inspiración biológica imitando el comportamiento social de algunos grupos de animales como las aves y los peces. Estos grupos utilizan la inteligencia colectiva o *Inteligencia de Cúmulo* para

resolver distintas situaciones.

Este comportamiento no es privativo de los animales ya que los seres humanos también lo utilizan. Por ejemplo, si un grupo de personas se encuentra atrapado en un incendio tenderán a buscar una salida rápida. Quienes no puedan hallarla observarán el comportamiento de los demás y seguirán a la mayoría aun sin saber si la dirección elegida es la correcta. Como puede observarse, según este enfoque, el conocimiento actual de cada individuo se ve afectado por el entorno.

Para más detalles sobre la inteligencia de cúmulo o los aspectos filosóficos de PSO puede consultarse el libro de Kennedy y Eberhart (Kennedy and Eberhart, 2001).

En (Kennedy and Eberhart, 1995), el cúmulo es de tamaño fijo y se denomina *población*. Cada individuo, llamado *partícula*, es una solución del problema que permanentemente busca mejorarse a si mismo teniendo en cuenta tres factores:

- conocimiento actual (su capacidad para resolver el problema).
- conocimiento histórico o experiencias anteriores (su memoria o conocimiento cognitivo).
- conocimiento de los individuos situados en su vecindario (su conocimiento social).

Los dos primeros hacen referencia a los aspectos culturales de los individuos del cúmulo y el último a su capacidad de observar al grupo e imitar comportamiento brindándole un aspecto social.

Características

La definición original de PSO opera sobre un espacio de búsqueda n-dimensional continuo (Kennedy and Eberhart, 1995).

A diferencia de otras estrategias, en PSO cada individuo o partícula está siempre en continuo movimiento explorando el espacio de búsqueda y nunca muere.

Cada partícula está compuesta por tres vectores y dos valores de fitness:

- Vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ almacena la posición actual de la partícula en el espacio de búsqueda.
- Vector $pBest_i = (p_{i1}, p_{i2}, \ldots, p_{in})$ almacena la mejor solución encontrada por la partícula hasta el momento.
- Vector velocidad $V_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ almacena el gradiente (dirección) según el cual se moverá la partícula.

- El valor fitness $fitness\_x_i$ almacena el valor de aptitud de la solución actual (vector X_i).
- El valor fitness $fitness\_pBest_i$ almacena el valor de aptitud de la mejor solución local encontrada hasta el momento (vector $pBest_i$)

La población se inicializa generando las posiciones y las velocidades iniciales de las partículas aleatoriamente. Una vez que la población ha sido creada, los individuos comienzan a moverse por el espacio de búsqueda por medio de un proceso iterativo.

Con la nueva posición del individuo, se calcula y actualiza su fitness ($fitness\_x_i$). Además, si el nuevo fitness del individuo es el mejor encontrado hasta el momento, se actualizan los valores de la mejor posición $pBest_i$ y su valor de fitness correspondiente $fitness\_pBest_i$.

Como se explicó anteriormente, el vector velocidad es modificado tomando en cuenta su experiencia y su entorno, según la siguiente expresión:

$$V_{id}(t+1) = w.V_{id}(t) + \varphi_1.rand_1.(pBest_{id} - X_{id}(t)) + \varphi_2.rand_2.(G_{id} - X_{id}(t)) \tag{3.1}$$

donde

- w representa el factor inercia y decrece en forma lineal a medida que avanzan las iteraciones del algoritmo.
- Las constantes φ_1 y φ_2 son las encargadas de medir la importancia que se le da a los factores cognitivo y social.
- $rand_1$ y $rand_2$ son valores aleatorios pertenecientes al intervalo [0,1] que hacen que el movimiento no sea excesivamente determinístico.
- G_i representa la posición de la partícula con el mejor $pBest$ en el entorno de X_i ($lBest$ o $localbest$) o de todo el cúmulo ($gBest$ o $globalbest$).

Los valores de w, φ_1 y φ_2 son importantes para asegurar que el algoritmo converja. Una ventaja de esta técnica es que no requiere ninguna información de gradiente. Esto permite aplicarla a problemas donde la información de gradiente o bien no es accesible o bien es muy costosa de calcular. Para más detalles sobre la elección de estos valores consultar (Shi and Eberhart, 1998) y (Pedersen, 2010) y en especial (Clerc and Kennedy, 2002).

Finalmente, se actualiza la posición de la partícula de la siguiente forma:

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \tag{3.2}$$

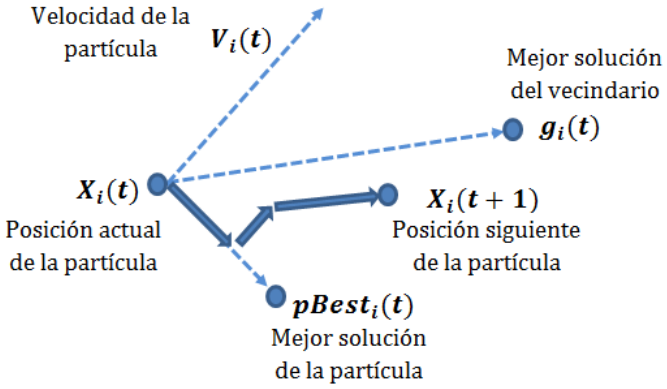


Figura 3.1: Movimiento de una partícula en el espacio de soluciones.

La figura 3.1 ilustra el movimiento de una partícula en el espacio de soluciones y el algoritmo 1 describe el pseudocódigo del proceso de búsqueda y optimización que realiza PSO.

Algoritmo 1: Algoritmo PSO Básico

$Pop = CrearPoblacion(N)$;
while *no se alcance la condición de terminación* **do**
  Ajustar el valor de w;
  for $i = 1$ **to** $size(Pop)$ **do**
    Evaluar Partícula X_i del cúmulo Pop;
    if $fitness(X_i)$ *es mejor que* $fitness(pBest_i)$ **then**
      $pBest_i \leftarrow X_i$;
      $fitness(pBest_i) \leftarrow fitness(X_i)$;
  for $i = 1$ **to** $size(Pop)$ **do**
    Elegir g_i de acuerdo al criterio de vecindario usado;
    $V_i \leftarrow w.V_i + (\varphi_1.rand_1.(pBest_i - X_i) + \varphi_2.rand_2.(g_i - X_i)$;
    $X_i \leftarrow X_i + V_i$;
Result: la mejor solución encontrada

Un aspecto deseable en todo proceso de búsqueda es dotar a los individuos de una buena capacidad exploratoria al inicio del proceso para luego intensificar la búsqueda en las áreas más prometedoras. Esto se logra con una gran inercia inicial, dada por el factor w que controla la proporción del vector velocidad que se utilizará para mover la partícula. El valor de este factor debe reducirse durante el proceso de búsqueda para permitir que el algoritmo converja. A medida que las partículas van perdiendo velocidad, la información referida a su mejor desempeño anterior y a la mejor solución del vecindario comenzarán a tomar importancia, provocando su estabilización. Con respecto al mejor global, la inercia aplicada,

por poca que esta sea, hará oscilar a la partícula cerca de la zona del óptimo hallado. Esta oscilación puede provocar pérdida en la precisión de la solución final.

Con la intención de corregir estos detalles dentro del funcionamiento del algoritmo se han propuesto, en el marco de esta tesis, algunas variantes originales de PSO, las cuales se detallan en la sección siguiente.

3.2 PSO Binario

En diversas aplicaciones prácticas cada vez es más frecuente la presencia de problemas de optimización que involucran variables que deben tomar valores discretos. Se trata de problemas de naturaleza combinatoria donde una de las formas de resolverlos es la discretización de valores continuos. Considerando que cualquier problema de optimización, discreto o continuo, puede ser expresado en notación binaria, Kennedy y Eberthart en (Kennedy and Eberhart, 1997), consideraron de utilidad el poder disponer de una versión binaria discreta de PSO para problemas de optimización discretos que denominaron *PSO Binario*.

En un espacio binario, una partícula se mueve en las esquinas de un hipercubo, cambiando los valores de los bits que determinan su posición. En este contexto, la velocidad de una particula puede ser vista como la cantidad de cambios ocurridos por iteración o la distancia de Hamming entre las posiciones de la partículas en el instante t y el $t+1$. Una partícula con 0 bits cambiados, de una iteración a la otra, no se mueve mientras que otra partícula que cambia por el valor contrario todos sus bits, se desplaza a velocidad máxima.

Repitiendo la notación de la sección anterior, la partícula de PSO binario está formada por

- Vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ almacena la posición actual de la partícula en el espacio de búsqueda. Es decir que se trata de un vector binario.

- Vector $pBest_i = (p_{i1}, p_{i2}, \ldots, p_{in})$ almacena la mejor solución encontrada por la partícula hasta el momento. Por lo tanto, también es binario.

- Vector velocidad $V_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ es un vector de valores reales.

- El valor fitness $fitness\_x_i$ almacena el valor de aptitud de la solución actual (vector x_i).

- El valor fitness $fitness\_pBest_i$ almacena el valor de aptitud de la mejor solución local encontrada hasta el momento (vector $pBest_i$)

El valor de la velocidad para la partícula i en la dimensión d se actualiza según (3.3)

$$v_{id}(t+1) = w.v_{id}(t) + \varphi_1.rand_1.(p_{id} - x_{id}(t)) + \varphi_2.rand_2.(g_{id} - x_{id}(t)) \tag{3.3}$$

Nótese que si bien las ecuaciones (3.3) y (3.1) coinciden, ahora p_{id} y x_{id} son enteros en $\{0,1\}$. Luego, a diferencia de lo indicado en (3.2), la nueva posición de la partícula no se obtiene sumando el vector velocidad recientemente obtenido sino que se lo utiliza como argumento de la función sigmoide indicada en (3.4) a fin de obtener un vector binario.

$$sig(v_{id}(t)) = \frac{1}{1 + e^{-v_{id}(t)}} \tag{3.4}$$

Luego, el vector posición de la partícula se actualiza de la siguiente forma

$$x_{id}(t+1) = \begin{cases} 1 & if\ rand() < sig(v_{id}(t+1)) \\ 0 & if\ not \end{cases} \tag{3.5}$$

donde $rand()$ es un número aleatorio con distribución uniforme en [0,1] distinto para cada dimensión.

Por lo tanto, la función sigmoide indicada en (3.4) convierte al vector velocidad V_i en un vector formado por valores de probabilidad siendo $sig(v_{ij})$ la probabilidad de que el elemento x_{ij} del vector que contiene la posición actual, se convierta en 1.

El algoritmo 2 detalla el pseudocódigo correspondiente.

En la versión continua de PSO el valor de v_{id} también se encontraba acotado a un intervalo cuyos valores eran parámetros del algoritmo.

En el caso discreto, el valor de la velocidad es acotado según(3.6)

$$|v_{id}| < Vmax \tag{3.6}$$

siendo $Vmax$ un parámetro del algoritmo. Es importante notar que $Vmax$ debe tomar valores adecuados para permitir que las partículas sigan explorando mínimamente alrededor del óptimo. Por ejemplo, si $Vmax = 6$ las probabilidades de cambio $sig(v_{id})$ estarán limitadas a 0.9975 y 0.0025. Esto permite una reducida probabilidad de cambio. Pero si $Vmax$ tomara un valor extremo, por ejemplo 50, sería muy poco probable que una partícula cambie su posición luego de alcanzar un óptimo (que tal vez sea local).

Es importante remarcar que la incorporación de la función sigmoide cambia radicalmente la manera de utilizar el vector velocidad para actualizar la posición de la partícula. En PSO continuo, el vector velocidad toma valores mayores al inicio para facilitar la exploración del espacio de soluciones y al final se reduce para permitir que la partícula se estabilice.

Algoritmo 2: Algoritmo PSO Binario

```
Pop = CrearPoblacion(N);
while no se alcance la condición de terminación do
    for i = 1 to size(Pop) do
        Evaluar Partícula X_i del cúmulo Pop;
        if fitness(X_i) es mejor que fitness(pBest_i) then
            pBest_i ← X_i;
            fitness(pBest_i) ← fitness(X_i);
    for i = 1 to size(Pop) do
        Elegir g_i de acuerdo al criterio de vecindario usado;
        V_i ← w.V_i + (φ1.rand1.(pBest_i − X_i) + φ2.rand2.(g_i − X_i);
        for d = 1 to n do
            if rand() < sig(V_id) then
                X_id = 1
            else
                X_id = 0
```

Result: la mejor solución encontrada

En PSO binario ocurre precisamente todo lo contrario. Los valores extremos, al ser mapeados por la función sigmoide, producen valores de probabilidad similares, cercanos a 0 o a 1, reduciendo la chance de cambio en los valores de la partícula. Por otro lado, valores del vector velocidad cercanos a cero incrementan la probabilidad de cambio. Es importante considerar que si la velocidad de una partícula es el vector nulo, cada uno de los dígitos binarios que determina su posición tiene probabilidad 0.5 de cambiar a 1. Es la situación más aleatoria que puede ocurrir.

Cada partícula incrementa su capacidad exploratoria a medida que el vector velocidad reduce su valor; es decir que , cuando v_{id} tiende a cero, $\lim_{t\to\infty} sig(v_{id}(t)) = 0.5$, permitiendo que cada dígito binario tome el valor 1 con probabilidad 0.5. Es decir que puede tomar cualquiera de los dos valores.
Por el contrario, cuando los valores del vector velocidad se incrementan, $\lim_{t\to\infty} sig(v_{id}(t)) = 1$, y por lo tanto todos los bits tienden a 1, mientras que cuando el vector velocidad decrece, tomando valores negativos, $\lim_{t\to\infty} sig(v_{id}(t)) = 0$ y todos los bits cambian a 0. Es importante remarcar que limitando los valores del vector velocidad entre -3 y 3, $sig(v_{id}) \in [0.0474, 0.9526]$, mientras que para valores superiores a 5, $sig(v_{id}) \simeq 1$ y para valores inferior a -5, $sig(v_{ij}) \simeq 0$.

3.2.1 Variante de PSO Binario

Si bien se han definido distintas alternativas al algoritmo PSO Binario descripto en la sección anterior, una variante interesante es la descripta en (Khanesar et al., 2007) por su buen desempeño y facilidad de implementación. Allí se cuestiona la dificultad de selección de parámetros del PSO Binario. En especial se menciona la importancia que el parámetro de inercia tiene en la capacidad exploratoria del método. Por este motivo, en (Khanesar et al., 2007) la velocidad de una partícula ya no incide en la probabilidad de cambiar a 1, sino en la probabilidad de cambiar desde su estado previo a su valor complementario.

En esta nueva definición, la velocidad de partícula como también sus parámetros tienen el mismo rol que en la versiones de PSO descriptas previamente. La mejor posición que visitó la partícula $PBest_i$ y la mejor posición global $gBest$ son actualizadas como en la versión continua o binaria de PSO.

Sin embargo, por cada partícula se introducen dos vectores: V_i^0 es la probabilidad que los bits de la partícula cambien a cero y V_i^1 es la probabilidad que los bits de la partícula cambien a uno. Dado que en la ecuación de actualización de estas velocidades el término de inercia es utilizado, estas velocidades no son complementarias. La probabilidad de cambio en el j-esimo bit de la i-esima partícula es definido como sigue:

$$v_{ij}^c = \begin{cases} v_{ij}^1 & if\ x_{ij} = 1 \\ v_{ij}^0 & if\ x_{ij} = 0 \end{cases} \tag{3.7}$$

La manera en que estos vectores son actualizados es la siguiente: Si el j-ésimo bit en la mejor posición global es 0 (cero) o si el j-ésimo bit en la mejor solución hallada por la partícula es cero, la velocidad v_{ij}^0 es incrementada y la probabilidad de cambiar a uno decrece. Por el contrario, si el j-ésimo bit en la mejor posición global es 1 o si el j-ésimo bit en la mejor solución hallada por la partícula es 1, v_{ij}^1 es incrementada y v_{ij}^0 decrece.

Usando este concepto, se extraen las siguientes reglas:

$$if\ PBest_{ij} = 1\ then\ d_{ij,1}^1 = c_1 r_1\ and\ d_{ij,1}^0 = -c_1 r_1$$
$$if\ PBest_{ij} = 0\ then\ d_{ij,1}^0 = c_1 r_1\ and\ d_{ij,1}^1 = -c_1 r_1$$
$$if\ gBest_j = 1\ then\ d_{ij,2}^1 = c_2 r_2\ and\ d_{ij,2}^0 = -c_2 r_2$$
$$if\ gBest_j = 0\ then\ d_{ij,2}^0 = c_2 r_2\ and\ d_{ij,2}^1 = -c_2 r_2$$

donde d_{ij}^1 y d_{ij}^0 son dos valores temporarios, r_1 y r_2 son dos valores aleatorios con distribución uniforme en (0,1) que se actualizan en cada iteración. Las constantes c_1 y c_2 son parámetros del algoritmo y se definen a priori.

Luego, los vectores velocidad se actualizan según (3.8) y (3.9)

$$v_{ij}^1 = wv_{ij}^1 + d_{ij,1}^1 + d_{ij,2}^1 \tag{3.8}$$

$$v_{ij}^0 = wv_{ij}^0 + d_{ij,1}^0 + d_{ij,2}^0 \tag{3.9}$$

donde w es un término de inercia. Desde este enfoque la dirección de cambio, hacia 1 o hacia 0, para cada bit, es considerada por separado.
Luego de actualizar la velocidad de las partículas, se obtiene la velocidad de cambio, como se indicó en (3.7).

Finalmente, para obtener la nueva posición de la partícula, se utiliza la función sigmoide definida en (3.4) y se calcula la nueva posición siguiendo lo indicado en (3.10)

$$x_{ij}(t+1) = \begin{cases} \overline{x_{ij}}(t) & if\ r_{ij} < sig(v_{ij}(t+1)) \\ x_{ij}(t) & if\ not \end{cases} \tag{3.10}$$

donde $\overline{x_{ij}}$ es el complemento a 2 de x_{ij}. Es decir, si $x_{ij} = 0$ entonces $\overline{x_{ij}} = 1$ y si $x_{ij} = 1$ entonces $\overline{x_{ij}} = 0$. De la misma forma que en el PSO Binario convencional, r_{ij} es un valor aleatorio con distribución uniforme entre 0 y 1.

3.3 PSO de población variable

El algoritmo descripto en la sección 3.1 trabaja sobre una población de tamaño fijo cuyo tamaño debe definirse a priori antes de comenzar el proceso de búsqueda. Esto condiciona la eficiencia y eficacia del algoritmo ya que si se utilizan pocos individuos quedarán zonas del espacio de búsqueda sin analizar y por el contrario, si son demasiados, el tiempo de convergencia se incrementará excesivamente.

Por tal motivo, en (Lanzarini et al., 2008) se ha propuesto una extensión original de PSO, denominada VarPSO, que incorpora los conceptos de edad y vecindad para permitir la variación del tamaño de la población. De esta forma, no es necesario definir a priori la cantidad de soluciones a utilizar, evitando así condicionar la calidad de la solución a obtener.

La variación del tamaño de la población se basa en una modificación del proceso adaptativo permitiendo el agregado y/o eliminación de individuos en función de su aptitud

para resolver el problema planteado. Esto se realiza principalmente a través del concepto de *tiempo de vida* que permite determinar el tiempo de permanencia de cada elemento dentro de la población. Además, dado que PSO tiende a poblar rápidamente las zonas exploradas con buen fitness, para no poblar excesivamente un mismo lugar del espacio de soluciones, se analiza el entorno de cada individuo y se eliminan las peores soluciones de las zonas muy pobladas. Ambos procesos se explican a continuación con mayor detalle:

3.3.1 Tiempo de vida

Uno de los conceptos más importantes de VarPSO es el tiempo de vida de una partícula ya que determina la duración de su permanencia dentro de la población. Dicho valor se expresa en cantidad de iteraciones, transcurridas las cuales, la partícula es eliminada. Este valor tiene una estrecha relación con la aptitud de la partícula y permite que los mejores individuos permanezcan en la población por mayor tiempo, influenciando el comportamiento del resto.

Para estimar el tiempo de vida de cada individuo de la población se los agrupó según su valor de aptitud en k grupos utilizando un método de clustering competitivo del tipo winner-take-all. Como resultado se obtuvo un conjunto de valores reales correspondientes a los centroides de cada agrupamiento. Por tratarse de valores numéricos, estos centroides pueden ser ordenados y representados como un vector de la forma $G = (g_1, g_2, \ldots, g_k)$. Luego, sobre el resultado de este agrupamiento puede aplicarse uno de los siguientes métodos:

a) **Asignación de tiempo de vida fijo por grupo**

Se divide el máximo tiempo de vida a asignar, MAX_LT, por la cantidad de grupos formados, k. Esto determina el rango que le corresponde a cada grupo. Luego, cada individuo recibirá un valor de tiempo de vida proporcional a su valor de aptitud dentro del grupo. Por ejemplo, Si el individuo X_i pertenece al grupo g_a su tiempo de vida, $TiempoDeVida\_x_i$, se calcula de la siguiente forma:

$AnchoClase := MAX\_LT/k$
$TVClasesAnt := (a-1) * AnchoClase$
$TVClaseActual := AnchoClase$
$MinFit = minimo\ fitness\ del\ grupo\ g_a$
$MaxFit = maximo\ fitness\ del\ grupo\ g_a$
$Desplazamiento = (fitness\_x_i - MinFit)/(MaxFit - MinFit)$
$TiempoDeVida\_x_i := trunc(TVClasesAnt + TVClaseActual * Desplazamiento)$
donde

- $AnchoClase$ es el rango del tiempo de vida asignado a cada grupo (dada por MAX_LT / Número de clases).

- a es el número del grupo al que pertenece el individuo x_i.
- $TVClaseActual$ es el rango de tiempo de vida del grupo al que pertenece el individuo.
- $TVClasesAnt$ es el rango de tiempo de vida asignado a los grupos anteriores a g_a
- $MinFit$ y $MaxFit$ son los valores de aptitud mínimo y máximo del grupo al que pertenece el individuo en consideración.
- $fitness\_x_i$ es el valor de aptitud del i-ésimo individuo de la población.

b) **Asignación de tiempo de vida proporcional a la cantidad de individuos de cada clase**

Cada clase recibe un rango de tiempo de vida proporcional a la cantidad de elementos que contiene. Es decir, que los individuos pertenecientes a las clases numerosas podrán tener un rango de tiempo de vida más amplio. El cálculo es el siguiente:
$TotalAnt := 0$
for m:=1 to a-1 do TotalAnt := TotalAnt + $Cantidad\_g_m$
$TVAnterior := MAX\_LT * TotalAnt/TotalIndiv$
$TVGrupoActual := MAX\_LT * Cantidad\_g_a/TotalIndiv$
$MinFit = minimo\ fitness\ del\ grupo\ g_a$
$Desplazamiento = (fitness\_x_i - MinFit)/(MaxFit - MinFit)$
$TiempoDeVida\_x_i := trunc(TVAnterior + TVGrupoActual * Desplazamiento)$
donde

- a es el número del grupo al que pertenece el individuo x_i cuyo tiempo de vida se desea calcular.
- $Cantidad\_g_i$ representa la cantidad total de individuos del grupo g_i.
- $TotalIndiv$ es la cantidad total de individuos de la población.
- $TotalAnt$ es la cantidad total de individuos pertenecientes a los grupos anteriores a g_a.
- $TVAnterior$ es el rango de tiempo de vida cubierto por los grupos anteriores a g_a.
- $TVGrupoActual$ es el rango de tiempo de vida cubierto por el grupo g_a.

Las figuras 3.2 y 3.3 ilustran los procesos antes descriptos.

Estas dos formas de calcular los tiempos de vida de los individuos deben combinarse a fin de lograr una asignación correcta. Se propone aplicar la asignación b) durante un cierto porcentaje de la cantidad de generaciones máximas del algoritmo y en las

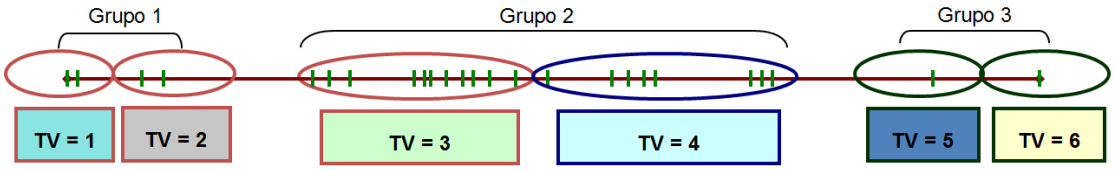


Figura 3.2: Asignación de tiempos de vida fijo por grupo utilizando tres agrupamientos ($k = 3$) y un tiempo de vida máximo de 6 iteraciones ($MAX\_LT = 6$)

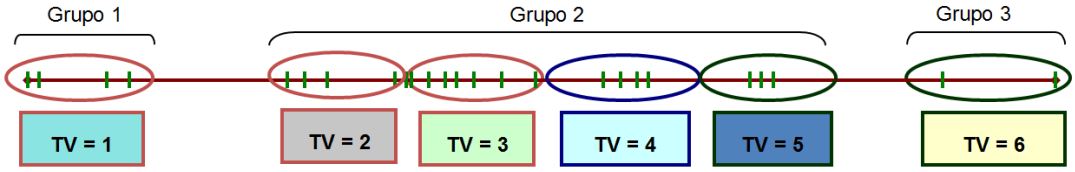


Figura 3.3: Asignación de tiempos de vida proporcional dentro de cada grupo utilizando tres agrupamientos ($k = 3$) y un tiempo de vida máximo de 6 iteraciones ($MAX\_LT = 6$)

restantes utilizar a). Esto se debe a que los agrupamientos iniciales se realizan sobre individuos que aun no están lo suficientemente adaptados y por consiguiente dan lugar a agrupamientos de tamaños muy disímiles. Si sobre estos agrupamientos se aplicara directamente la distribución indicada en a) muchos individuos recibirían tiempos de vida similares llevando al algoritmo a incrementar innecesariamente la cantidad de individuos de la población.

3.3.2 Inserción de partículas

La inserción de partículas tiene dos objetivos: incrementar la velocidad de convergencia incorporando individuos en las zonas menos pobladas y compensar la eliminación de partículas provocada por el cumplimiento de los respectivos tiempos de vida. Determinar los lugares convenientes, dentro del espacio de búsqueda, donde deben insertarse los nuevos individuos no es una tarea trivial. En realidad, se trata de una situación de compromiso entre la identificación de las zonas óptimas y la velocidad del proceso de inserción de los nuevos individuos.

La solución adoptada divide el problema original en dos partes: en primer lugar busca determinar cuantas partículas es necesario incorporar para luego establecer dónde deben posicionarse dentro del espacio de búsqueda.

La cantidad de partículas incorporadas en cada iteración coincide con la cantidad de individuos aislados. Se considera un individuo aislado a aquel que no posea ningún vecino dentro de un radio r preestablecido. Las ecuaciones 3.11 y 3.12 indican la forma de calcular dicho radio. Como puede observarse, r se calcula como el promedio de las distancias de

cada partícula con su vecino más cercano.

$$d_i = min\{\|x_i - x_j\|; \qquad \forall j \neq i \quad i = 1..n \tag{3.11}$$

$$r = \frac{\sum_{i=1}^{n} d_i}{n} \tag{3.12}$$

Sólo resta determinar dónde posicionar estos nuevos individuos. El criterio adoptado fue el siguiente: el 20% de estas nuevas partículas reciben el vector posición de los mejores individuos de la población pero su vector velocidad es aleatorio; el 80% restante se ubica en una posición aleatoria. De esta forma, una parte de los nuevos individuos comenzarán a moverse desde las posiciones que mejor desempeño han demostrado hasta el momento pero con dirección y velocidad distintas a las de los mejores individuos. El 80% restante permitirá llegar a explorar otras zonas del espacio de búsqueda. Es importante remarcar que, la eficacia de la medida de distancia utilizada en 3.11 dependerá de la representación del espacio de búsqueda seleccionada.

3.3.3 Algoritmo propuesto

El algoritmo comienza con una población de N individuos generados al azar dentro del espacio de búsqueda y calcula para cada uno de ellos su fitness y tiempo de vida correspondientes. Durante el proceso, los individuos se desplazan según las ecuaciones 3.2 y 3.1. La inercia utilizada para actualizar los vectores de velocidad es ajustada según 3.13

$$w = w_{start} - \frac{(w_{start} - w_{end})}{TotalIteraciones}.CurrentIteration \tag{3.13}$$

donde w_{start} es el valor inicial de w y w_{end} es el valor final.

El uso de un peso de inercia variable facilita la adaptación de la población. Un valor de w alto al comienzo de la evolución le permite a las partículas realizar movimientos grandes ubicándose en distintas posiciones del espacio de búsqueda. A medida que avanza el número de iteraciones, el valor de w se reduce permitiéndoles realizar un ajuste más fino.

A partir de las nuevas posiciones dentro del espacio de búsqueda, se recalcula el valor de fitness de los individuos y se obtiene el radio r según la ecuación 3.12.

Luego, se crean tantos individuos nuevos como partículas existan en la población sin vecinos dentro de este radio. Estos nuevos individuos tendrán vectores de velocidad aleatorios, dentro de los rangos permitidos. El 20% de estas nuevas partículas recibirán los

vectores de posición de los mejores individuos de la población y el 80% restante tendrán vectores de posición aleatorios. Para estas nuevas partículas, se evalúa su fitness y se las incorpora a la población. Utilizando la población completa se calcula el tiempo de vida de los recientemente incorporados.

Se decrementa en 1 el tiempo de vida de todos los individuos y aquellos que hayan alcanzado el valor cero son eliminados de la población.

El algoritmo propuesto utiliza elitismo por lo que el mejor individuo de cada iteración es preservado. De esta forma se garantiza que la población tendrá al menos una partícula. Esto se realiza reemplazando a la partícula con menor fitness por la mejor de la iteración anterior.

Finalmente, el algoritmo termina cuando se cumple una de las siguientes condiciones:

- Se alcanzó la cantidad máximas de iteraciones indicadas inicialmente.
- El mejor fitness no se ha modificado durante el 15% de las iteraciones totales.

El algoritmo 3 contiene el pseudocódigo del algoritmo descripto.

La función $CrearPoblacion$ recibe como parámetro la cantidad de partículas a crear y devuelve un cúmulo con vectores de posición y velocidad aleatorios dentro de los límites establecidos y con tiempos de vida nulos. Para estimarlos es preciso evaluar previamente el fitness de cada individuo.

El proceso $Calcular\_Tiempos\_de\_Vida$ recibe un cúmulo completo y sólo calcula el tiempo de vida correspondiente a las partículas que poseen tiempo de vida nulo al momento de hacer la invocación. No es posible aplicarlo únicamente al cúmulo $Nuevos$ porque el cálculo depende del agrupamiento de todos los individuos según su valor de fitness. El segundo parámetro corresponde al tipo de cálculo que debe realizar y vale 1 para la asinación descripta en 3.3.1.a) y 2 para la descripta en 3.3.1.b).

El cálculo del radio según la ecuación 3.12 se realiza dentro del proceso $CalcularRadios$ que recibe como parámetro el cúmulo completo y devuelve la cantidad de partículas nuevas que deben insertarse en la población. Este mismo módulo es el encargado de evitar la concentración de varias partículas en un mismo lugar del espacio de búsqueda, por tal motivo, también retorna la lista de individuos que tienen vecinos muy próximos. Dichas partículas son eliminadas en el módulo $VerEntorno$ en función de su fitness y la cantidad de hijos generados.

3.3.4 Resultados obtenidos

VarPSO fue utilizado para obtener el valor mínimo de distintas funciones. Por lo tanto, cada partícula contiene en su vector posición los valores de los argumentos de la función.

Algoritmo 3: Algoritmo PSO de población variable

$Pop = CrearPoblacion(N)$;
$Calcular\_Tiempos\_de\_Vida(Pop, 2)$;
$w \leftarrow INERCIA_MAXIMA$;
while *no se alcance la condición de terminación* **do**
    Guardar el individuo con mayor fitness;
    for $i = 1$ **to** $size(Pop)$ **do**
        Evaluar Partícula X_i del cúmulo Pop;
        if $fitness(X_i)$ *es mejor que* $fitness(pBest_i)$ **then**
            $pBest_i \leftarrow X_i$;
            $fitness(pBest_i) \leftarrow fitness(X_i)$;
    for $i = 1$ **to** $size(Pop)$ **do**
        Elegir g_i de acuerdo al criterio de vecindario usado;
        $V_i \leftarrow w.V_i + (\varphi_1.rand_1.(pBest_i - X_i) + \varphi_2.rand_2.(g_i - X_i)$;
        $X_i \leftarrow X_i + V_i$;
    $Calcular\_Fitness(Pop)$;
    $CalcularRadios(Pop, CantHijos, sentenciados)$;
    $Nuevos = CrearPoblacion(CantNuevos)$;
    Asignar al 20% de estos nuevos individuos los vectores de posición de los mejores individuos de Pop;
    $Calcular\_Fitness(Nuevos)$;
    $VerEntorno(Pop, sentenciados, CantHijos)$;
    $Pop = Pop \cup Nuevos$;
    $Calcular\_Tiempos\_de\_Vida(Pop)$;
    if $IteracionActual >$ *al 5% de las* $ITERACIONES\_TOTALES$ **then**
        $Calcular\_Tiempos\_de\_Vida(Pop, 1)$
    else
        $Calcular\_Tiempos\_de\_Vida(Pop, 2)$
    Restar 1 al tiempo de vida de cada partícula;
    Eliminar las partículas con tiempo de vida nulo;
    Reemplazar el peor individuo por el guardado al inicio de esta iteración;
    $w \leftarrow$ modificar dinámicamente la inercia;
Result: la mejor solución encontrada

La aptitud de cada partícula se calcula de la siguiente forma:

$$(c\_max - Valor\_de\_la\_Particula) \tag{3.14}$$

donde c_max representa una cota superior de la función en el intervalo a optimizar y $Valor\_de\_la\_Particula$ es el resultado de evaluar la función en el vector posición de la partícula correspondiente.

A continuación se detallan las funciones utilizadas. Para cada una de ellas se indica el intervalo utilizado para determinar el espacio de búsqueda y el valor de c_max empleado para calcular el fitness.

$F1(x,y) = x^2 + y^2 \qquad x,y \in [-1,5]; c\_max = 50$

$F2(x) = -x * sin(10 * \pi * x) + 1 \qquad x \in [-2,1]; c\_max = 3$

$$F3(x,y) = 0.5 + \frac{(sin(\sqrt{x^2+y^2+4}))^2 - 0.5}{(1+0.001.(x^2+y^2))^2} \qquad x,y \in [-50,50]; c\_max = 1$$

$$F4(x_1,x_2) = \frac{1}{0.002 + \sum_{j=1}^{25} \frac{1}{50j + \sum_{i=1}^{2}(x_i - a_{ij})^6}} \qquad x_1,x_2 \in [-50,50]; c\_max = 500$$

$$F5(x_1,x_2) = \frac{1}{0.002 + \frac{1}{1+(x_1+1)^{12}+(y_1+1)^{12}}}) \qquad x_1,x_2 \in [-200,200]; c\_max = 500$$

$$F6(x_1,x_2) = \frac{1}{0.002 + \sum_{j=1}^{3} \frac{1}{50j^2 + \sum_{i=1}^{2}(x_i - b_{ij})^{12}}} \qquad x_1,x_2 \in [-50,50]; c\_max = 500$$

$$b_{ij} = \begin{pmatrix} -30 & 16 & 30 \\ -40 & -32 & 35 \end{pmatrix}$$

Las seis funciones presentan un único mínimo. Cada una de ellas busca medir un aspecto diferente del algoritmo propuesto. F1 permite analizar la precisión de la solución obtenida. F2, F3 y F4 muestran su capacidad para moverse sobre un espacio de búsqueda con valores de fitness muy cambiantes. Nótese que la función F4 es una modificación de la función 5 de De Jong. Las funciones F5 y F6 fueron introducidas para analizar la capacidad exploratoria del algoritmo. En F5 aparece un único hueco que lleva al mínimo dentro de una superficie totalmente plana. F6 es similar pero presenta tres huecos con profundidades muy distintas.

Para cada función se realizaron 100 pruebas utilizando en cada caso una cantidad máxima de 500 iteraciones. Los valores de aprendizaje cognitivo y social utilizados, φ_1 y φ_2 descriptos en la ecuación 3.1, fueron ambos en 0.5. Los valores de inercia entre 0.2 y 1.5. El rango de velocidades permitidas fue establecido entre -0.5 y 0.5. La cantidad de clases

utilizadas para calcular el tiempo de vida de los individuos fue 4. Se realizaron pruebas con valores entre 2 y 10 confirmando que un valor alto para el número de clases, mejora la distinción del fitness entre los individuos pero incrementa sensiblemente el tamaño de la población. Por el otro lado, si la cantidad de clases es muy baja, el tamaño del cúmulo puede disminuir considerablemente. Un valor de 4 clases resulta adecuado para la minimización de las funciones antes indicadas. El valor utilizado como tiempo de vida máximo fue de 9 iteraciones salvo en las funciones F5 y F6 que se utilizó un valor de 12. La tabla 3.1 permite comparar los resultados obtenidos al aplicar el algoritmo propuesto en (Lanzarini et al., 2008) con el algoritmo PSO de población fija.

Se realizaron pruebas con tamaño de población inicial 5, 10, 20, 30, 40, 50, 60 y 70 partículas. En todos los casos, los valores corresponden a los promedios de las 100 pruebas realizadas para cada función tomando el tamaño de población para la cual se obtuvo el mejor fitness máximo promedio.

Como puede observarse, en todos los casos el método propuesto es superior o igual a la solución basada en población fija presentando una mayor diversidad de población. Además, salvo la función F1 para la cual es simple llegar al fitness máximo, el método propuesto utiliza una población inicial inferior al método de población fija. Si se analiza la cantidad de partículas desplazadas en promedio en cada caso puede verse que, en la mayoría de las funciones, el método propuesto realiza menos de la mitad de trabajo que las soluciones con población fija. Esto último no se verifica para las funciones F1 y F6. En el caso de F1, se debe a la simplicidad de la función que contrasta con el comportamiento explorador del método propuesto y en el caso de F6, se debe a que el método no converge prematuramente como su par de población fija sino que continua analizando el espacio de búsqueda obteniendo mejores soluciones.

La figura 3.4 muestra la capacidad del método propuesto para adaptarse a la superficie de búsqueda hallando una solución casi óptima independiente del tamaño de la población inicial. También puede observarse que los métodos de población fija ofrecen resultados correctos cuando se utiliza la población inicial adecuada. Cada punto de la figura 3.4 corresponde al fitness máximo promedio de las 600 pruebas (100 para cada función) realizadas para cada tamaño de población inicial. En cada caso, el fitness ha sido escalado linealmente a [0,1] dividiéndolo por el fitness máximo correspondiente a la función.

Si se analiza la cantidad de iteraciones promedio realizada por cada método en función del tamaño de la población inicial puede observarse que para las soluciones de población fija tiende a incrementarse levemente mientras que para las de población variable decrece rápidamente a medida que la población inicial aumenta. Esto se encuentra representado en la Figura 3.5 y refleja el comportamiento de cada método. Los de población fija dependen sólo del desplazamiento de las partículas iniciales mientas que las versiones de población variable realizan un aumento inicial de la población que les permite explorar rápidamente un

Function F1	Avg.Ite.	Min. Fit.	Med. Fit.	Max. Fit.	Ini.Pop	Fin.Pop
gBest PSO	106.90	48.0544	49.2226	50.0000	20	20.0
lBest PSO	142.08	48.0373	49.2735	50.0000	20	20.0
gBestVarPSO	181.89	30.4806	45.3319	49.9996	30	33.8
lBestVarPSO	169.82	30.3254	45.0502	49.9995	30	32.8
Function F2	Avg.Ite.	Min. Fit.	Med. Fit.	Max. Fit.	Ini.Pop	Fin.Pop
gBest PSO	102.67	1.6791	2.1150	3.8466	60	60.0
lBest PSO	102.40	1.7325	2.0647	3.8456	70	70.0
gBestVarPSO	115.22	1.3443	2.3861	3.8489	40	13.1
lBestVarPSO	122.18	1.2349	2.3705	3.8489	30	17.7
Function F3	Avg.Ite.	Min. Fit.	Med. Fit.	Max. Fit.	Ini.Pop	Fin.Pop
gBest PSO	231.06	0.7288	0.9074	0.9762	70	70.0
lBest PSO	197.59	0.6130	0.8607	0.9717	60	60.0
gBestVarPSO	175.17	0.3032	0.5857	0.9942	20	53.3
lBestVarPSO	154.88	0.3025	0.5866	0.9936	40	62.6
Function F4	Avg.Ite.	Min. Fit.	Med. Fit.	Max. Fit.	Ini.Pop	Fin.Pop
gBest PSO	284.10	441.7729	445.3109	446.6185	60	60.0
lBest PSO	218.77	437.8757	443.5489	444.9094	70	70.0
gBestVarPSO	132.04	145.0876	285.1007	453.0275	30	30.7
lBestVarPSO	124.64	156.9515	299.9142	454.5333	40	55.0
Function F5	Avg.Ite.	Min. Fit.	Med. Fit.	Max. Fit.	Ini.Pop	Fin.Pop
gBest PSO	140.19	468.3641	482.4536	484.0312	70	70.0
lBest PSO	138.37	448.4032	477.8286	479.0419	70	70.0
gBestVarPSO	117.01	171.4216	280.7481	494.0287	40	16.2
lBestVarPSO	121.72	134.1485	238.5021	496.0285	30	23.8
Function F6	Avg.Ite.	Min. Fit.	Med. Fit.	Max. Fit.	Ini.Pop	Fin.Pop
gBest PSO	103.27	373.4571	391.0167	391.8489	60	60.0
lBest PSO	115.49	404.7819	440.9218	442.8571	50	50.0
gBestVarPSO	92.56	71.9364	194.3381	443.2076	40	85.7
lBestVarPSO	108.44	120.5613	228.8209	443.4110	20	52.9

Tabla 3.1: Resultados obtenidos con PSO y VarPSO.

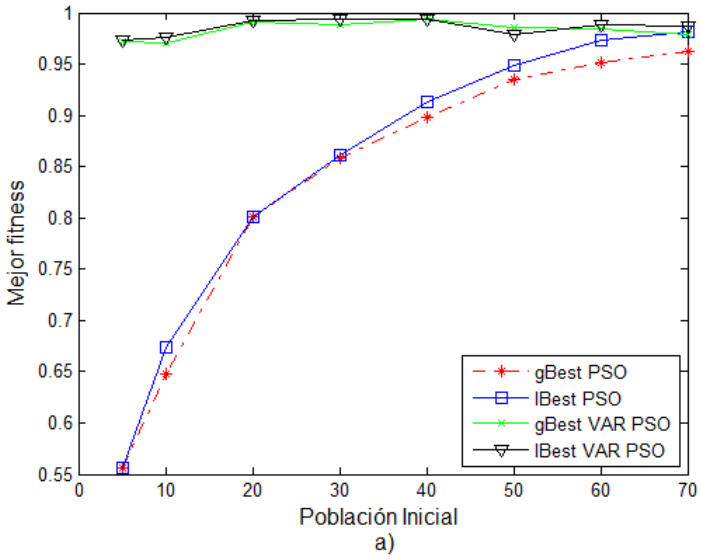


Figura 3.4: Fitness promedio máximo obtenido para distintos valores de población inicial

área más amplia del espacio de búsqueda llegando al óptimo con un número de iteraciones menor.

Finalmente, las Figuras 3.6 y 3.7 muestran el crecimiento promedio de la población para las dos variantes del método propuesto. Como puede observarse, el comportamiento es muy similar en los dos casos y tiene una fase de crecimiento, dentro de las primeras 30 iteraciones, seguida de una fase de reducción y estabilización.

3.4 PSO Binario con control de velocidad

Tanto el método PSO Binario original como la variante descripta en la sección anterior dejan en evidencia la importancia que tiene una adecuada modificación del vector velocidad. En el caso del PSO Binario original, el problema central se encuentra en el escaso control que se realiza a la hora de acotarlo. Si el vector velocidad toma valores excesivos, el aplicarle la función sigmoide hace que la probabilidad de cambio sea casi nula. Como forma de compensar este efecto, el método definido en (Khanesar et al., 2007) buscó descomponer el vector velocidad en dos partes para poder tener una opinión de cambio por el valor contrario al que actualmente tiene la partícula.

Ambos métodos trabajan sobre partículas cuyas posiciones actuales se expresan de manera binaria.

En (Lanzarini et al., 2011a) se propone modificar el vector velocidad utilizando una

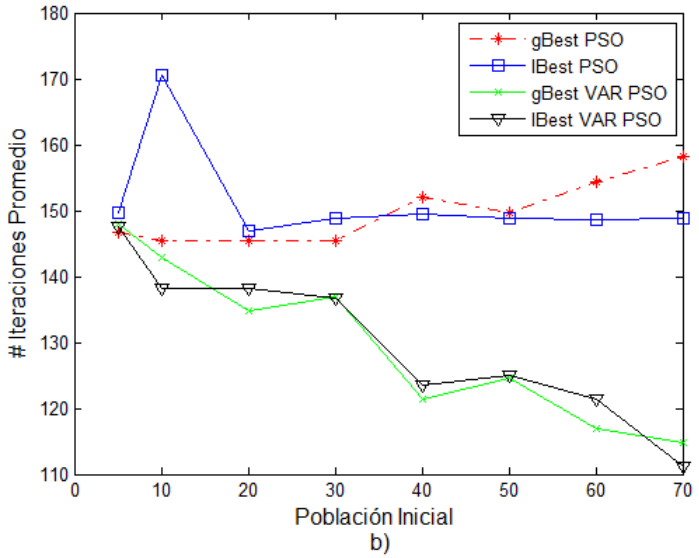


Figura 3.5: Variación del número de iteraciones promedio necesarias para obtener el mejor fitness en función del tamaño de la población inicial.

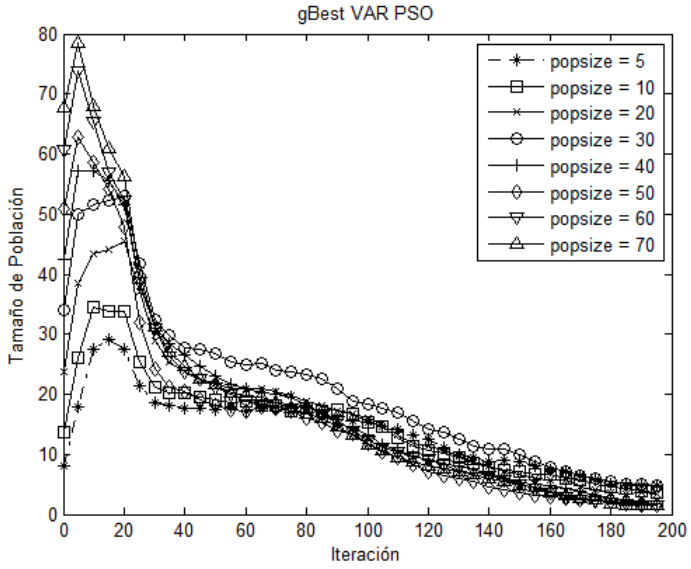


Figura 3.6: Tamaño promedio de la población usando $gBestVarPSO$. Cada punto es el resultado de promediar los mejores fitness de cada una de las 600 pruebas realizadas para cada método utilizando una población inicial determinada.

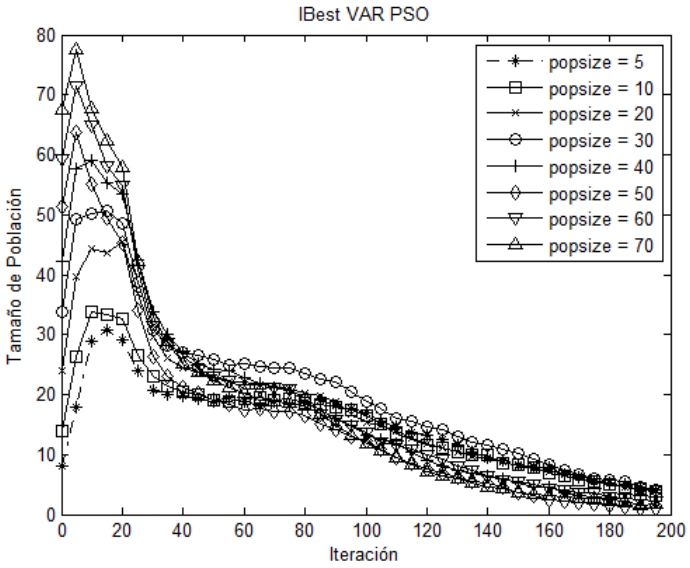


Figura 3.7: Tamaño promedio de la población usando$lBestVarPSO$. Cada punto es el resultado de promediar los mejores fitness de cada una de las 600 pruebas realizadas para cada método utilizando una población inicial determinada.

versión modificada del algoritmo gBest PSO continuo. Es decir que, bajo esta propuesta cada partícula tendrá dos vectores velocidad, V1 y V2. El primero se actualiza según (3.15).

$$V1_i(t+1) = w.V1_i(t) + \varphi_1.rand_1.(2 * pBest_i - 1) + \varphi_2.rand_2.(2 * gBest - 1) \tag{3.15}$$

donde las variables $rand_1$, $rand_2$, φ_1 y φ_2 funcionan de la misma forma que en (3.3). Los valores p_i y g_i corresponden al i-ésimo dígito binario de los vectores $pBest_i$ y $gBest$, respectivamente.
La diferencia más importante entre (3.3) y (3.15) es que en la segunda, el desplazamiento del vector $V1$ en las direcciones correspondientes a la mejor solución encontrada por la partícula y al mejor global no dependen de la posición actual de dicha partícula.
Luego, cada elemento del vector velocidad $V1$ es controlado según (3.16)

$$v1_{ij}(t) = \begin{cases} \delta 1_j & if\ v1_{ij}(t) > \delta 1_j \\ -\delta 1_j & if\ v1_{ij}(t) \leq -\delta 1_j \\ v1_{ij}(t) & if\ not \end{cases} \tag{3.16}$$

donde

$$\delta 1_j = \frac{limit1_{upper_j} - limit1_{lower_j}}{2} \tag{3.17}$$

Es decir que, el vector velocidad $V1$ se calcula según (3.15) y se controla según (3.16). Su valor se utiliza para actualizar el valor del vector velocidad $V2$, como se indica en (3.18).

$$V2(t+1) = V2(t) + V1(t+1) \tag{3.18}$$

El vector $V2$ también se controla de manera similar al vector $V1$ cambiando $limit1_{upper_j}$ y $limit1_{lower_j}$ por $limit2_{upper_j}$ y $limit2_{lower_j}$ respectivamente. Esto dará lugar a $\delta 2_j$ que será utilizado como en (3.16) para acotar los valores de $V2$. Luego se le aplica la función sigmoide y se calcula la nueva posición de la partícula según (3.5).

El concepto de control de velocidad se basa en el método descripto en (López et al., 2009) donde se utiliza un control similar para evitar las oscilaciones finales de las partículas alrededor del óptimo.

3.4.1 Comparación de resultados

En esta sección se compara la performance de la variante de PSO binario propuesta con el método propuesto por Kennedy y Eberhart en (Kennedy and Eberhart, 1997) y el PSO binario definido en (Khanesar et al., 2007), en la minimización de un conocido conjunto de funciones de prueba N dimensionales las cuales se detallan a continuación

a) **Función Sphere** La definición de la función es la siguiente:

$$f_1(X) = \sum_{i=1}^{n} x_i^2$$

Esta es la función de prueba más sencilla. Es continua, convexa y unimodal. El mínimo global es 0 y se encuentra ubicado en $x(i) = 0$ para $i = 1:n$

b) **Función Rosenbrock** En esta función el óptimo global está dentro de un largo y estrecho valle plano con forma parabólica. Si bien encontrar el valle es trivial, la convergencia hacia el óptimo global es difícil y por lo tanto, este problema se ha utilizado en varias ocasiones en evaluar el rendimiento de los algoritmos de optimización.

La definición de la función es

$$f_2(X) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i^2$$

El mínimo global es 0 y se encuentra ubicado en $x(i) = 1$ para $i = 1:n$

c) **Función Griewangk** Se trata de una función con muchos óptimos locales y se define así

$$f_3(X) = \sum_{i}^{n} \frac{x(i)^2}{4000} - \Pi_{i=1}^{n}(cos(\frac{x(i)}{sqrt(i)}) + 1$$

El mínimo global es 0 y se encuentra ubicado en $x(i) = 1$ para $i = 1:n$

d) **Función Rastrigin** La función se define de la siguiente forma

$$f_4(X) = 10n + \sum_{i=1}^{n}(x_i^2 - 10cos(2\pi x_i)$$

Esta función se basa en la función 1 y agrega la función coseno para producir muchos mínimos locales. Esto da lugar a una función multimodal. El mínimo global es 0 y se encuentra ubicado en $x(i) = 0$ para $i = 1 : n$

3.4.2 Pruebas realizadas y parámetros utilizados

Se realizaron 40 corridas independientes de cada uno de los métodos utilizando 2000 iteraciones máximas. Se trabajó con N=3, 5 y 10 variables. El tamaño de la población en todos los casos fue de 20 partículas. Los valores de $limite1$ y $limite2$ son iguales para todas las variables; estos son [0;1] y [0;6] respectivamente. Por lo tanto, los valores de los vectores velocidad $V1$ y $V2$ fueron limitados a los rangos $[-0.5, 0.5]$ y $[-3, 3]$ respectivamente. Es decir que pueden obtenerse probabilidades en el intervalo $[0.0474, 0.9526]$. Los valores para φ_1 y φ_2 fueron establecidos en 0.25. Con respecto a los métodos (Kennedy and Eberhart, 1997) y (Khanesar et al., 2007), se establecieron límites de velocidad entre $[-3, 3]$ a fin de mantener el mismo rango de probabilidades.

3.4.3 Resultados obtenidos

La tabla 3.2 muestra el fitness de la mejor solución encontrada por cada método, así como el valor del fitness promedio de las 40 ejecuciones. Las funciones de prueba utilizadas son las siguientes: Sphere, Rosenbrock, Griewangk y Rastrigin, las cuales aparecen numeradas del 1 al 4 respectivamente. Se recuerda que en todas las funciones se trata de un problema de minimización y por lo tanto se considera mejor a la solución que posea el menor valor de aptitud.

Puede observarse que el método propuesto encuentra las mejores soluciones y posee los menores valores de fitness promedio.
Para analizar si existen diferencias significativas entre los tres métodos analizados se realizó un test ANOVA de un solo factor con un nivel de significación de 0.05 para cada una de las dimensiones consideradas (N=3, 5, 10 y 10). La tabla 3.3 muestra el *p-valor* obtenido en cada caso. Nótese que salvo la función 2 en 3 variables, el resto posee un valor muy inferior al nivel de significación indicando que se rechaza la hipótesis nula.

Para identificar la o las medias significativamente distintas se contruyeron los intervalos de confianza simultáneos para los tres métodos utilizando el mismo nivel de significación. Las figuras 3.8, 3.9, 3.10 y 3.11 grafican los resultados obtenidos.

Tabla 3.2: Resultados Obtenidos al utilizar el método propuesto y los métodos definidos en (Kennedy and Eberhart, 1997) y (Khanesar et al., 2007) para encontrar el mínimo de las funciones Sphere, Rosenbrock, Griewangk y Rastrigin numeradas en la tabla del 1 al 4 respectivamente

		Método Propuesto (Lanzarini et al., 2011a)		PSO Binario (Kennedy and Eberhart, 1997)		Variante de PSO Binario (Khanesar et al., 2007)	
Nro. Var	Nro. Func	Mejor Fitness	Mejor Fitness Promedio	Mejor Fitness	Mejor Fitness Promedio	Mejor Fitness	Mejor Fitness Promedio
3	1	0	**0**	0	1,2e-09	1,8e-08	6,3e-07
3	2	7,0e-04	**2,8**	1,1e-05	3,0	2,0e-03	5,6
3	3	2,1e-09	**3,3e-03**	2,1e-09	6,8e-03	4,2e-06	8,8e-03
3	4	5,4e-08	**5,4e-08**	5,4e-08	**5,4e-08**	8,9e-06	7,6e-04
5	1	0,0	**3,1e-09**	1,4e-07	1,3e-05	1,7e-04	1,2e-03
5	2	2,2	**28,7**	2,1	111,5	7,2	278,3
5	3	2,6e-09	**8,2e-03**	1,6e-03	2,0e-02	1,9e-02	6,6e-02
5	4	9,0e-08	**1,3e-07**	2,3e-04	5,1e-01	5,0e-01	3,7
10	1	8,2e-05	**9,8e-04**	1,3e-02	7,1e-02	9,7e-02	6,2e-01
10	2	7,3	**141,0**	334,0	2812,8	92013,0	613510,0
10	3	1,3e-02	**7,6e-02**	7,7e-02	1,9e-01	5,2e-01	7,3e-01
10	4	0,8	**4,3**	7,5	15,3	13,7	44,0
20	1	3,3e-01	**8,1e-01**	1,7e+00	3,9e+00	1,2e+01	1,9e+01
20	2	1865,9	**10105**	2,8e+05	1,2e+07	1,2e+08	5,0e+08
20	3	1,4e-01	**2,7e-01**	9,3e-01	1,0e+00	1,3e+00	1,4e+00
20	4	27,7	**43,0**	52,7	88,9	169,8	215,2

Tabla 3.3: Resultado del test ANOVA de un solo factor para decidir si existe una diferencia significativa entre los resultados promedio de los métodos utilizados para minimizar las funciones utilizando un nivel de significación de 0.05. La hipótesis nula sostiene que todas las medias son iguales. Para cada caso se indica el *p-valor* obtenido.

nro.Var	Función 1	Función 2	Función 3	Función 4
3	3,029718e-07	0,305974	0,000157	0,003812
5	0	0,000317	0	0
10	0	1,178835e-12	0	0
20	0	0	0	0

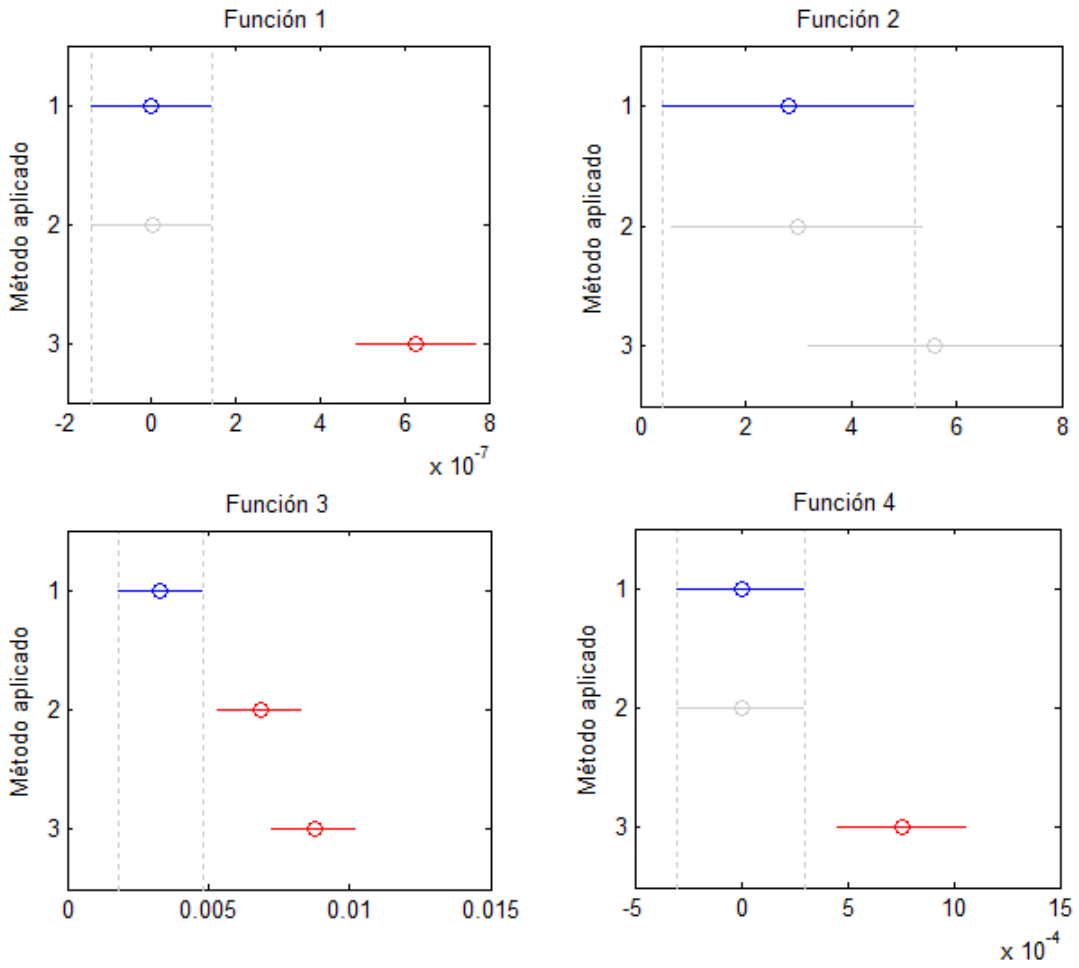


Figura 3.8: Intervalos de confianza simultáneos para el fitness promedio de la mejor solución encontrada por cada uno de los métodos utilizando 3 variables. La numeración asignada es: 1 para el método propuesto (Lanzarini et al., 2011a); 2 y 3 para los métodos definidos en (Kennedy and Eberhart, 1997) y (Khanesar et al., 2007) respectivamente.

Finalmente la tabla 3.4 resume las comparaciones de a pares de medias e indica si la diferencia es significativa o no. En cada caso la media corresponde al promedio de los 40 mejores fitness obtenidos por cada uno de los tres métodos. En dicha tabla se ha utilizado el símbolo ▲ para representar que el IC no contiene al 0, indicando que la hipótesis nula debe ser rechazada. El nivel de significación utilizado es 0.05. El símbolo ▽ indica que la hipótesis nula no se rechaza, es decir que las medias son iguales.

La figura 3.12 muestra los diagramas de caja calculados sobre los mejores resultados obtenidos en cada una de las 40 corridas. Cada columna corresponde a una función distinta. Los diagramas de una misma fila corresponden a la misma cantidad de variables. Considerando de arriba hacia abajo, las filas 1, 2, 3 y 4 corresponden a los resultados obtenidos al evaluar las funciones en 3, 5, 10 y 20 variables respectivamente. En cada figura, los

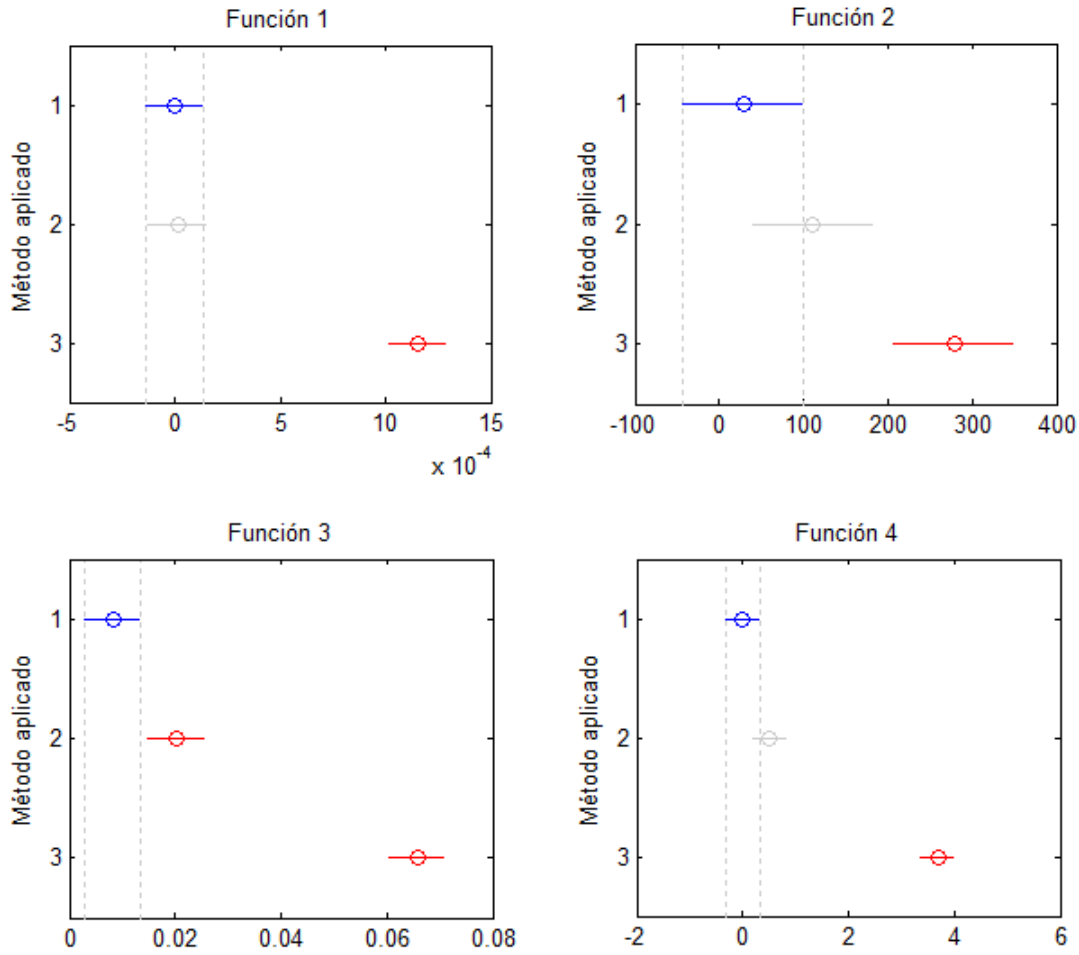


Figura 3.9: Intervalos de confianza simultáneos para el fitness promedio de la mejor solución encontrada por cada uno de los métodos utilizando 5 variables. La numeración asignada es: 1 para el método propuesto (Lanzarini et al., 2011a); 2 y 3 para los métodos definidos en (Kennedy and Eberhart, 1997) y (Khanesar et al., 2007) respectivamente.

métodos están numerados del 1 al 3 correspondiendo a: el método propuesto, PSO Binario (Kennedy and Eberhart, 1997) y PSO Binario (Khanesar et al., 2007) respectivamente. Como puede observarse, efectivamente, el método propuesto ofrece mejores soluciones que las otras dos alternativas de PSO.

La figura 3.13 está organizada de la misma forma que la figura 3.12 y muestra los diagramas de caja correspondientes al fitness promedio de cada una de las 40 corridas realizadas para cada función y para cada cantidad de variables consideradas. En ella puede observarse la diversidad poblacional de cada método. En general, a partir de la altura de cada caja, puede decirse que si bien el método (Khanesar et al., 2007) presenta los mayores rangos intercuartiles, las soluciones que ofrece son las peores. En cuanto al método propuesto y el PSO Binario (Kennedy and Eberhart, 1997), los rangos son equivalentes.

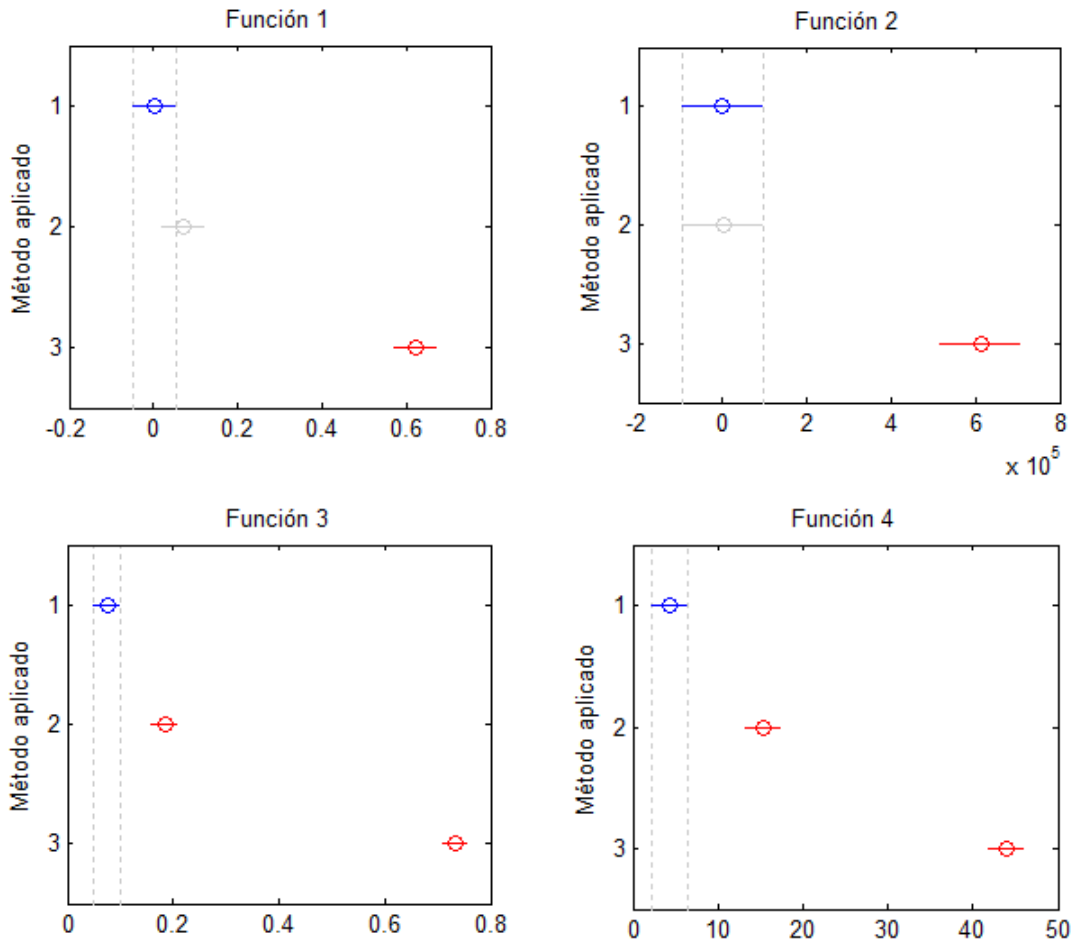


Figura 3.10: Intervalos de confianza simultáneos para el fitness promedio de la mejor solución encontrada por cada uno de los métodos utilizando 10 variables. La numeración asignada es: 1 para el método propuesto (Lanzarini et al., 2011a); 2 y 3 para los métodos definidos en (Kennedy and Eberhart, 1997) y (Khanesar et al., 2007) respectivamente.

En resumen, (Lanzarini et al., 2011a) propone una variante del método PSO binario que controla las modificaciones del vector velocidad. Los resultados obtenidos al minimizar un conjunto de funciones de prueba son mejores a los arrojados por los métodos definidos en (Kennedy and Eberhart, 1997) y (Khanesar et al., 2007).

La tabla 3.4 permite ver que a medida que aumenta la cantidad de variables utilizadas, la diferencia de medias se hace cada vez más significativa.

La figura 3.12 permite mostrar que en la mayoría de las ejecuciones, los resultados obtenidos por el método propuesto han sido superiores a los de los otros dos. Asimismo, en la figura 3.13 se observa que el método propuesto es el que genera los mejores resultados de fitness promedio de la población, por lo menos en las funciones de prueba evaluadas. Si se

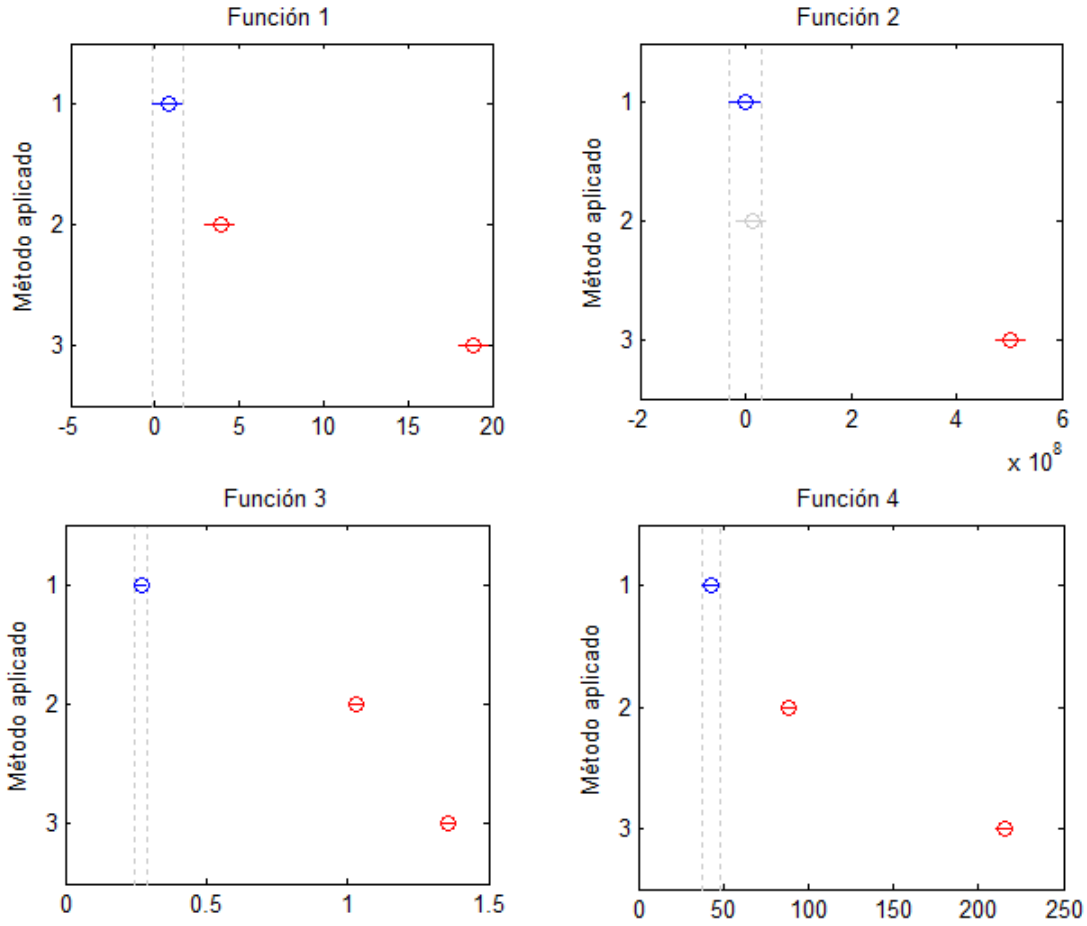


Figura 3.11: Intervalos de confianza simultáneos para el fitness promedio de la mejor solución encontrada por cada uno de los métodos utilizando 20 variables. La numeración asignada es: 1 para el método propuesto (Lanzarini et al., 2011a); 2 y 3 para los métodos definidos en (Kennedy and Eberhart, 1997) y (Khanesar et al., 2007) respectivamente.

considera el rango intercuartil del fitness promedio (amplitud de los diagramas de caja de la figura 2) puede afirmarse que el método propuesto en (Lanzarini et al., 2011a) y el PSO Binario de (Kennedy and Eberhart, 1997) son equivalentes, ofreciendo el primero las mejores soluciones.

3.5 Conclusiones

En este capítulo se detallaron distintos mecanismos de optimización basados en cúmulo de partículas. En particular, los descriptos en las secciones 3.3 y 3.4 forman parte del aporte de esta investigación ya que fueron definidos por la autora de esta tesis y utilizados para

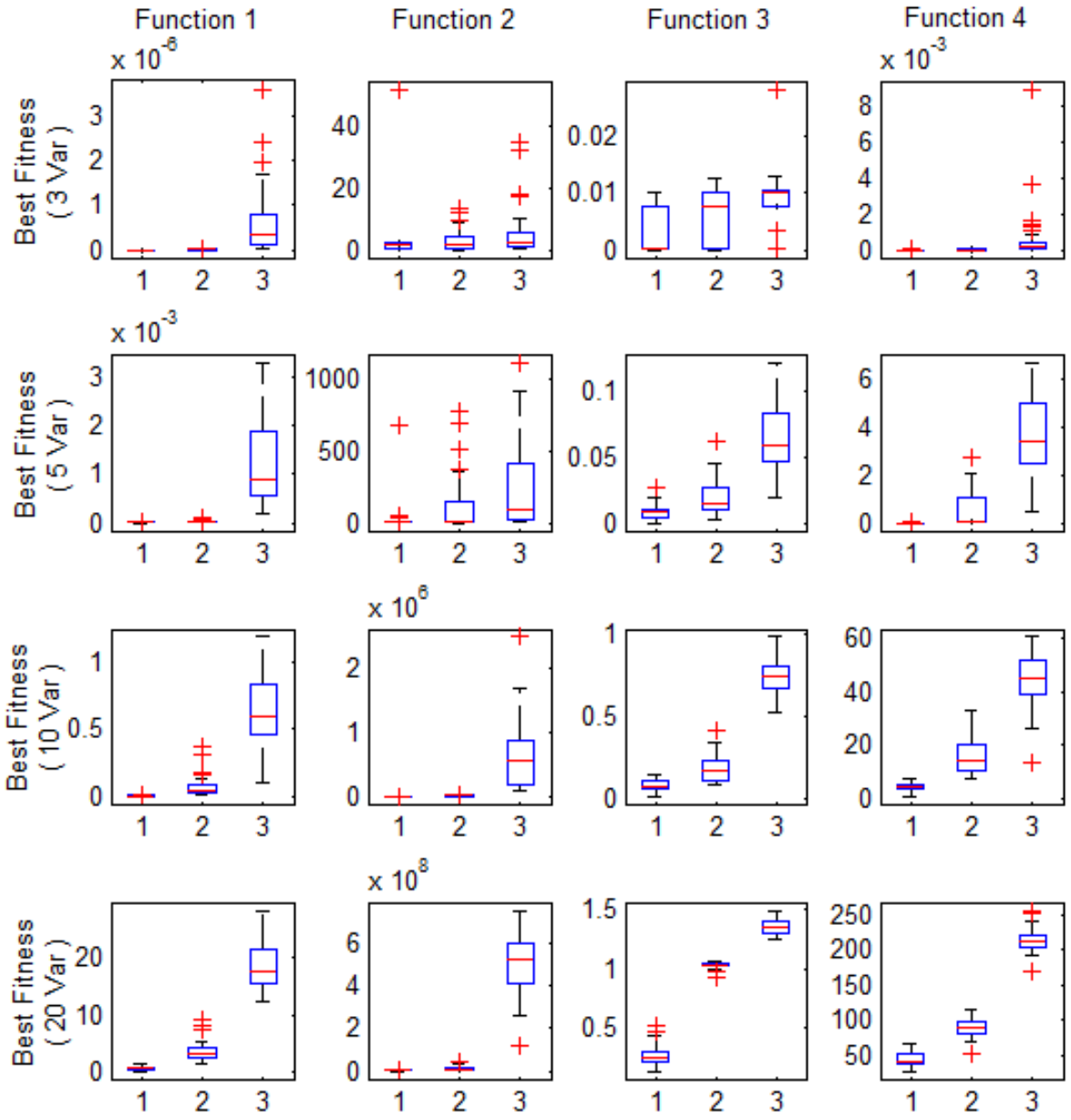


Figura 3.12: Diagramas de caja correspondientes a las mejores soluciones obtenidas en cada una de las 40 corridas independientes. Sobre el eje de las abscisas se indica el método: 1 = Método propuesto (Lanzarini et al., 2011a), 2 = PSO Binario (Kennedy and Eberhart, 1997) y 3 = PSO Binario (Khanesar et al., 2007). Cada fila indica los resultados obtenidos con 3, 5, 10 y 20 variables

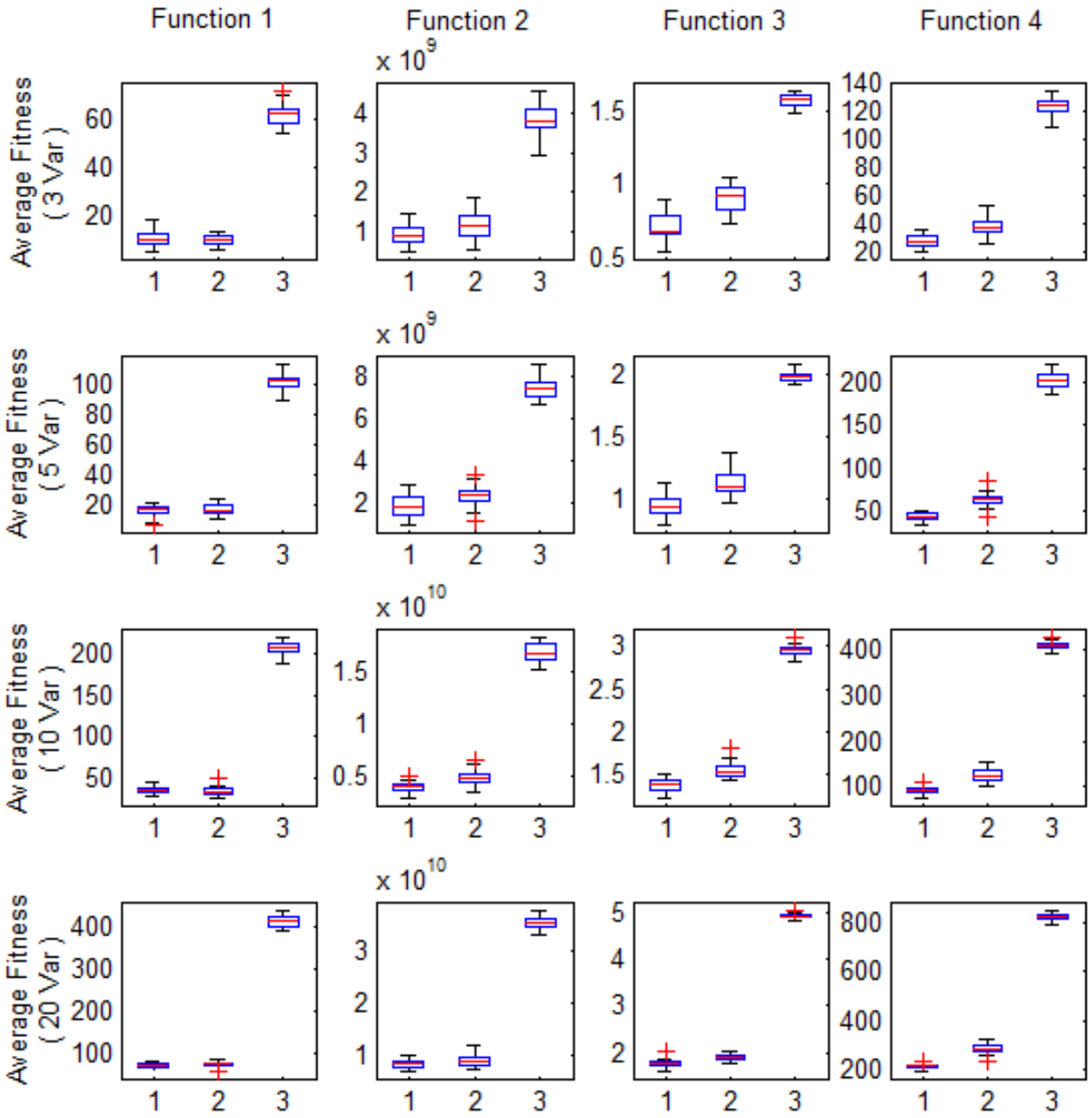


Figura 3.13: Diagramas de caja correspondientes al fitness promedio de cada una de las 40 corridas independientes. Sobre el eje de las abscisas se indica el método: 1 = Método propuesto (Lanzarini et al., 2011a), 2 = PSO Binario (Kennedy and Eberhart, 1997) y 3 = PSO Binario (Khanesar et al., 2007). Cada fila indica los resultados obtenidos con 3, 5, 10 y 20 variables.

Tabla 3.4: Resultados de las comparaciones de los promedios de los 40 mejores fitness obtenidos por cada uno de los métodos. Se han calculado los IC de a pares para un nivel de significación 0.05. El símbolo ▲ representa que el IC no contiene al 0 indicando que la hipótesis nula, que afirma que la media del método indicado en la fila es igual a la del método indicado en la columna, debe ser rechazada

nro Var.	Método	Binary PSO (Kennedy and Eberhart, 1997)				Binary PSO (Khanesar et al., 2007)			
3	Método propuesto (Lanzarini et al., 2011a)	▽	▽	▲	▽	▲	▽	▲	▲
3	Binary PSO (Kennedy and Eberhart, 1997)					▲	▽	▽	▽
5	Método propuesto (Lanzarini et al., 2011a)	▽	▽	▲	▽	▲	▲	▲	▲
5	Binary PSO (Kennedy and Eberhart, 1997)					▲	▽	▲	▲
10	Método propuesto (Lanzarini et al., 2011a)	▽	▽	▲	▲	▲	▲	▲	▲
10	Binary PSO (Kennedy and Eberhart, 1997)					▲	▲	▲	▲
20	Método propuesto (Lanzarini et al., 2011a)	▲	▽	▲	▲	▲	▲	▲	▲
20	Binary PSO (Kennedy and Eberhart, 1997)					▲	▲	▲	▲

dar respuesta al objetivo central de esta tesis.

La optimización por cúmulo de partículas es una técnica sumamente elegante en lo que se refiere a su funcionamiento por el fundamento biológico basado en la teoría darwiniana que le da movimiento a las partículas. Sin embargo, desde el punto de vista del cálculo vectorial, posee un comportamiento elitista que ejerce una fuerte presión sobre el movimiento de las partículas reduciendo abruptamente la diversidad de soluciones ofrecidas por la población. El control del tamaño de la población indicado en 3.3 ha sido fundamental para esta tesis. Es a través de la posibilidad de agregar y quitar individuos que se pudo lograr inspeccionar los lugares adecuados del espacio de búsqueda tal como se detallará en el capítulo siguiente.

Por otro lado, si bien se trata de un problema que escapa a los alcances de este documento, una aplicación interesante obtenida a partir de estas variantes es la construcción de un reconocedor biométrico que permite identificar a una persona por la imagen de su rostro. Dicho reconocedor se basa en la comparación de los descriptores SIFT calculados sobre la imagen con los correspondientes a las imágenes de archivo. El reconocedor propuesto por (Lowe, 2004) posee una alta tasa de falsos positivos. Este es un aspecto no deseado si por ejemplo desea utilizarse el reconocedor para brindar acceso automático a un área restringida. Para mejorar este aspecto se utilizó PSO para seleccionar los descriptores SIFT más representativos de la imagen. De esta forma no sólo se mejoró el reconocimiento sino que también se utilizó dicha selección para reducir el tamaño del almacenamiento de los descriptores correspondientes a las imágenes de archivo. Esto último además redunda en la reducción del tiempo de reconocimiento. El apéndice B contiene un detalle de las pruebas

realizadas así como de los resultados obtenidos.

En el próximo capítulo se hará uso de las variantes de PSO aquí descriptas para hallar un modelo definido por un conjunto de reglas con las características mencionadas originalmente en los objetivos de esta tesis: baja cardinalidad, simplicidad en la definición de las reglas y precisión aceptable para poder determinar un modelo claramente descriptivo.

CAPITULO 4

MÉTODO DE EXTRACCIÓN DE REGLAS PROPUESTO

La obtención de reglas de clasificación es una tarea supervisada que en función de los ejemplos disponibles busca establecer las mejores condiciones para cubrir de manera adecuada la mayor cantidad de casos.

En esta tesis se presenta un nuevo método que utiliza la optimización mediante cúmulo de partículas para construir el conjunto de reglas (Lanzarini et al., 2015d) (Lanzarini et al., 2015c). El énfasis está puesto en alcanzar una buena cobertura utilizando un número reducido de reglas donde cada una de ellas posea un número mínimo de conjunciones en su antecedente. De esta forma se facilitará la interpretación del modelo ayudando a la toma de decisiones.

Para poder aplicar una técnica de optimización poblacional como PSO al problema de extracción de reglas primero debe indicarse la forma en que se resolverán las cuestiones particulares del problema. Estas son:

- **Información a manejar por cada individuo**
  Tal como se describió en el capítulo anterior, PSO es una estrategia poblacional donde los individuos buscan mejorar su capacidad de resolver el problema a medida que el proceso evolutivo avanza. La pregunta aquí es cuál es la función que cumple cada individuo de la población dentro de la solución del problema. Resolver esta cuestión determinará cuál es la información que deberá contener obligatoriamente cada partícula.

  En particular, la extracción de reglas requiere de la combinación de las dos representaciones utilizadas en el capítulo 3: una representación binaria para elegir los

atributos que participarán de la regla y otra continua para delimitar los atributos cuantitativos.

- **Inicialización de la población**
  La ubicación de los individuos en el espacio de búsqueda es un aspecto determinante para lograr buenos resultados. En especial, si se debe trabajar con representaciones binarias se corre el riesgo de obtener valores de aptitud muy disimiles aunque se hayan producido pequeños ajustes en la representación. Por esto, iniciar cerca del óptimo incrementa las calidad de los resultados a la vez que reduce el tiempo de cómputo. Para identificar las zonas más prometedoras se utilizaron dos técnicas de agrupamiento partitivo basadas en centroides.

- **Desplazamiento de las partículas dentro del espacio de búsqueda**
  Operar con una representación doble implica definir el criterio con el que se realizará el movimiento. Aquí deben hacerse ajustes para poder hallar la solución más adecuada dentro de un entorno reducido.

  En este punto también cabe mencionar la variante definida en (Lanzarini et al., 2008) y descripta en la sección 3.3 que ha sido utilizada en esta tesis para mejorar la calidad de la búsqueda. Este enfoque facilita la búsqueda permitiendo realizar una breve expansión al inicio para concentrarse rápidamente en la solución esperada. De no contar con la posibilidad de trabajar con una población variable es preciso indicar a priori la cantidad de partículas que formarán la población. Si este valor se encuentra por debajo de lo necesario, no se logrará llegar a una solución adecuada y por el contrario, si es excesivo el costo computacional se verá incrementado.

- **Evitar la convergencia al mismo conjunto de reglas**
  La optimización por cúmulo de partículas, independientemente de los valores de los parámetros utilizados, basa su capacidad de exploración en la velocidad de inercia de la partícula. Esta velocidad se va reduciendo a lo largo del proceso y la partícula comienza a perder iniciativa acercándose cada vez más al óptimo local encontrado. Esto lleva a que de alguna forma, varios individuos de la población concluyan el mismo resultado cuando analizan los mismo ejemplos. Por esto debe pensarse en alguna estrategia que garantice la diversidad de las reglas que compondrán el conjunto respuesta.

El resto del capítulo buscará dar detalle a cada uno de estos interrogantes.

4.1 Representación de Reglas

En toda estrategia adaptativa, el primer problema a resolver tiene que ver con la definición de la representación a utilizar. Dicha representación es dependiente del problema a resolver y establece qué tipo de información estará contenida en cada elemento de la población.

Existen dos opciones: cada individuo puede corresponderse con el conjunto de reglas completo (enfoque *Pittsburgh* (Smith, 1980)) o cada individuo podrá contener la información de una única regla. Además, si se decide utilizar individuos que representen una única regla, hay dos posibilidades según la manera en que se obtenga el resultado:

- En el enfoque *Michigan* cada individuo codifica una única regla pero la solución final será o bien un subconjunto de la población o la población completa (Holland and Reitman, 1977).

- En el enfoque *IRL (por sus siglas del inglés Iterative Rule Learning)* se utiliza un proceso iterativo que por cada ejecución extrae el mejor individuo, el cual nuevamente contiene una única regla. Luego de la obtención de cada regla se quitan del conjunto de ejemplo de entrada los casos correctamente cubiertos y se repite el proceso. Esto evita generar reglas similares pero, a diferencia del enfoque *Michigan*, las reglas deben ser aplicadas en el mismo orden en que fueron generadas (Venturini, 1993).

La decisión de cuál representación utilizar depende del conocimiento que se tenga del problema.

Por un lado, si se utiliza una representación donde un mismo individuo contiene el conjunto de reglas completo se facilita la evaluación de la aptitud de la solución encontrada ya que un mismo individuo contiene todos los datos necesarios para hacerlo. Sin embargo el problema está en la cantidad de reglas a utilizar porque, dado que por lo general se desconoce a priori la cantidad exacta de reglas que conforman el modelo, deberá trabajarse con individuos excesivamente largos o de longitud variable. Por lo tanto, se obtendrán algoritmos computacionalmente más costosos o más complejos respectivamente.

Por otro lado, si se utiliza una representación donde cada individuo contiene una única regla tendrá una longitud menor pero la evaluación de la aptitud de la solución hallada dependerá de evaluar a más de un individuo. Además, debe considerarse el uso de alguna técnica de especiación ya que el uso de una técnica de optimización hará que varios individuos converjan en la misma regla.

En esta tesis se decidió utilizar el enfoque *IRL (Iterative Rule Learning)* para no tener que establecer a priori la cantidad máxima de reglas que conformarán el modelo. El problema de la convergencia prematura a un único individuo se resolverá mediante ejecuciones sucesivas del mismo proceso sobre conjuntos de ejemplos cada vez más reducidos.

Decidido el enfoque de la representación, debe indicarse la manera en que se codificará el antecedente y el consecuente de la regla.

4.1.1 Representación del antecedente

Las reglas a obtener serán de la forma:

```
SI <condición 1>  Y <condición 2>  Y ... Y <condición N>  entonces
                              <consecuente>
```

El antecedente es una conjunción donde cada condición se refiere a un atributo diferente.

Poder operar con atributos nominales y numéricos requiere incluir en la representación una manera de determinar cuáles serán los atributos que conforman la regla y cual será su valor o rango de valores según sean nominales o numéricos respectivamente.

Con respecto al antecedente, se utilizará una representación de longitud fija que operará sobre atributos numéricos y nominales formada por dos partes:

- **Parte binaria**
  Permite identificar cuales son las condiciones que formarán parte del antecedente. En el caso de los atributos cualitativos utiliza tantos dígitos binarios como valores diferentes presente dicho atributo. En el caso de los atributos numéricos utiliza sólo dos dígitos binarios para indicar si interviene el límite inferior y/o superior de dicho atributo. Su valor será 1 cuando se utiliza y 0 si no.

- **Parte Real**
  Contiene los valores que permiten acotar a los atributos numéricos a la hora de conformar la condición. Dichos valores son calculados a través de la técnica de optimización evitando de esta forma tener que discretizar los atributos numéricos antes de iniciar el proceso. Su longitud coincide con la parte binaria aunque sólo se utiliza para los atributos numéricos. Este aspecto, si bien incrementa levemente la longitud del individuo facilita el funcionamiento del algoritmo redundando en un menor tiempo de convergencia.

La figura 4.1 ejemplifica la codificación del antecedente de una regla para la base StatLog (German Credit Data) del repositorio UCI (Bache and Lichman, 2013). Se trata de una base con 20 atributos de los cuales 7 son numéricos y 13 son categóricos o cualitativos. En el ejemplo sólo cuatro atributos presentan el valor 1 en la parte binaria y por lo tanto serán los únicos que conformarán el antecedente. Nótese que la parte real sólo presenta valores distintos de cero en las posiciones correspondientes a los atributos numéricos y que es el vector binario el que indica si deben utilizarse o no. También se observa en la

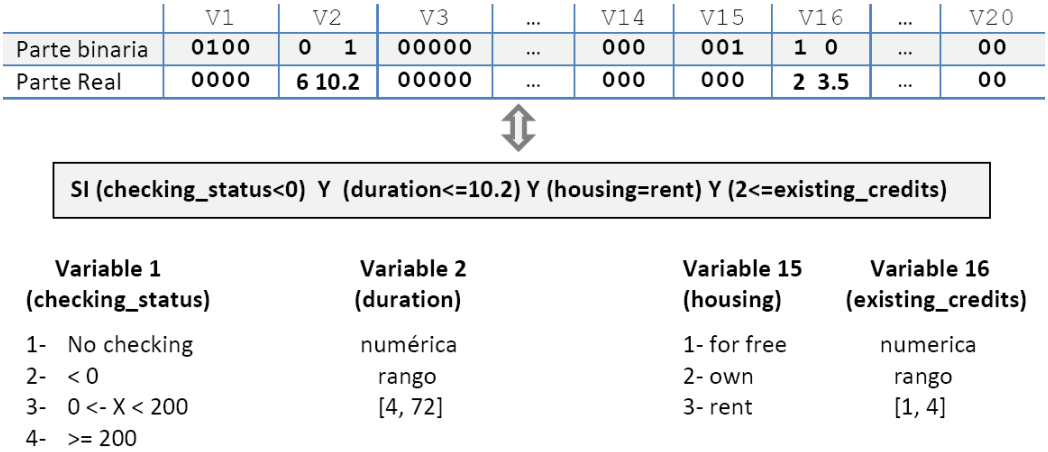


Figura 4.1: Ejemplo de codificación del antecedente de una regla para la base StatLog (German Credit Data) del repositorio UCI (Bache and Lichman, 2013)

figura que cada atributo cualitativo tiene a los sumo un único dígito binario en 1. Esto es regulado por la técnica de optimización y tiene por objetivo obtener antecedentes fáciles de leer entendiendo por tales aquellos que estén formados por pocas condiciones.

4.1.2 Representación del consecuente

En lo que se refiere al consecuente, en (Freitas, 2003) se describen al menos tres representaciones diferentes que pueden utilizarse a la hora de aplicar una estrategia evolutiva:

La primera posibilidad es codificarlo dentro del genoma del individuo y posiblemente someterlo a evolución.

La segunda posibilidad es asociar todos los individuos de la población con la misma clase la cual es elegida antes de comenzar a ejecutar el algoritmo y no se modifica durante el proceso. Es decir que si se busca construir un conjunto de reglas que deban cubrir un conjunto k de clases diferentes se deberá ejecutar el algoritmo al menos k veces de manera que en el i-ésima ejecución, el algoritmo descubra reglas sólo para la i-ésima clase, con $i = 1..k$.

La tercera posibilidad es elegir la clase más adecuada para formar la regla una vez que el antecedente haya sido construido. Esta elección puede basarse en criterios tales como la clase que tiene más representantes en el conjunto de ejemplos que cumplen con el antecedente o la clase que maximiza el valor de fitness del individuo.

Analizando las ventajas y desventajas de cada una puede decirse que la primera y la tercera tienen la ventaja de permitir que individuos distintos dentro de la misma población representen reglas correspondientes a clases diferentes. Como desventaja debe

considerarse la complejidad de los operadores genéticos en la primera y el tiempo requerido para identificar la clase adecuada para formar el consecuente en la tercera.

En esta tesis, se ha decidido utilizar una representación de longitud fija donde sólo se codificará el antecedente de la regla. Es decir que se optará por la segunda posibilidad donde todos los individuos de la población corresponden a reglas de una clase predeterminada. En resumen, no se incluirá el consecuente de la regla en la codificación del individuo.

4.2 Estructura de una partícula

La obtención de reglas de clasificación utilizando PSO, capaces de operar sobre atributos nominales y numéricos, requiere de una combinación de los métodos citados anteriormente (secciones 3.1 y 3.2) ya que es preciso decir cuáles serán los atributos que formarán parte del antecedente (discreto) y para los atributos numéricos es preciso determinar el intervalo a utilizar (continuo).

La i-ésima partícula de la población se representa de la siguiente forma:

- $pBin_i = (pBin_{i1}, pBin_{i2}, \ldots, pBin_{in})$ almacena la posición actual de la partícula e indica cuáles son los ítems o condiciones que componen el antecedente de la regla según PSO.
- $v1_i = (v1_{i1}, v1_{i2}, \ldots, v1_{in})$ y $v2_i = (v2_{i1}, v2_{i2}, \ldots, v2_{in})$ se combinan para determinar la dirección en la cual se moverá la partícula.
- $pBestBin_i = (pBestBin_{i1}, pBestBin_{i2}, \ldots, pBestBin_{in})$ almacena la mejor solución encontrada por la partícula hasta el momento.
- $fitness_i$ es el valor de aptitud del individuo.
- $fitness\_pBest_i$ es el valor de aptitud de la mejor solución local encontrada (vector $pBestBin_i$)
- $pReal_i = (pReal_{i1}, pReal_{i2}, \ldots, pReal_{in})$ se utiliza para los atributos numéricos y contiene los límites actuales de los intervalos normalizados linealmente entre 0 y 1.
- $v3_i = (v3_{i1}, v3_{i2}, \ldots, v3_{in})$ indica la dirección de cambio de $pReal_i$.
- $pBestReal_i = (pBestReal_{i1}, pBestReal_{i2}, \ldots, pBestReal_{in})$ almacena la mejor solución encontrada por la partícula para los límites de los intervalos.
- $sopBin_i = (sopBin_{i1}, sopBin_{i2}, \ldots, sopBin_{in})$ indica cuáles son los ítems o condiciones que componen el antecedente de la regla que efectivamente representa la partícula y cuyo fitness se encuentra en $fitness_i$.

- TV es un número entero que indica el tiempo de vida que le queda a la partícula. Este dato sólo se utiliza cuando se trabaja con un tamaño de población variable.

Cada vez que la i-ésima partícula se mueve se modifica su posición actual y los intervalos correspondientes a los atributos numéricos de la siguiente forma:

Parte binaria

La identificación de los atributos que forman el antecedente de la regla se realiza mediante el vector binario $pBin_i$. Su movimiento se controla a través de dos vectores de velocidad $v1_i$ y $v2_i$ con el objetivo de mejorar la estabilidad durante el desplazamiento de la partícula. También se tiene en cuenta la mejor solución encontrada por la partícula, $pBestBin_i$ y la solución más cercana, dentro del espacio de búsqueda, con un valor de aptitud superior al de la partícula a mover, $lBestBin_i$. La ecuación (4.1) indica la manera en que se actualiza el vector $v1_i$

$$v1_{ij}(t+1) = w_{bin} \cdot v1_{ij}(t) + \varphi_1 \cdot rand_1 \cdot (2\, pBestBin_{ij} - 1) + \varphi_2 \cdot rand_2 \cdot (2\, lBestBin_{ij} - 1) \tag{4.1}$$

donde, w_{bin} representa el factor de inercia, $rand_1$ y $rand_2$ son valores aleatorios con distribución uniforme en [0,1] y φ_1 y φ_2 son valores constantes que indican la importancia que se desea darle a las respectivas soluciones halladas previamente. Los valores $pBestBin_{ij}$ y $lBestBin_{ij}$ corresponden al j-ésimo dígito de los vectores binarios $pBestBin_i$ y $lBestBin_i$ respectivamente. Como se explicó previamente, en el método propuesto, cada partícula tendrá en cuenta la posición de su vecino más cercano con un valor de aptitud superior al suyo; por lo tanto el valor de $lBestBin_i$ corresponde al vector de $pBestBin_j$ de la partícula más cercana a $pBestBin_i$ siempre que $fitness_j$ sea mayor que $fitness_i$ usando distancia euclídea.

Nótese que, a diferencia del PSO Binario descripto en la sección 3.2, el desplazamiento del vector $v1_i$ no depende de la posición actual de la partícula, $pBin_i$, sino que se realiza en las direcciones correspondientes a la mejor solución encontrada por la partícula $pBestBin_i$ y al mejor local $lBestBin$. Además, en la ecuación (4.1) se transforma cada posición de ambos vectores en bipolar para permitir que la velocidad pueda disminuir de valor.

Luego, cada elemento del vector velocidad $v1_i$ es controlado según 4.2 como fue indicado en la variante de PSO Binario definida en la sección 3.4

$$v1_{ij}(t) = \begin{cases} \delta 1_j & si\ v1_{ij}(t) > \delta 1_j \\ -\delta 1_j & si\ v1_{ij}(t) \leq \delta 1_j \\ v1_{ij}(t) & en\ caso\ contrario \end{cases} \tag{4.2}$$

donde

$$\delta 1_j = \frac{limite1_{superior_j} - limite1_{inferior_j}}{2} \tag{4.3}$$

Es decir que, el vector velocidad $v1_i$ se calcula según 4.1 y se controla según 4.2. Su valor se utiliza para actualizar el valor del vector velocidad $v2_i$, como se indica en 4.4.

$$v2_{ij}(t+1) = v2_{ij}(t) + v1_{ij}(t+1) \tag{4.4}$$

El vector $v2_i$ también se controla de manera similar al vector $v1_i$ cambiando $limite1_{superior_j}$ y $limite1_{inferior_j}$ por $limite2_{superior_j}$ y $limite2_{inferior_j}$ respectivamente. Esto dará lugar a $\delta 2_j$ que será utilizado como en 4.2 para acotar los valores de $v2_i$. Luego se le aplica la función sigmoide 4.5 y se calcula la nueva posición de la partícula según (4.6).

$$sig(x) = \frac{1}{1+e^{-x}} \tag{4.5}$$

$$pBin_{ij}(t+1) = \begin{cases} 1 & si\ rand_{ij} < sig(v2_{ij}(t+1)) \\ 0 & si\ no \end{cases} \tag{4.6}$$

donde $rand_{ij}$ es un número aleatorio con distribución uniforme en [0,1].

Parte continua

Esta parte controla los límites de los atributos numéricos. Se calculan siempre pero sólo intervienen en la regla si la parte binaria así lo indica. Se calcula de la manera habitual sumando al vector posición $pReal_i$ el valor del vector velocidad $v3_i$ y controlando que no supere los límites permitidos.

$$pReal_{ij}(t+1) = pReal_{ij}(t) + v3_{ij}(t+1) \tag{4.7}$$

$$\begin{aligned} v3_{ij}(t+1) \quad = \quad & w_{Real} \cdot v3_{ij}(t) + \varphi_3 \cdot rand_3 \cdot (pBestReal_{ij} - pReal_{ij}) + \\ & \varphi_4 \cdot rand_4 \cdot (lBestReal_{ij} - pReal_{ij}) \end{aligned} \tag{4.8}$$

donde nuevamente, w_{Real} representa el factor de inercia, $rand_3$ y $rand_4$ son valores aleatorios con distribución uniforme en [0,1] y φ_3 y φ_4 son valores constantes de indican la

importancia que se desea darle a las respectivas soluciones halladas previamente. En este caso, $lBestReal_i$ corresponde al vector $pReal_j$ de la partícula más cercana a $pReal_i$ donde $fitness_j$ es mayor que $fitness_i$ usando distancia euclídea. Esta es la misma partícula de la que se tomó el vector $pBin_i$ para realizar el ajuste de $v1_i$ en 4.1. Los valores asignados a w_{Real}, $\varphi 1$, $\varphi 2$, $\varphi 3$ y $\varphi 4$ son importantes para garantizar la convergencia del algoritmo.

Un detalle a tener en cuenta es la manera en que debe modificarse el vector velocidad cuando se utiliza función sigmoide 3.4. En PSO continuo el objetivo del vector velocidad es permitir que la partícula realice inicialmente una tarea exploratoria para luego especializar la búsqueda en una zona identificada como prometedora. Para esto, recibe un valor inicial que se va decrementando generalmente en forma proporcional a la cantidad de iteraciones máximas a realizar. En este caso el vector velocidad representa la inercia de la partícula y es el único factor que evita que sea atraída fuertemente o bien por sus experiencias anteriores o bien por la mejor solución hallada por el cúmulo.

Por el contrario, cuando se utiliza una representación binaria, si bien el movimiento sigue siendo real, es la función sigmoide la que se ocupa de binarizar el resultado identificando la nueva posición. En este caso, para tener capacidad exploratoria, es preciso que la función sigmoide comience evaluándose en valores cercanos al cero donde tiene mayor posibilidad de cambio. En particular, la sigmoide indicada en 3.4 cuando x vale 0 da como resultado 0.5. Este es el mayor estado de incertidumbre cuando la respuesta esperada es 0 o 1. Luego, a medida que se aleja del 0, ya sea en forma positiva o negativa, su valor se estabiliza. Por lo tanto, a diferencia de lo realizado sobre la parte continua, cuando se opera con PSO binario debe comenzarse con una velocidad cercana a 0 para luego incrementar o decrementar su valor.

En este trabajo, los valores de $limite1$ y $limite2$ son iguales para todos los valores de los vectores velocidad de la parte binaria; estos son [0,1] y [0,6] respectivamente. Por lo tanto, los valores de los vectores velocidad $v1$ y $v3$ fueron limitados a los rangos [-0.5, 0.5] y [-3,3] respectivamente. Es decir que pueden obtenerse probabilidades en el intervalo [0.0474, 0.9526]. Los valores para $\varphi 1$, $\varphi 2$, $\varphi 3$ y $\varphi 4$ fueron establecidos en 0.25, 0.25, 0.5 y 0.25 respectivamente. Los valores de w_{bin} y w_{Real} fueron establecidos entre 1.25 y 0.25 de manera lineal y proporcional a la cantidad de iteraciones realizadas, en forma ascendente para w_{bin} y en forma descendente para w_{Real}.

La eficiencia de las técnicas de optimización poblacionales se encuentra estrechamente relacionada con el tamaño de la población. Por tal motivo, el método propuesto utiliza la estrategia de población variable definida en (Lanzarini et al., 2008). De esta forma, es posible comenzar con una población de tamaño mínimo e ir ajustando la cantidad de partículas durante el proceso adaptativo.

4.3 Aptitud de una partícula

El valor de aptitud de cada partícula se calcula de la siguiente forma:

$$Fitness = \alpha * balance * support * confidence - \beta * lengthAntecedent$$

donde

- $support$: es el soporte de la regla. Esto es el cociente entre la cantidad de ejemplos que cumplen con la regla dividida por la cantidad total de ejemplos que se están analizando.
- $confidence$: es la confianza de la regla que se calcula como el cociente entre la cantidad de ejemplos que cumplen con la regla y la cantidad de ejemplos que cumplen sólo con el antecedente.
- $lengthAntecedent$: es el cociente entre la cantidad de condiciones utilizadas en el antecedente dividida por la cantidad total de atributos. Debe considerarse que un mismo atributo sólo puede aparecer una vez dentro del antecedente de la regla.
- α, β: son dos constantes que representan la importancia que se le da a cada término.
- $balance$: toma valores entre (0,1] y permite compensar el efecto que tiene el desbalance entre clases a la hora de calcular el soporte. Sólo se aplica cuando se está trabajando con clases que poseen una cantidad de ejemplos superior a la media. Sean $C_1, C_2, ..., C_i, ..., C_N$ las clases en las que se dividen los ejemplos. N es la cantidad total de clases. Sea E_i la cantidad de ejemplos de la i-ésima clase. Sea T el total de ejemplos con los que se está trabajando. Es decir que

  $$T = \sum_{i=1}^{N} E_i$$

  Sea j la clase a la que pertenece la regla representada por la partícula. Sea S_i la cantidad de ejemplos de la clase C_i cubiertos por la regla. Note que S_j se corresponde con el soporte de la regla y que

  $$\sum_{i=1, j \neq i}^{N} S_i$$

  es la cantidad total de ejemplos incorrectamente cubiertos por dicha regla. Luego, el valor de este factor se calcula de la siguiente forma

  $$balance = \sum_{i=1, i \neq j}^{N} \frac{E_i - S_i}{T - E_j}$$

Es decir que $balance$ valdrá 1 si la regla es perfecta, es decir, si tiene confianza 1 y valdrá 0 si la regla cubre a todos los ejemplos con los que se esté trabajando sin importar a qué clase pertenezcan.

La evaluación del desempeño del individuo no se limita sólo a una única regla sino que utiliza al individuo binario como la selección de los ítems que pueden formar el antecedente pero no se considera obligatorio utilizarlos a todos.

Para ayudar en la simplificación del antecedente cada vez que se evalúa la partícula también se analizan las distintas reglas que se forman al suprimir una a una las condiciones que participan en el antecedente. De todas las reglas evaluadas se toma la de mayor fitness. Si la regla seleccionada no es la original, se repite el proceso en busca de una nueva reducción. Por lo tanto, si la regla original tiene$N$ condiciones, se evaluarán como mínimo N y como máximo $N*(N-1)$ variantes de la regla propuesta por la partícula.

El resultado de esta reducción hace que el vector $pBin_i$ se convierta en el vector $sopBin_i$ conservando sólo las condiciones relevantes con valor 1 y el resto con 0. Para no ejercer una influencia excesiva en la manera en que opera la técnica de optimización, las condiciones que hubieran sido eliminadas de $pBin_i$ para dar lugar a $sopBin$ recibirán una reducción del 2% en $v1$ y del 25% en $v2$. De esta forma se reduce la posibilidad de que las condiciones eliminadas sean seleccionadas en el próximo movimiento de la partícula pero no se anulan totalmente dándole la posibilidad a PSO de explorar cerca de la solución actualmente propuesta por la partícula.

Finalmente, el fitness de la partícula corresponderá al antecedente indicado por $sopBin$ aunque la partícula se siga desplazando de la manera convencional utilizando ambos vectores de velocidad.

4.4 Método propuesto

El proceso comienza con el agrupamiento de los ejemplos de entrenamiento a través de una red neuronal competitiva. Dado que las redes neuronales sólo operan con datos numéricos, los atributos nominales son representados en forma binaria. Además, antes de iniciar el entrenamiento, cada atributo numérico es escalado linealmente entre 0 y 1. La medida de similitud utilizada es distancia euclídea. Según quiera utilizarse una estrategia adaptativa supervisada o no puede aplicarse o bien una red LVQ o una red SOM. El tamaño de la red varía según la versión de PSO que se utilice siendo menor cuando la población es de tamaño variable. Para utilizar una población de tamaño fijo debe estimarse adecuadamente la cantidad de neuronas competitivas que participarán de la arquitectura o directamente emplear una red competitiva dinámica. Cuando haya finalizado el entrenamiento de la red neuronal se dispondrá de información referida a las zonas más prometedoras del espacio de

búsqueda que puede ser aprovechada cada vez que deban crearse partículas, ya sea para formar la población inicial como para generar nuevas cuando la población deba incrementar su tamaño.

Por otro lado, deben definirse las cantidades mínimas de ejemplos que pueden quedar sin cubrir dentro de cada clase. El objetivo del método propuesto es hallar las reglas más representativas por lo que es de esperar que la cobertura no sea total. Este dato no es privativo del método propuesto ya que forma parte de la cota de error tolerable para el modelo a determinar.

A partir de este momento comienza un proceso iterativo que aplica la variante correspondiente de PSO a la clase con mayor cantidad de ejemplos. Como se explicó previamente, el consecuente de la regla no se encuentra indicado en la partícula ya que todas corresponden a la misma clase. La clase mayoritaria.

Una vez seleccionada la clase, se aplica la técnica de optimización para hallar el antecedente de la primera regla.

La población de partículas se inicializa utilizando la red neuronal competitiva previamente entrenada. Se crearán tantas partículas como neuronas haya. Para reducir el tiempo de creación de la población, cada neurona contendrá, en los pesos de los arcos que llegan hasta ella, información de la ubicación del centroide y de la dispersión de los ejemplos a los cuales representa. Se trata de dos vectores con la misma dimensión que los ejemplos de entrenamiento y con valores equivalentes al promedio de los que se consideran representados por dicha neurona y a las desviaciones estándar correspondientes. La información del centroide se utiliza para determinar el vector $v2$ descripto en la sección 4.2. Si se trata de un atributo nominal, dicha información se escala linealmente en el intervalo $[limite2_{inferior_j}, limite2_{superior_j}]$ pero si se trata de un atributo numérico el valor a escalar es el máximo entre 0 y $(1-2*desviacion_j)$ siendo $desviacion_j$ la j-ésima posición de la desviación de los ejemplos representados por el centroide. En ambos casos se pretende operar con un valor entre 0 y 1 que mida el grado de participación del atributo (si es numérico) o del valor del atributo (si es nominal) en la construcción del antecedente de la regla. En el caso de los atributos nominales, hay una posición dentro del centroide para cada uno de sus valores posibles. Por lo tanto, el valor que cada una de esas posiciones coincidirá con la proporción de ejemplos representados que lo posean. Al escalarlo linealmente entre $[limite2_{inferior_j}, limite2_{superior_j}]$ tendrán mayor posibilidad de ser elegidos para formar parte del antecedente los que se encuentren presentes en un mayor números de ejemplos. En cambio, en el caso de los atributos numéricos, es la desviación la que permite determinar la cercanía del valor del ejemplo con el centroide. Por lo tanto, se buscará darle las mayores oportunidades de ser seleccionados a los que posean desviación nula. Como se trata de promedios de valores numéricos entre 0 y 1, los desvíos suelen ser muy inferiores a 1; de allí que el valor a escalar entre $[limite2_{inferior_j}, limite2_{superior_j}]$ sea el máximo entre 0 y

$(1-2*desviacion_j)$. En todos los casos la velocidad $v1$ se inicializa en forma aleatoria en $[limite1_{inferior_j}, limite1_{superior_j}]$.

Una vez creada la población inicial, se la evoluciona utilizando la versión de PSO elegida.

Debe recordarse que si se trabaja con un cúmulo de partículas de población variable es posible incorporar nuevos elementos durante la adaptación (3.3). Se trata de un proceso de dos partes: primero deben identificarse cuáles son las partículas aisladas y luego agregar una variante del centroide que le dió origen a cada una de ellas de manera de incentivar la búsqueda en las zonas menos exploradas. Además cada partícula llevará un tiempo de vida asociado que se irá reduciendo a lo largo de las sucesivas iteraciones. Cuando este valor llegue a cero, la partícula será eliminada de la población.

Al finalizar este proceso, se selecciona la mejor regla de la población y si cumple con los requerimientos de soporte y confianza pedidos se la agrega al conjunto de reglas y los ejemplos correctamente cubiertos por ella son retirados del conjunto de ejemplos de entrenamiento. En este último caso, se produce una reducción de ejemplos sin cubrir dentro de la clase y por lo tanto el porcentaje mínimo de ejemplo que puede quedar sin cubrir debe reducirse.

Este proceso continúa hasta lograr cubrir todos los ejemplos o hasta que la cantidad de ejemplos no cubiertos de cada clase se encuentre por debajo del respectivo soporte mínimo establecido o hasta que se hayan realizado la máxima cantidad de intentos por obtener una regla, lo que ocurra primero. Es importante tener en cuenta es que, dado que los ejemplos son retirados del conjunto de datos de entrada a medida que son cubiertos por las reglas, las mismas constituyen una lista de clasificación. Es decir que, para clasificar un ejemplo nuevo, las reglas deben ser aplicadas en el orden en que fueron obtenidas y el ejemplo será clasificado con la clase correspondiente al consecuente de la primera regla cuyo antecedente se verifique para el ejemplo en cuestión.

El algoritmo 4 contiene el pseudocódigo del método propuesto.

4.5 Resultados obtenidos

Para medir el desempeño del método propuesto se buscó obtener los conjuntos de reglas adecuados para describir y clasificar la información almacenada en 13 bases de datos del repositorio UCI (Bache and Lichman, 2013). En la tabla 4.1 se indica para cada una de ellas la cantidad de ejemplos que contiene, la cantidad de atributos que poseen de cada tipo y la cantidad de clases en las que dichos ejemplos pueden ser clasificados.

La inicialización del cúmulo de partículas se realizó de dos formas distintas: con una red SOM y con una red LVQ. Cuando se trabajó con poblaciones de tamaño fijo se utilizaron

Algoritmo 4: Pseudocódigo del método propuesto

```
Entrenar la red competitiva utilizando todos los ejemplos de entrenamiento;
Calcular el soporte mínimo para cada clase;
while no se alcance el criterio de terminación do
    Elegir la clase con mayor número de ejemplos no cubiertos;
    Construir una poblacion reducida de individuos a partir de los centroides;
    Evolucionar la población utilizando una versión de PSO;
    Obtener la mejor regla de la población;
    if la regla cumple con el soporte y la confianza pedidos then
        Agregar la regla al conjunto de reglas ;
        Considerar como cubiertos los ejemplos correctamente clasificados por la regla
        anterior;
        Recalcular el soporte mínimo para esta clase;
```

redes neuronales de 30 neuronas competitivas. Para la red SOM se utilizó una grilla de 6x5 con 4 vecinas como máximo por neurona. Estas combinaciones aparecen en las tablas de resultados como somPSO y lvqPSO. Las versiones con población variable utilizan la estrategia de edad para modificar la cantidad de individuos y pueden comenzar con una población menor ya que agregarán o quitarán partículas durante el proceso de búsqueda según consideren adecuado.

La cantidad de ejemplos que pueden quedar sin cubrir dentro de cada clase se fijó linealmente entre un 6% y un 1% según en número de intentos realizados.

Los resultados obtenidos fueron comparados con los métodos PART (Frank and Witten, 1998b), cAntMinerPB (Medland et al., 2012) y C4.5 (Quinlan, 1993) descriptos en el capítulo 2.

Se realizaron 30 corridas independientes de cada método. Para los métodos somVPSO y lvqVPSO se utilizó una población inicial de 20 partículas con un tiempo de vida máximo de 10 iteraciones asignado en forma proporcional durante las primeras 5 y luego en forma fija por grupo. Se agruparon los valores de fitness en 4 grupos.

Para los métodos PART y C4.5 se utilizó un factor de confianza para el podado del árbol de 0.3 y 0.25 respectivamente. El método cAntMiner fue ejecutado con una colonia de 9 hormigas. Para el resto de los parámetros de estos métodos se utilizaron los valores por defecto.

Las tablas 4.2, 4.3 and 4.4 resumen los resultados obtenidos de aplicar los métodos a cada una de las bases calculando media y desviación. En cada caso se ha considerado no sólo la precisión de la cobertura del conjunto de reglas (tabla 4.2) sino también la claridad del modelo obtenido; esto último se refleja en la cantidad promedio de reglas obtenidas (tabla 4.3) y en la cantidad promedio de términos utilizados para formar el antecedente

Tabla 4.1: Bases de datos utilizadas para medir el desempeño del método propuesto

Base de Datos	#Ejemplos	#Atrib.Numéricos	#Atrib.Nominales	#Clases
Balance scale	625	4	0	3
Breast cancer	286	0	9	2
Breast w	683	9	0	2
Diabetes	768	8	0	2
Heart-c	296	6	7	2
Heart-statlog	270	13	0	2
Iris	150	4	0	3
Kr_vs_kp	3196	0	36	2
Mushroom	5644	0	21	2
Promoters	106	0	57	2
Soybean	562	0	35	15
Wine	178	13	0	3
Zoo	101	1	16	7

(tabla 4.4).

Se ha realizado en cada caso un test de diferencia de medias con un nivel de significación de 0.05 donde la hipótesis nula establece que las medias son iguales. En base a los resultados obtenidos, las mejores soluciones aparecen marcadas en negrita.

Observando la tabla 4.2, si se comparan los resultados obtenidos por las distintas versiones del método propuesto, puede afirmarse que lvqVPSO es el que ofrece los mejores resultados brindando una solución óptima en 8 casos, seguido por las variantes somVPSO y lvqPSO con 4 casos y finalmente somPSO con sólo 2. Es decir que una inicialización supervisada combinada con un cúmulo de partículas de tamaño variable permite obtener los mejores resultados.

Con respecto al resto de los métodos puede verse que cAntMiner tiene una performance ligeramente mejor con 9 casos de los 13 analizados mientras que C4.5 y PART lo logran sólo en 5 y 6 casos respectivamente.

Un aspecto que también se observa en la tabla 4.2 es la dificultad del método lvqVPSO para obtener buenas soluciones cuando la proporción de ejemplos es baja con respecto a la cantidad de atributos. Esto ocurre especialmente en dos casos: la base 'Kr_vs_kp' que posee 3196 ejemplos y 36 atributos nominales y la base 'Promoters' con 106 ejemplos y 57 atributos nominales. En el primer caso, los mejores resultados los ofrecen C4.5 y PART mientras que lvqVPSO se encuentra un 6% por debajo pero en el segundo la diferencia es aún mayor. Esto tiene que ver con que en la base 'Promoters' cada atributo nominal posee 4 valores diferentes incrementando la dimensión de entrada de 57 a 228, lo que dificulta la inicialización de la red LVQ e impide realizar la búsqueda en los lugares adecuados.

Tabla 4.2: Resultados obtenidos al aplicar las cuatro variantes del método propuesto y con los métodos cAntMiner, C4.5 y PART. Para cada base de datos se indica la precisión promedio de cada método luego de 30 ejecuciones independientes.

BBDD	somPSO	lvqPSO	somVPSO	lvqVPSO	cAntMiner	C4.5	PART
Balance scale	0.7573 ±0.0190	0.7521 ±0.0143	0.7587 ±0.0132	0.7492 ±0.0148	0.7696 ±0.0105	0.7703 ±0.0068	**0.8181** ±0.0118
Breast cancer	**0.7186** ±0.0235	**0.7097** ±0.0171	**0.7062** ±0.0213	**0.7007** ±0.0251	**0.7088** ±0.0208	0.5802 ±0.0433	0.6626 ±0.0220
Breast-w	0.9519 ±0.0058	0.9572 ±0.0085	**0.9663** ±0.0064	**0.9622** ±0.0042	0.9485 ±0.0049	0.9536 ±0.0051	0.9552 ±0.0054
Diabetes	0.7358 ±0.0168	0.7240 ±0.0225	0.7247 ±0.0106	**0.7489** ±0.0174	**0.7409** ±0.0050	**0.7444** ±0.0120	0.7355 ±0.0117
Heart disease	0.7200 ±0.0200	**0.7700** ±0.0200	**0.7600** ±0.0200	**0.7700** ±0.0300	**0.7598** ±0.0164	0.7459 ±0.0184	**0.7634** ±0.0224
Heart statlog	0.7300 ±0.0200	0.7600 ±0.0200	0.7500 ±0.0200	0.7500 ±0.0200	0.7648 ±0.0141	**0.7807** ±0.0111	**0.7727** ±0.0188
Iris	**0.9417** ±0.0084	**0.9489** ±0.0269	0.9341 ±0.0261	**0.9424** ±0.0257	**0.9380** ±0.0122	**0.9467** ±0.0108	**0.9416** ±0.0102
Kr-vs-kp	0.9334 ±0.0083	0.9365 ±0.0037	0.9351 ±0.0042	0.9331 ±0.0032	0.9812 ±0.0010	**0.9929** ±0.0008	**0.9910** ±0.0013
Mushroom	0.9671 ±0.0076	0.9502 ±0.0018	0.9873 ±0.0047	**0.9901** ±0.0017	**0.9969** ±0.0001	0.9859 ±0.0238	**0.9937** ±0.0137
Promoters	0.6506 ±0.0328	0.6977 ±0.0261	0.6740 ±0.0237	0.6247 ±0.0281	**0.7917** ±0.0250	0.6997 ±0.0413	0.6897 ±0.0547
Soybean	0.8563 ±0.0146	0.8735 ±0.0156	0.8527 ±0.0139	0.8777 ±0.0341	**0.9066** ±0.0056	**0.8995** ±0.0135	0.8895 ±0.0097
Wine	0.8700 ±0.0300	0.8700 ±0.0100	0.8958 ±0.0266	**0.9083** ±0.0116	**0.9192** ±0.0124	0.8789 ±0.0138	0.8828 ±0.0178
Zoo	0.9200 ±0.0300	**0.9400** ±0.0200	**0.9300** ±0.0141	**0.9383** ±0.0279	**0.9408** ±0.0153	0.3393 ±0.0307	0.3417 ±0.0323

Tabla 4.3: Cardinalidad promedio del conjunto de reglas obtenido con las cuatro variantes del método propuesto y con los métodos cAntMiner, C4.5 y PART

BBDD	somPSO	lvqPSO	somVPSO	lvqVPSO	cAntMiner	C4.5	PART
Balance scale	**9.5200** ±0.3259	**9.5100** ±0.4748	**9.3300** ±0.3743	**9.3000** ±0.3528	14.6900 ±0.4483	41.5100 ±1.3371	38.9700 ±1.0764
Breast cancer	6.0600 ±0.3373	6.1400 ±0.4351	**5.4267** ±0.3535	**5.4800** ±0.2210	25.4100 ±1.0744	11.4367 ±1.0118	19.3867 ±1.1584
Breast-w	2.3700 ±0.1494	**2.3300** ±0.1418	**2.0200** ±0.0632	2.5700 ±0.2830	12.9000 ±0.5477	10.9033 ±0.7137	10.4267 ±0.5166
Diabetes	**4.0400** ±0.0843	**4.0600** ±0.0966	**4.0000** ±0.0000	**4.0000** ±0.0000	26.6400 ±0.8996	20.9333 ±2.1729	7.7233 ±0.4804
Heart disease	**3.1900** ±0.1800	4.9200 ±0.2800	3.9800 ±0.3800	4.0400 ±0.1300	16.0800 ±0.7406	24.0067 ±1.2600	19.2267 ±0.7529
Heart statlog	**4.2800** ±0.3500	**4.6400** ±0.3800	**4.5000** ±0.1400	**4.4600** ±0.0500	15.1200 ±0.6663	17.8467 ±1.0938	17.7367 ±0.6223
Iris	3.5750 ±0.2062	**3.4667** ±0.2517	**3.3556** ±0.2068	**3.4273** ±0.2005	8.3500 ±0.3808	4.6600 ±0.1380	3.7767 ±0.3036
Kr-vs-kp	**3.0000** ±0.0000	3.6500 ±0.3274	3.9667 ±0.0577	3.9667 ±0.0577	18.0900 ±0.9049	29.2000 ±0.5795	22.2733 ±0.6878
Mushroom	5.0500 ±0.0707	**4.7000** ±0.2828	**4.2900** ±0.1663	**4.3700** ±0.2003	22.6900 ±2.0328	18.6767 ±0.3014	11.2000 ±0.2560
Promoters	7.0133 ±0.2636	7.4250 ±0.3775	7.3571 ±0.3505	**5.5571** ±0.9880	11.5300 ±0.6430	16.8800 ±0.5623	7.2900 ±0.4254
Soybean	24.8567 ±0.5070	24.6857 ±0.7151	**22.4000** ±0.4243	**21.9500** ±0.2121	62.9700 ±2.1334	43.5667 ±0.3487	31.7900 ±0.5148
Wine	4.8000 ±0.3900	**3.7800** ±0.0900	**3.9500** ±0.0577	**3.9250** ±0.0957	6.6900 ±0.3900	7.8833 ±0.4742	5.5300 ±0.2020
Zoo	**6.9900** ±0.1000	**6.9500** ±0.0500	7.0500 ±0.0837	**7.0000** ±0.0000	10.9700 ±0.2214	8.3567 ±0.0679	7.6433 ±0.0568

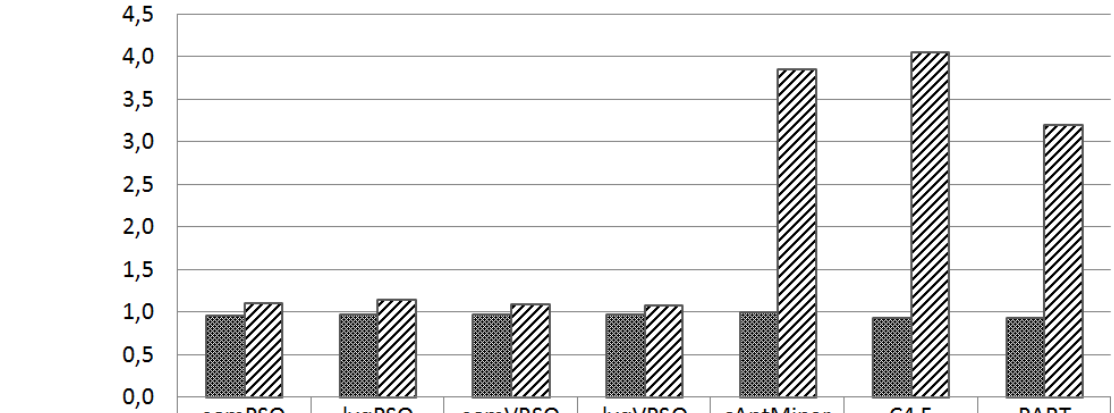
Un aspecto que sin duda debe destacarse y que queda en evidencia a través de los valores de la tabla 4.3 es la escasa cantidad de reglas que se generan con las cuatro variantes propuestas. Este es el objetivo central del método propuesto en esta tesis ya que se busca generar un conjunto reducido de reglas que sean fáciles de analizar convirtiéndose en una clara ayuda a la toma de decisiones.

Puede verse que en los 13 casos, el menor número de reglas se ha obtenido con una de las variantes del método propuesto y que aún en los casos en que lvqVPSO no ha sido la mejor, es la más cercana a la mejor solución encontrada en lo que se refiere al número de reglas.

Tabla 4.4: Longitud promedio del antecedente de cada conjunto de reglas obtenido luego de aplicar las cuatro variantes del método propuesto y los métodos cAntMiner, C4.5 y PART

BBDD	somPSO	lvqPSO	somVPSO	lvqVPSO	cAntMiner	C4.5	PART
Balance scale	2.1762 ±0.0630	2.1623 ±0.0700	2.3780 ±0.0452	2.3365 ±0.0745	**1.5900** ±0.0527	6.3419 ±0.0614	3.0965 ±0.0588
Breast cancer	**1.5725** ±0.0417	**1.5997** ±0.0442	**1.5815** ±0.0372	**1.5948** ±0.0630	1.8464 ±0.0578	2.1223 ±0.0913	2.0310 ±0.0646
Breast-w	2.6617 ±0.0910	2.6400 ±0.1550	3.0367 ±0.1119	2.7833 ±0.1012	**1.1622** ±0.0329	3.9226 ±0.1827	2.0917 ±0.0851
Diabetes	2.1592 ±0.1309	2.2597 ±0.1024	2.5250 ±0.1299	2.5250 ±0.1984	**1.1627** ±0.0238	5.5708 ±0.2982	1.9742 ±0.0965
Heart disease	**1.5700** ±0.0800	1.8800 ±0.1000	2.7000 ±0.1200	2.7000 ±0.1600	1.7455 ±0.0682	3.9114 ±0.0770	2.5603 ±0.0908
Heart statlog	1.6200 ±0.1300	1.8500 ±0.0700	2.5800 ±0.0200	2.5700 ±0.0600	**1.2927** ±0.0346	4.6512 ±0.1260	2.8984 ±0.1169
Iris	1.1917 ±0.0226	1.1806 ±0.0699	1.2222 ±0.0382	1.2015 ±0.0398	1.1793 ±0.0329	2.6117 ±0.0772	**0.9924** ±0.0208
Kr-vs-kp	2.3622 ±0.0611	2.4750 ±0.1236	2.4083 ±0.0946	2.4500 ±0.0901	**1.4986** ±0.1125	7.8003 ±0.0375	3.1285 ±0.0737
Mushroom	2.1001 ±0.2827	2.0738 ±0.0477	1.9708 ±0.0558	1.9562 ±0.0916	**1.0899** ±0.0294	2.6341 ±0.0371	1.2629 ±0.0219
Promoters	1.0977 ±0.0272	1.1131 ±0.0302	3.6342 ±0.0929	2.4911 ±0.2145	**1.0409** ±0.0527	2.2921 ±0.0282	**1.0049** ±0.0329
Soybean	5.9627 ±0.2650	3.1299 ±0.2543	3.9839 ±0.0462	4.2217 ±0.1504	3.3858 ±0.0537	6.0139 ±0.0435	**2.7056** ±0.0592
Wine	3.0500 ±0.4500	2.7800 ±0.2200	2.4763 ±0.2538	2.6033 ±0.1663	**1.0626** ±0.0340	3.1543 ±0.1163	1.5548 ±0.0653
Zoo	**1.5300** ±0.1000	1.6700 ±0.0500	2.2988 ±0.2608	1.8643 ±0.1291	1.6888 ±0.0601	4.0077 ±0.0235	**1.4748** ±0.0130

La figura 4.2 permite relacionar la precisión de cada método con respecto a la cantidad de reglas que genera. En dicha figura se ha calculado para ambos valores su relación con la mejor solución. En el caso de la precisión el promedio ha sido calculado dividiendo el valor obtenido por el método por la precisión máxima para la correspondiente base de datos mientras que para la cantidad de reglas se dividió el valor obtenido por el método por la cardinalidad mínima. Como puede observarse en la figura 4.2, las cuatro variantes del método propuesto tienen una cardinalidad parecida y son las de menor tamaño. En particular, lvqVPSO es, en promedio, un 7% superior a la solución de menor tamaño mientras que los restantes incrementan este valor entre un 284% y un 300%. Es decir que existe una gran diferencia entre la cardinalidad de la solución del método propuesto



	somPSO	lvqPSO	somVPSO	lvqVPSO	cAntMiner	C4.5	PART
Prec/MAX	0,9529	0,9631	0,9622	0,9632	0,9883	0,9218	0,9218
# Reglas/MIN	1,0951	1,1350	1,0791	1,0768	3,8447	4,0526	3,1931

Figura 4.2: Análisis comparativo de la precisión y la cardinalidad del conjunto de reglas. Los valores representados son los promedios de los cocientes entre la precisión y la cantidad de reglas de la solución hallada por cada método y la mejor solución encontrada para cada base.

y los restantes métodos analizados. Con respecto a la precisión cAntMiner es el método que ofrece los mejores resultados siendo un 2% superior al método propuesto pero como se dijo anteriormente, para lograr este 2% de mejora en la precisión utiliza en promedio un conjunto de reglas cuya cardinalidad casi cuadruplica la del método propuesto.

Haciendo el mismo análisis sobre la longitud del antecedente de las reglas obtenidas se observa en la figura 4.3 que cAntMiner es el método que posee las reglas más cortas siendo las variantes del método propuesto entre un 50% y un 80% más largas. Si bien estos valores parecen algo extremos, dado que las longitudes promedio son sumamente bajas, la diferencia mencionada es del orden de sólo una comparación adicional por regla en promedio.

En resumen, si se piensa en el conjunto de reglas final que constituyen el modelo ofrecido por cada método y se calcula la cantidad promedio de comparaciones que utiliza cada uno se observa en la figura 4.3 que las variantes del método propuesto requieren menos del 50% de las utilizadas por cAntMiner, aproximadamente el 15% de las necesarias para C4.5 y el 35% de las empleadas por PART.

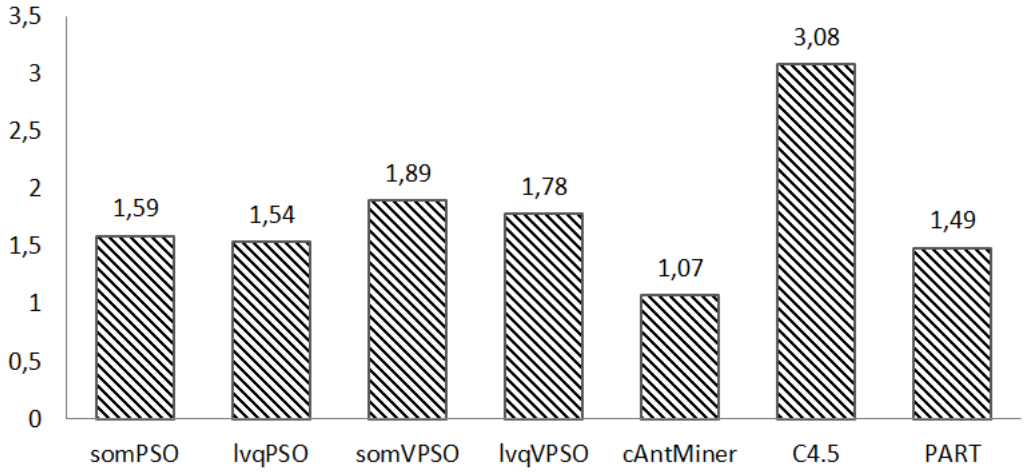


Figura 4.3: Análisis comparativo de longitud del antecedente de las reglas. Los valores representados son los promedios de los cocientes entre la longitud promedio de las reglas halladas por cada método y la de menor tamaño obtenida para cada base.

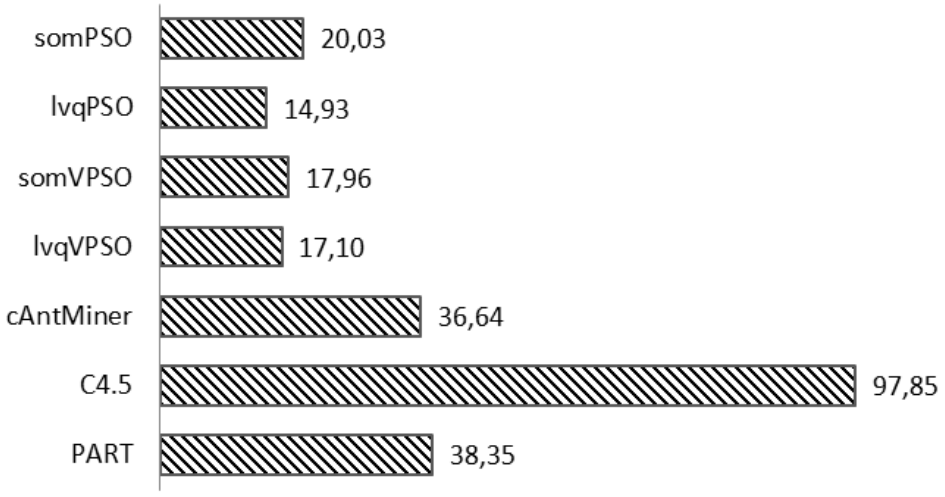


Figura 4.4: Simplicidad del modelo obtenido. Los valores corresponden al producto de la cantidad de reglas obtenidas y la longitud promedio del antecedente para cada uno de los métodos analizados

4.6 Reglas de clasificación aplicables a riesgo crediticio

Con el objetivo de mostrar un ejemplo de aplicación real del método propuesto, en esta sección se utilizará el método propuesto para hallar las reglas de clasificación correspondientes a información de crédito para consumo provenientes de dos entidades financieras de Ecuador. También se han agregado dos bases de datos pertenecientes al UCI Machine Learning Repository con información de operaciones de créditos realizados en Alemania y en Australia (Bache and Lichman, 2013).

A continuación se hará una breve introducción del tipo de crédito que interesa analizar en la bases de datos reales seguida por la descripción de la información registrada. Luego se detallarán los resultados obtenidos. Puede consultarse (Lanzarini et al., 2017) y (Jimbo et al., 2016) para más detalles.

4.6.1 Introducción

El siglo XXI presenta un aumento en el desarrollo y consumo de bienes. La ampliación de los servicios financieros en las economías emergentes es particularmente importante. La intermediación financiera proporciona una solución al consumo inmediato de bienes duraderos, ayudando a diferir el pago varios meses o años.

Este tipo de "democratización" del consumo plantea un desafío a las instituciones financieras. Muy lejos ha quedado el artículo (FitzPatrick, 1932) sobre predicción de bancarrota o el primer modelo z-score definido en (Altman, 1968). En ese momento, la principal preocupación de los bancos era clasificar a las empresas de acuerdo con su riesgo de crédito, ya que eran los principales clientes. Sin embargo, en las últimas décadas, ha habido un aumento en el crédito al consumo. La banca minorista se convirtió en una industria en crecimiento. No sólo ha habido un auge en la membresía de tarjetas de crédito, especialmente en las economías emergentes, sino también un aumento en los créditos de consumo pequeño.

Mientras que las solicitudes de préstamos hipotecarios, debido a su número comparativamente reducido de prestatarios, se pueden decidir a un ritmo más lento, las necesidades de préstamos de consumo necesitan procedimientos de decisión más rápidos. Incluso en algunos casos deben realizarse en forma instantánea. Los prestatarios quieren créditos pequeños para la compra de equipos para el hogar, un automóvil, un viaje, etc. Están deseosos de una respuesta rápida.

Se trata de un problema con dos puntos de vista. Por un lado se encuentran los prestatarios, quienes quieren recibir una rápida respuesta positiva a sus solicitudes. Por otro lado, las instituciones financieras quieren encontrar las reglas apropiadas para aprobar la solicitud de crédito sólo a los buenos prestatarios, es decir, aquellos que pagan sus

compromisos financieros.

Las instituciones financieras suelen pedir información exhaustiva sobre el cliente potencial: edad, estado civil, salario, otras deudas, tipo de trabajo, etc. Esta información se recopila para ser analizada, usando algún modelo de decisión. El resultado de este análisis es otorgar o rechazar el crédito.

El creciente número de solicitantes y datos plantea la necesidad de técnicas adecuadas que aborden la complejidad de este problema multidimensional. Resolución oportuna de las solicitudes de crédito es un elemento clave a la hora de decidir un método de puntuación de crédito.

Es en este contexto donde el método propuesto en esta tesis resulta de sumo interés por ofrecer un conjunto de reglas con una precisión aceptable y dos características claramente definidas desde el inicio: cardinalidad reducida y simplicidad de interpretación. La primera de estas características se relaciona directamente con la reducción de tiempo en el proceso de toma de decisiones de las instituciones financieras. La segunda tiene que ver con la necesidad de la empresa de explicar las decisiones tomadas. Si el conjunto de reglas es fácil de comprender, se obtendrá una mejora automática en la transparencia en las operaciones.

4.6.2 Casos de estudio

Tal como se explicó inicialmente, para medir la efectividad del método propuesto en esta área se probó su efectividad en dos bases de datos reales y dos datos financieros de crédito al consumidor del UCI Machine Learning Repository (Bache and Lichman, 2013).

Una de las bases de datos reales proviene de una importante entidad de ahorro y crédito de Ecuador con más de 20 años de trayectoria en el mercado interno.

Estos datos comprenden las operaciones de crédito entre 2011 y agosto de 2014, con los siguientes atributos: estado, fecha de aplicación, localidad, provincia, monto requerido, cantidad autorizada, propósito del crédito, efectivo, cuentas bancarias, inversiones, otros activos, pasivos y salario del solicitante, fecha de verificación de la información, fecha de autorización, fecha de aprobación o rechazo, efectivo, cuentas bancarias, inversiones, otros activos, pasivos y salario del socio de los solicitantes. En caso de que el solicitante sea una pequeña empresa, los datos solicitados son ingresos y gastos del negocio. La variable 'estado' corresponde a la situación del crédito. Las solicitudes pueden ser denegadas o aceptadas. En caso de ser aceptado, el estado se clasifica entre créditos debidamente reembolsados y aquellos con algún retraso en la devolución. A su vez, los créditos vencidos se clasifican según el manual de procedimientos de crédito entre aquellos con menos de 90 días de vencimiento y aquellos con más de 90 días de vencimiento (iniciación de acciones legales).

La otra base de datos real pertenece a la Cooperativa de Ahorro y Crédito, una institu-

Tabla 4.5: Bases de datos utilizadas para medir el desempeño del método propuesto e la obtención de reglas de riesgo crediticio

Base de Datos	#Ejemplos	#Atrib.Numéricos	#Atrib.Nominales	#Clases
Australiana	653	6	9	2
Alemana	1000	7	13	2
Cooperativa	22473	18	3	2
Bco.del Ecuador	36356	18	3	2

ción de ahorro mutuo de Ecuador, con las mismas variables descritas anteriormente, con operaciones entre 2011 y 2015.

La tabla 4.5 resume las características de las bases de datos utilizadas.

4.6.3 Resultados obtenidos

Los resultados de la aplicación de las cuatro variantes del método propuesto han sido comparados con los métodos C4.5 y PART. En esta oportunidad no se ha incluido cAntMiner por el excesivo tiempo requerido por este método para brindar el conjunto de reglas.

Se realizaron en total 30 ejecuciones independientes de cada método. Para PSO de población fija, se utilizó una red competitiva de 30 neuronas, mientras que para el caso de población variable, el tamaño comienza con 20 neuronas. PART se ejecutó con un factor de confianza de 0,3 para el árbol podado. Para los parámetros restantes se utilizaron los valores por defecto.

Las tablas 4.6, 4.7, 4.8 y 4.9 resumen los resultados obtenidos aplicando cada método en cada base de datos. En cada caso se consideró no sólo la exactitud de cobertura del conjunto de reglas, sino también la "transparencia" del modelo obtenido. Esta "transparencia" se refleja en el número promedio de reglas obtenidas y el número promedio de términos utilizados para formar el antecedente.

La característica más importante de estos resultados es que las cuatro variantes del método propuesto permiten obtener un conjunto de reglas con una cardinalidad significativamente baja, frente a los algoritmos C4.5 y PART.

Con respecto a la precisión los dos casos provenientes del repositorio han sido resueltos exitosamente mientras que los dos casos reales han sido clasificados con mayor precisión por parte de los algoritmos basados en partición C4.5 y PART. Sin embargo, tal como se observa claramente en la figura 4.5 esto es a expensas de un número mucho mayor de reglas. De hecho, la diferencia de precisión entre ambos tipos de métodos está dentro del rango de 1 a 3 puntos porcentuales. Debe destacarse que la exactitud de la clasificación

Tabla 4.6: Resultados obtenidos al aplicar las variantes del método propuesto y los métodos C4.5 y PART a la base de datos Australiana. En cada caso se indican la precisión y el desvío promedios de 30 ejecuciones independientes.

Metodo	Verdadero Negado	Verdadero Otorgado	Falso Negado	Falso Otorgado	Precision	# Reglas	Long. Anteced.
somPSO	0.4263 ±0.0209	0.4321 ±0.0172	0.1089 ±0.0116	0.0326 ±0.0068	**0.8584** ±0.0140	3.0100 ±0.0316	1.3525 ±0.0650
somVPSO	0.4247 ±0.0169	0.4289 ±0.0143	0.1080 ±0.0087	0.0383 ±0.0079	**0.8536** ±0.0126	3.0600 ±0.0699	1.7258 ±0.1084
lvqPSO	0.4238 ±0.0205	0.4403 ±0.0157	0.1027 ±0.0118	0.0330 ±0.0088	**0.8641** ±0.0130	3.0200 ±0.0421	1.3925 ±0.0569
lvqVPSO	0.4121 ±0.0115	0.4421 ±0.0099	0.1123 ±0.0111	0.0333 ±0.0071	**0.8543** ±0.0106	3.1100 ±0.1286	1.7258 ±0.1038
C4.5	0.3938 ±0.0088	0.4601 ±0.0086	0.0868 ±0.0084	0.0591 ±0.0084	**0.8540** ±0.0061	18.6066 ±2.1500	4.8638 ±0.2598
PART	0.3578 ±0.0155	0.3779 ±0.0413	0.1690 ±0.0413	0.0950 ±0.0155	0.7358 ±0.0340	33.4800 ±1.9028	2.4820 ±0.0829

basada en PSO es muy buena y comparable con los otros métodos. Sin embargo, en cuanto al número de reglas es entre 10 y 20 veces mayor en los métodos de partición.

En consecuencia, hay una especie de compensación entre la sencillez y la precisión. Dado que las reglas de crédito deben ser simples, para dar a los clientes una respuesta rápida (por ejemplo, en los créditos en línea de los consumidores) se considera que el método propuesto constituye una solución adecuada para este problema.

4.7 Conclusiones

En este capítulo se ha descripto un método original de extracción de reglas de clasificación capaz de operar con atributos cualitativos y cuantitativos el cual constituye el aporte central de esta tesis. Su funcionamiento se basa en una búsqueda realizada por una variante original de PSO inicializada a través de una red neuronal competitiva.

Este capítulo es un claro ejemplo de los puntos fundamentales que son necesarios definir para aplicar una técnica de optimización: representación a utilizar, mecanismo para medir el desempeño de las soluciones halladas, operadores para modificar y mejorar las soluciones existentes, criterios de terminación y valores adecuados de los parámetros que controlan la técnica. Cada uno de estos puntos requiere de un estudio minucioso de la situación.

El método propuesto ha sido aplicado a 15 bases de datos de repositorio y a dos casos

Tabla 4.7: Resultados obtenidos al aplicar las variantes del método propuesto y los métodos C4.5 y PART a la base de datos Alemana. En cada caso se indican la precisión y el desvío promedios de 30 ejecuciones independientes.

Metodo	Verdadero Negado	Verdadero Otorgado	Falso Negado	Falso Otorgado	Precision	# Reglas	Long. Anteced.
somPSO	0.1431 ±0.0253	0.5553 ±0.0436	0.1419 ±0.0365	0.1595 ±0.0204	**0.6984** ±0.0220	6.3444 ±1.6094	2.5248 ±0.3045
somVPSO	0.1230 ±0.0219	0.5789 ±0.0273	0.1188 ±0.0255	0.1788 ±0.0230	**0.7020** ±0.0139	6.3909 ±1.3888	2.4756 ±0.2248
lvqPSO	0.1406 ±0.0274	0.5416 ±0.0551	0.1626 ±0.0641	0.1548 ±0.0374	0.6823 ±0.0298	6.3555 ±1.6194	2.4468 ±0.3458
lvqVPSO	0.1286 ±0.0126	0.5696 ±0.0339	0.1287 ±0.0327	0.1730 ±0.0166	**0.6981** ±0.0231	6.5700 ±0.9129	2.5548 ±0.2268
C4.5	0.1223 ±0.0089	0.5882 ±0.0065	0.1117 ±0.0065	0.1776 ±0.0089	**0.7105** ±0.0072	85.0266 ±4.4466	5.6155 ±0.1678
PART	0.1386 ±0.0090	0.5554 ±0.0130	0.1445 ±0.0130	0.1613 ±0.0090	0.6940 ±0.0130	71.0600 ±1.8257	2.9978 ±0.0774

Tabla 4.8: Resultados obtenidos al aplicar las variantes del método propuesto y los métodos C4.5 y PART a la base de datos de la Cooperativa de Ahorro y Crédito de Ecuador. En cada caso se indican la precisión y el desvío promedios de 30 ejecuciones independientes.

Metodo	Verdadero Negado	Verdadero Otorgado	Falso Negado	Falso Otorgado	Precision	# Reglas	Long. Anteced.
somPSO	0.1847 ±0.0095	0.6066 ±0.0113	0.1069 ±0.0128	0.1016 ±0.0110	0.7913 ±0.0027	3.8250 ±0.3862	1.7102 ±0.0749
somVPSO	0.1972 ±0.0128	0.5940 ±0.0107	0.1191 ±0.0112	0.0896 ±0.0134	0.7912 ±0.0021	4.7000 ±0.8445	1.8697 ±0.2261
lvqPSO	0.1904 ±0.0129	0.6018 ±0.0147	0.1114 ±0.0125	0.0962 ±0.0106	0.7923 ±0.0038	4.0749 ±0.4787	1.6464 ±0.0845
lvqVPSO	0.1937 ±0.0183	0.6027 ±0.0223	0.1116 ±0.0204	0.0918 ±0.0166	0.7965 ±0.0040	4.7750 ±0.9394	1.7308 ±0.0840
C4.5	0.1785 ±0.0013	0.6320 ±0.0013	0.0819 ±0.0013	0.1074 ±0.0013	**0.8105** ±0.0011	114.2600 ±6.0543	9.6762 ±0.1143
PART	0.1825 ±0.0064	0.6228 ±0.0065	0.0910 ±0.0065	0.1035 ±0.0064	0.8054 ±0.0023	42.3566 ±2.1661	4.6956 ±0.0880

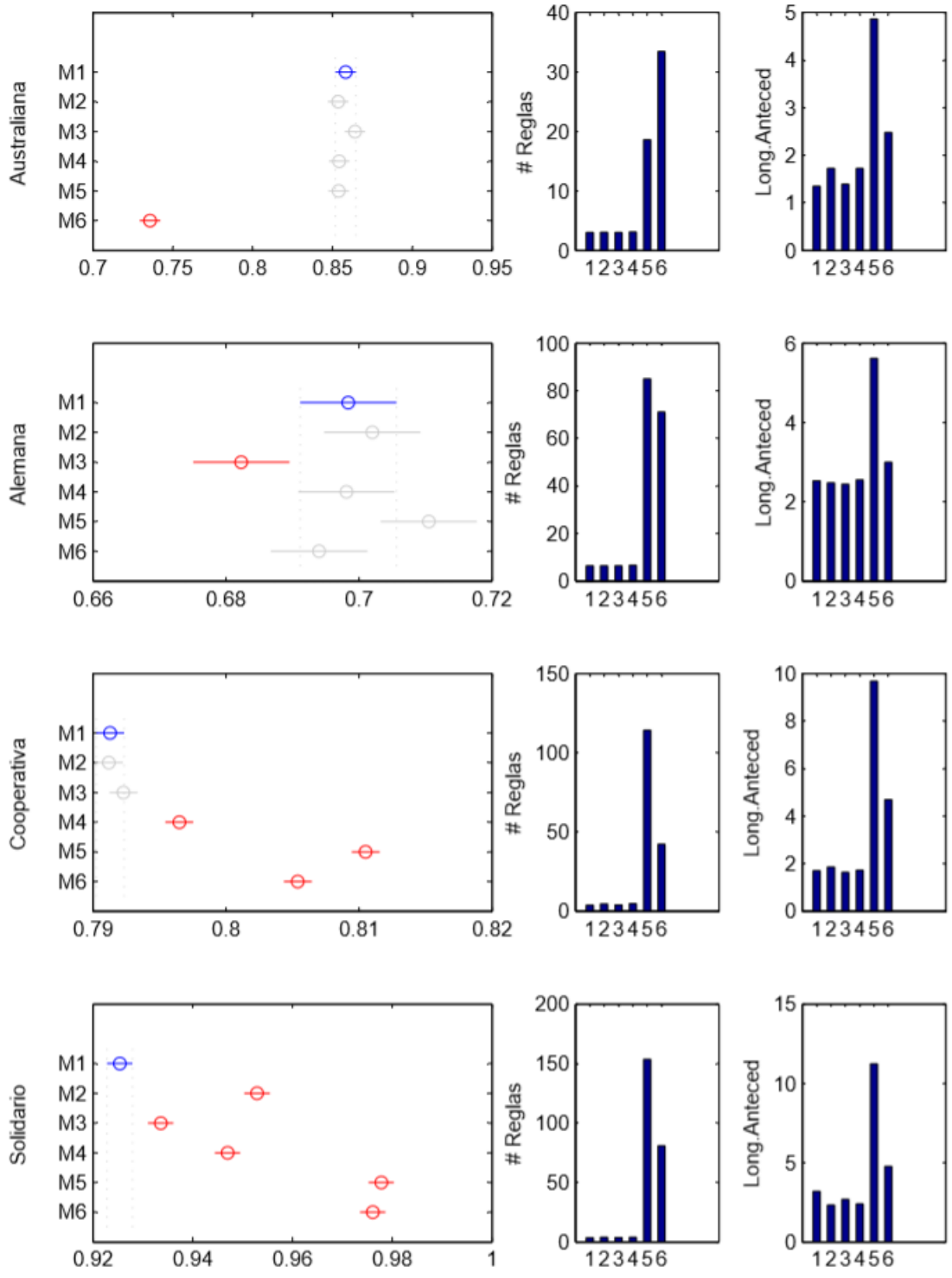


Figura 4.5: Comparación de los distintos conjuntos de reglas obtenidos por los métodos: somPSO (M1), somVPSO (M2), lvqPSO (M3), lvqVPSO (M4), C4.5 (M5) y PART (M6). Para cada base de datos se indica la precisión promedio de cada método (incluyendo el intervalo de confianza para cada media), la cardinalidad del conjunto de reglas y la longitud promedio del antecedente.

Tabla 4.9: Resultados obtenidos al aplicar las variantes del método propuesto y los métodos C4.5 y PART a la base de datos de un Bco. del Ecuador. En cada caso se indican la precisión y el desvío promedios de 30 ejecuciones independientes

Metodo	Verdadero Negado	Verdadero Otorgado	Falso Negado	Falso Otorgado	Precision	# Reglas	Long. Anteced.
somPSO	0.0567 ±0.0070	0.8687 ±0.0129	0.0401 ±0.0122	0.0342 ±0.0060	0.9254 ±0.0063	3.5666 ±0.2081	3.1905 ±0.4328
somVPSO	0.0621 ±0.0022	0.8908 ±0.0038	0.0177 ±0.0031	0.0288 ±0.0018	0.9529 ±0.0028	4.0142 ±0.3184	2.3164 ±0.2695
lvqPSO	0.0589 ±0.0064	0.8747 ±0.0199	0.0341 ±0.0192	0.0319 ±0.0061	0.9336 ±0.0139	3.6833 ±0.1471	2.6933 ±0.2149
lvqVPSO	0.0610 ±0.0042	0.8859 ±0.0047	0.0218 ±0.0051	0.0308 ±0.0035	0.9470 ±0.0069	3.9333 ±0.2658	2.3983 ±0.2092
C4.5	0.0761 ±0.0003	0.9016 ±0.0003	0.0072 ±0.0003	0.0148 ±0.0003	**0.9778** ±0.0003	153.5733 ±5.1686	11.2348 ±0.1564
PART	0.0748 ±0.0010	0.9012 ±0.0013	0.0077 ±0.0013	0.0161 ±0.0010	**0.9761** ±0.0007	80.9400 ±2.2033	4.7650 ±0.0687

reales con resultados satisfactorios. En todos los casos se ha cumplido con las tres características planteadas originalmente: precisión aceptable, baja cardinalidad y simplicidad de las reglas. todos los modelos obtenidos han sido los de menor tamaño y la precisión no ha sido inferior en promedio a un 2% de lo ofrecido por otros métodos.

Por lo anterior, se considera que el objetivo inicialmente propuesto ha sido alcanzado.

CAPITULO 5

CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

5.1 Conclusiones

La extracción de conocimiento a partir de la información disponible es un tema de sumo interés en la actualidad que está muy lejos de ser resuelto de manera automática. Aún no es posible desarrollar una única aplicación que sin importar cuál sea el origen de los datos pueda obtener el conocimiento deseado sin ningún tipo de intervención.

Las distintas etapas del conocido proceso de KDD no pueden ser recorridas secuencialmente sino que se requiere de una visión integral del problema a resolver para poder seleccionar las técnicas adecuadas que permitirán extraer a partir de los datos esos *"patrones válidos, novedosos, potencialmente útiles y en última instancia comprensibles ..."*tal como lo expresó Fayyad hace 20 años.

Algunos autores describen el proceso de KDD como formado sólo por tres etapas: la obtención de la vista minable que implica todo lo necesario para obtener los datos con los que se va a trabajar, la generación del modelo y la última pero no por eso menos importante etapa de interpretación y análisis de los resultados obtenidos. Sin importar el enfoque elegido, todos coinciden en la estrecha relación que existe entre ellas y en que antes de iniciar el proceso debe conocerse el tipo de respuesta que se espera obtener y obrar en consecuencia. Es con vista al modelo a generar que se realiza el preprocesamiento de la información. Es el modelo a construir el que condiciona las transformaciones que obligatoriamente deben realizarse sobre los datos de entrada. Luego de construido el modelo

existen distintas alternativas para medir y dar a conocer los resultados obtenidos.

Esta tesis tiene por objeto de estudio general a las reglas de clasificación. Pocos modelos son tan explicativos como las reglas. Incluso los árboles de clasificación son interpretados como reglas al recorrer sus ramas desde la raíz hasta las hojas. La necesidad de explicar las decisiones tomadas hacen de las reglas un modelo sumamente atractivo.

El aporte de esta tesis fue la definición de un método original capaz de obtener, a partir de atributos cualitativos y cuantitativos, un conjunto de reglas de clasificación con tres características fundamentales: precisión aceptable, baja cardinalidad y facilidad de interpretación. Para obtenerla se propuso combinar una variante original de PSO definida por la autora de esta tesis con una red neuronal competitiva.

Ninguna de las dos técnicas por separado es capaz de brindar el resultado adecuado. Las redes neuronales competitivas generan centroides con la misma dimensión que los ejemplos de entrada. No es sencillo elegir a partir de esta información la manera adecuada de construir las reglas. Por su parte, las técnicas de optimización dan buenos resultados en espacios continuos donde sea posible medir el desempeño de las partículas en posiciones cercanas. Esto no se cumple en el proceso de construcción de la regla ya que pequeñas modificaciones en el antecedente provocan grandes cambios en su valor de aptitud dificultando el proceso de búsqueda. La combinación de ambas técnicas lleva a inicializar el cúmulo de partículas en las zonas más prometedoras identificadas por la red neuronal reduciendo de esta forma la primera fase exploratoria y dedicando buena parte de la tarea a la explotación de zonas específicas del espacio de búsqueda.

A partir de los resultados obtenidos puede afirmarse que la mejor combinación es utilizar una red neuronal competitiva supervisada LVQ con un proceso de optimización mediante un cúmulo de partículas de tamaño variable. De esta forma se aprovecha mejor la clasificación existente en los ejemplos de entrada y se evita tener que dimensionar adecuadamente la población inicial. La unión de estas características ha permitido obtener resultados satisfactorios en las distintas situaciones de prueba.

En lo que respecta a la representación del problema, cada partícula contiene la información referida al antecedente de una única regla. Para operar con atributos cualitativos y cuantitativos fue preciso combinar una representación binaria y otra continua. La primera permite identificar cuáles son los atributos que forman parte del antecedente. En caso de ser cualitativos, la identificación se refiere al par (atributo,valor) mientras que para los cuantitativos sólo indica cuál o cuáles son los límites del intervalo que deben utilizarse. La parte continua tiene que ver precisamente con los límites de los intervalos que, si bien se calculan siempre, sólo participan del antecedente cuando la parte binaria así lo indica.

El método propuesto consiste básicamente en la aplicación reiterada de la técnica de optimización para obtener la mejor regla de clasificación para una clase seleccionada a

priori la cual generalmente es la clase mayoritaria. Por cada regla generada se retiran del conjunto de ejemplos de entrada los que se encuentren correctamente cubiertos. De esta forma, se garantiza la diversidad entre las reglas extraídas y se permite que las características aun no descubiertas sean ingresadas nuevamente al proceso de búsqueda.

Operar con un cúmulo capaz de modificar su tamaño durante la adaptación sin duda simplifica la generación del modelo. Los resultados obtenidos al aplicar el método propuesto sobre un conjunto de bases de datos de prueba permiten afirmar que el método lvqPSO obtiene un modelo más simple. En promedio utiliza aproximadamente el 40% de la cantidad de reglas que generan los otros métodos, con antecedentes formados por pocas condiciones y una precisión aceptable dada la simplicidad del modelo obtenido.

Por lo anterior se considera que los objetivos propuestos han sido cumplidos.

5.2 Líneas de trabajo futuras

Resta aún analizar algunos aspectos

- El método propuesto binariza los atributos cualitativos. Esto incrementa la longitud de la representación del antecedente dentro de cada partícula. Como consecuencia de esto, no sólo se incrementa el tiempo de cómputo sino que se requiere operar directamente sobre el mecanismo que simplifica las reglas cada vez que una partícula se mueve buscando evitar que reglas que se están formando adecuadamente sean calificadas con un valor de aptitud extremadamente bajo.

  Para resolver este problema podría pensarse en una representación alternativa para los atributos nominales. Tal vez algo más relacionado con el aspecto semántico del atributo sería de utilidad.

- El mecanismo de búsqueda realizado por el método propuesto, se encuentra controlado únicamente por PSO. Sin embargo podría ayudarse a este proceso si se tuviera información del soporte asociado a cada condición (o ítem) que podría participar del antecedente. Esta modificación representaría un cálculo adicional a realizar previo a la inicialización del cúmulo pero que seguramente ahorraría tiempo al ser utilizado para dirigir la búsqueda.

- Resultaría de interés analizar la paralelización del método propuesto ya que es una característica intrínseca de PSO y esta técnica se aplica reiteradamente hasta completar el conjunto de reglas completo.

- Finalmente, pensando en profundizar en el tema de reglas de clasificación aplicables a riesgo crediticio resulta de interés agregar al método propuesto la posibilidad de

operar con información macroeconómica. Este aspecto es muy importante ya que no sólo es el individuo sujeto de crédito quien posee las características que determinan la respuesta de la compañía sino también su entorno.

Esto implica que la existencia de dos niveles de información, una específica que cada persona y otra de la situación macroeconómica en la que esa persona se encuentra inmersa y que es común a varias otras. Las reglas a construir deberían surgir del análisis en paralelo de ambas situaciones. El problema actualmente se encuentra en análisis y se considera un desafío interesante para el método aquí propuesto.

ANEXO A

GENERACIÓN DE REGLAS DE ASOCIACIÓN USANDO FUZZY SOM

El análisis de documentos de textos cortos tiene múltiples aplicaciones que van desde la identificación de e-mails como spam hasta el análisis de sentimiento u opinión dentro de las redes sociales.

En particular, en este anexo se propone utilizar técnicas de minería de datos para analizar los e-mails correspondientes a cursos realizados a través de una plataforma de educación a distancia. Con este tipo de análisis se busca determinar los grupos de palabras relevantes que permitan establecer los temas de comunicación de interés. Si bien esta nueva información puede tener distintas aplicaciones, todas ellas implican una mejora en la atención de los alumnos. El método propuesto ha sido aplicado a los e-mails del proyecto PACENI (Proyecto de Apoyo para el Mejoramiento de la Enseñanza en Primer Año de Carreras de Grado de Ciencias Exactas y Naturales, Ciencias Económicas e Informática) con resultados satisfactorios.

A.1 Introducción

Las plataformas de educación a distancia constituyen un entorno de aprendizaje a través del cual docentes y alumnos interactúan realizando distinto tipo de actividades. En este contexto, la mensajería interna es el mecanismo más utilizado y por tal motivo resulta de interés el estudio de técnicas que permitan analizar y modelizar la información compartida

a través de este medio.

Por ejemplo, sería relevante conocer los temas que motivan las consultas más habituales por parte de los alumnos. Esto podría tener distintos usos:

- Permitiría detectar falencias en la información suministrada, por ejemplo, falta de información con respecto a las fechas de examen o necesidad de refuerzo en algún tema porque el material teórico suministrado no ha sido suficientemente claro.
- Organizar automáticamente los e-mails para mejorar la atención de los alumnos.
- Identificar automáticamente los temas centrales de discusión con el objetivo de mejorar la toma de decisiones.

Un e-mail posee una fecha, un conjunto de direcciones, un asunto y un cuerpo. Este último, si bien puede contener distinto tipo de información, básicamente está compuesto por texto y por lo tanto es posible utilizar técnicas de minería de textos para analizarlos.

La Minería de Textos pertenece a la Minería de Datos y tiene por objetivo central la extracción de información de alta calidad a partir de documentos. En la mayoría de los casos el objetivo central es la determinación de la relevancia del documento en función de una consulta realizada previamente. Esto posibilita su clasificación y acceso de una forma automática más eficiente.

Sin embargo, la extracción de información a partir de e-mails requiere de algunas consideraciones especiales por tratarse, en general, de textos cortos con una redacción bastante abreviada. Esto hace que pierdan relevancia algunas métricas utilizadas tales como la longitud del texto o la frecuencia con la que una palabra aparece dentro de él.

Este anexo ejemplifica la manera de analizar mensajes que fueron enviados dentro de una plataforma de educación a distancia y corresponden a las actividades realizadas dentro del Programa de Tutorías (PACENI). Este programa es impulsado por el Ministerio de Educación y está orientado a reducir la cantidad de alumnos que abandonan sus estudios universitarios durante el primer año. La UNLP puso en marcha este programa en el ciclo 2009. Por este medio, alumnos de primer año son acompañados por tutores, alumnos de postgrado o de grado avanzados, quienes les ayudan a superar las dificultades iniciales de la vida universitaria.

A.2 Método propuesto

Para el procesamiento se utilizó un diccionario construido automáticamente a partir de la reducción de cada palabra a su raíz (stemming) y su posterior selección. Utilizando el

diccionario se representó cada mail como un vector numérico y se los utilizó para entrenar un mapa auto-organizativo difuso o FSOM (Fuzzy Self Organizing Map).

Un detalle importante es que la red FSOM realiza un clustering difuso del espacio de entrada a partir de un entrenamiento competitivo similar al utilizado por SOM. Sin embargo, a diferencia de SOM que sólo aprende los centros de los clusters, FSOM aprende ambos, los centros y las desviaciones alrededor de los centros. Por tal motivo, los pesos de los arcos de la i-ésima neurona competitiva en FSOM se representan a través de un vector $S_i = (s_{1i}, s_{2i}, \ldots, s_{Di})$ siendo D la dimensión del vector de entrada y s_{ji} una tupla representada por los tres parámetros de la j-ésima función gaussiana de la forma $s_{ji} = (w_{ji}, \sigma 1_{ji}, \sigma 2_{ji})$. Además, en FSOM, cada vector de entrada no se encuentra asociado a una única neurona ganadora sino que posee un grado de pertenencia para cada una de ellas.

Finalmente, deben descartarse las palabras menos significativas, es decir, aquellas que no posean un peso propio suficiente como para lograr ser representadas claramente por un subconjunto acotado de neuronas. Esto ocurre con aquellas palabras que se combinan con muchos términos o que aparecen muy pocas veces. En cualquiera de los dos casos se trata de términos con poca información ya que en el primer caso no determinan el tema del cual se habla y en el segundo no tienen el suficiente soporte (cantidad de apariciones) como para considerarlas significativas. La red FSOM entrenada permite detectar estas palabras porque son aquellas que no superan un umbral mínimo en ninguno de los vectores asociados a las neuronas competitivas. Por lo tanto, se eliminaron las palabras irrelevantes utilizando dicho umbral.

A.3 Resultados obtenidos

El método descripto en la sección 4 fue aplicado a los 2995 e-mails correspondiente al Proyecto de Tutoría (PACENI) durante el período abril a noviembre de 2009.

El diccionario inicial estuvo formado por 2935 raíces de términos de las cuales, a través de un análisis estadístico se tomaron 287.

Se utilizó una red SOM de 13x13 neuronas competitivas con 4 vecinos por neuronas. Luego de entrenar la red, se eliminaron de la matriz W todos aquellos pesos que no fueran significativos utilizando un umbral de 0.85. A partir de los vectores de pesos de cada neurona competitiva se determinaron las combinaciones de términos que permiten agrupar los e-mails. Para medir su relevancia se los utilizó para formar las reglas de asociación correspondientes considerando todas las combinaciones posibles. Se asoció a la combinación el máximo valor de se obtiene al multiplicar sus valores de soporte y confianza. Luego de realizar 50 entrenamientos independientes de la red neuronal, las asociaciones que más

veces aparecieron fueron las siguientes:

- ('analitico', 'academico', 'aprobado', 'consejo', 'extender', 'entrega', 'certific')
- ('beta', 'aula', 'inscripción', 'inglés)'
- ('incluido', 'beca', 'ministerio', 'tics', 'encontrar', 'http', 'inscripción')

Estas combinaciones aparecen en distinto orden pero siempre dentro de las 20 primeras mejores posicionadas. Esto determina su importancia en el conjunto de e-mails. Otra característica observada luego de las distintas pruebas es que la red neuronal permite descartar entre 100 y 120 términos a través de la función umbral. Esto reduce considerablemente el tiempo de obtención de las reglas de asociación a medir.

Los detalles de la arquitectura FSOM utilizada así como la manera de identificar y seleccionar los ítems frecuentes se encuentra descripta en (Lanzarini et al., 2011b).

A.4 Conclusiones

Este anexo ejemplifica la manera de procesar un conjunto de documentos cortos para extraer términos comunes. En el caso de los datos de las Tutorías PACENI se obtuvieron distintos conjuntos de items frecuentes representativos de las consultas más comunes por parte de los estudiantes. Esto permite conocer sus temas de interés los cuales están fuertemente vinculados a falta de información ya sea porque la manera en que fue publicada no resultó clara para los alumnos o porque se omitió hacerlo. En cualquier caso, el método propuesto en este anexo es una buena herramienta para recomendar las correcciones que deben llevarse a cabo.

Un aspecto importante para la obtención de buenas combinaciones de términos es el armado del diccionario inicial. En este caso sólo se ha hecho un preprocesamiento estadístico con la intención de que el diccionario fuese generado de la manera más automática posible. Es posible mejorar esta etapa agregando información adicional ingresada manualmente.

ANEXO B

Reconocimiento de personas por la imagen de su rostro

La transmisión de información en formato de imagen ha ganado popularidad en los últimos años. Disponer de mecanismos que faciliten su almacenamiento y procesamiento son de sumo interés. Existen distintas transformaciones que pueden aplicarse sobre una imagen para facilitar su procesamiento. Una posibilidad es transformarla en un conjunto de vectores numéricos donde cada uno contiene información de las partes más representativas de la imagen.

Los aspectos más relevantes suelen estar relacionados con el almacenamiento y el reconocimiento de objetos presentes en una imagen. En el primer caso, la optimización mediante cúmulos de partículas o PSO puede ser de utilidad. En (Maulini and Lanzarini, 2012) se ejemplifica su uso en la selección de los vectores SIFT más representativos de una imagen. Esto reduce los falsos positivos a la hora de comparar dos imagenes de un mismo objeto y además reduce el almacenamiento necesario para guardarla. El reconocimiento de objetos es un tema más complejo generalmente resuelto a través de umbrales.

Este anexo muestra un método original que extiende la capacidad de una red neuronal SOM difusa para reconocer una persona por la imagen de su rostro. Para ello se utilizan las funciones de distribución de probabilidad de cada neurona competitiva para reconocer una cantidad finita de sujetos. Es un enfoque interesante ya que el elemento a reconocer no surge de un único vector numérico sino que requiere un conjunto de vectores para decidir de quien se trata.

B.1 Introducción

El reconocimiento de rostros es una técnica biométrica muy utilizada en diversas áreas como seguridad y control de accesos, medicina forense y controles policiales. La identificación de una persona por la imagen de su rostro es un problema difícil de resolver automáticamente debido a los cambios que distintos factores, como la expresión facial, el envejecimiento e incluso la iluminación, producen en la imagen.

La identificación de una persona por la imagen de su rostro es un proceso que consiste en comparar una imagen del sujeto de interés con un conjunto de imágenes almacenadas previamente en una base de datos. Para brindar más flexibilidad al reconocedor, la base suele contener varias imágenes de una misma persona buscando modelizar las distintas situaciones que pueden presentarse al momento de capturar una nueva imagen. Esto incluye expresiones faciales, cambios de posición de la cabeza, cambios de escala, etc.

La información a buscar en la base de imágenes no es directamente la imagen original sino una caracterización de ella. En particular, se trabajó con los descriptores generadores a partir del método SIFT, definido por Lowe en (Lowe, 2004) por considerarlos invariantes a la escala, rotación, punto de vista, oclusión parcial e iluminación. Estos descriptores convierten cada imagen en un conjunto de vectores numéricos. Las imágenes de la base se encuentran almacenadas con esta representación. Sin embargo, el reconocimiento utilizando vectores SIFT presenta la desventaja de generar falsos positivos. Esto lleva, en ocasiones, a un reconocimiento incorrecto. Para resolver este aspecto, en Maulini and Lanzarini (2012) se propone utilizar una variante de PSO definida en Lanzarini et al. (2011a). La técnica de optimización es utilizada para seleccionar los vectores SIFT más representativos logrando no sólo una reducción en los falsos positivos sino también en el tiempo de cómputo requerido para el procesamiento y en el almacenamiento de la base de imágenes.

B.2 Método propuesto

Las imágenes de la base, representadas a través de sus respectivos conjuntos de descriptores SIFT dan lugar a los datos que se utilizarán para entrenar una Fuzzy SOM neural network. Es decir que el conjunto de entrenamiento está formado por vectores numéricos de dimensión 128 correspondiente a las distintas imágenes de todos los sujetos de la base. Esto implica que para un descriptor SIFT dado se conoce a que imagen corresponde. Cada imagen está asociada a un sujeto y por lo tanto, también lo está cada descriptor SIFT.

El método utilizado para entrenar la red neuronal es el mismo que se utilizó en el anexo A. Una vez finalizado el entrenamiento, cada neurona de la red puede representar a más de una persona. Para determinarlo, se ingresan los datos de entrenamiento y se calcula

para cada uno de ellos, los grados de pertenencia a cada neurona. Luego, cada vector de entrenamiento será asignado en forma proporcional a su grado de pertenencia a las k neuronas competitivas más representativas.

Dado que cada vector está asociado a una persona, cada neurona competitiva llevará asociada una lista de la proporción de descriptores SIFT que posee de cada sujeto. El cálculo se efectúa de la siguiente forma:

- Sean L el número de personas en la base.
- Sean T el número total de descriptores SIFT correspondientes a todas las imágenes de la base. Cada descriptor pertenece a una de las imágenes de una persona.
- Sean N la cantidad de neuronas de la red. Para cada descriptor, d_j con $j = 1..T$ se calculan sus grados de pertenencia a cada neurona de la red y se ordenan las neuronas según estos grados en forma decreciente . Sean $n_1, n_2, \ldots, n_k$ las k neuronas competitivas con los grados de pertenencia más altos de manera que

  $$G(d_j, n_1) > G(d_j, n_2) > \ldots > G(d_j, n_k)$$

  y además

  $$G(d_j, n_k) > G(d_j, n_r) \;\; con\; r = k+1 : N$$

- Las k neuronas competitivas con los grados de pertenencia más altos compartirán la representación del descriptor SIFT d_j en forma proporcional como se indica a continuación:

  (B.1) $$Prop(n_i, d_j) = \frac{exp^{-i^2/2}}{\sum_{x=1}^{k} exp^{-x^2/2}} \;\; i = 1 : k$$

  Nótese que el valor del grado de pertenencia no interviene en la ecuación B.1. Sólo se tiene en cuenta el orden establecido entre las neuronas. El denominador de la ecuación (B.1) normaliza las proporciones al rango [0.1] de manera que si se suman los valores de $Prop(n_i, d_j)$ para todos los descriptos SIFT d_j correspondiente a una misma persona, s_l, se obtendrá un valor equivalente a la cantidad total de estos vectores SIFT que se consideran representados por la neurona n_i. Este valor no es entero ya que corresponde a la suma de las proporciones y será considerado el grado de reconocimiento de la neurona n_i por la persona s_l.

- La correspondencia entre el descriptor SIFT y la persona a la cual corresponde se indica de la siguiente forma

  (B.2) $$Q(s_k, d_j) = \begin{cases} 1 & if\ d_j\ corresponds\ to\ person\ s_k \\ 0 & if\ not \end{cases}$$

Luego, la frecuencia relativa de las personas reconocidas por la i-ésima neurona puede expresarse de la siguiente forma

$$P(s_k|n_i) = \frac{\sum_{t=1}^{T} Prop(n_i,d_t).Q(s_k,d_t)}{\sum_{t=1}^{T} Prop(n_i,d_t)} \qquad \forall s_k \in L \tag{B.3}$$

Nótese que una vez calculadas las probabilidades condicionales indicadas en (B.3), la red está en condiciones de estimar la similitud entre una imagen nueva y las almacenadas en la base. A continuación se describe este proceso.

B.3 Mecanismo de identificación

El mecanismo de identificación es probabilístico. Consiste en ingresar a la red entrenada todos los descriptores SIFT correspondientes a la imagen del sujeto que se desea reconocer. Para cada uno de ellos se calcula la neurona ganadora. Recuerde que cada neurona representa, generlamente, a varias personas.

La persona identificada por la red es la que cumple con la expresión (B.4)

$$s_k \in L \Longleftrightarrow s_k = arg\ max_r(P(s_r)) \ \ \forall r/s_r \in L \tag{B.4}$$

donde, aplicando el teorema de la probabilidad total $P(s_k)$ se calcula de la siguiente forma

$$P(s_k) = \sum_{i=1}^{N} PN(n_i).P(s_k|n_i) \tag{B.5}$$

donde

$$PN(n_i) = \frac{\sum_{t'=1}^{T'} Prop(n_i,d'_{t'})}{T'} \tag{B.6}$$

siendo T' la cantidad de descriptores SIFT que representan la imagen de entrada.

La ecuación (B.6) calcula la probabilidad (proporción) de ganar que tiene la i-ésima neurona al ingresar los T' descriptores SIFT de la imagen. Dado que cada neurona puede determinar la probabilidad con la que reconoce a los diferentes candidatos, la ecuación(B.5) permite obtener la respuesta de la red para cada persona a reconocer. Como indica la expresión (B.4), se seleccionará al sujeto con la probabilidad más alta.

B.4 Resultados obtenidos

Para medir la efectividad del método propuesto se utilizaron dos bases de datos obtenidas de (Grgic and Delac, 2017). La primera es la base de rostros YALE que contiene 165 imágenes de 15 sujetos distintos con 11 imágenes por persona. Cada imagen tiene una resolución de 320x243 pixels. La segunda base de rostros es la base $AT\&T$ que contiene 400 imágenes de 40 personas con 10 imágenes por individuo. El tamaño de cada imagen es de 112x92 pixels.

Los resultados obtenidos fueron comparados con los siguientes tres métodos:

- *SIFT*: Utiliza el criterio de reconocimiento establecido en (Lowe, 2004) donde para cada imagen se calcula el total de coincidencias con cada uno de los sujetos almacenados en la base y se selecciona el que posea el valor mayor. Este proceso implica comparar cada descriptor de la imagen de entrada con el conjunto de descriptores de cada una de las imágenes de la base. Se considerará que dicho descriptor se corresponde (hay coincidencia) con la imagen si la distancia al descriptor más cercano del conjunto es inferior al 50% de la distancia al segundo. Este porcentaje es un parámetro del algoritmo y fue seleccionado por ser el recomendado en (Lowe, 2004).

- *SIFT+PSO*: es una mejora del anterior definida en (Maulini and Lanzarini, 2012) y utiliza una técnica de optimización para seleccionar los descriptores SIFT más representativos. El proceso de reconocimiento es el mismo que en método *SIFT* con la ventaja de que opera sobre un número menor de descriptores (ver figura B.1).

- *ProbSOM* : Método definido en (Estrebou et al., 2010) que utiliza una estrategia probabilística para efectuar el reconocimiento pero al utilizar una clasificación no difusa, en todo el proceso los descriptores se encuentran asociados a un único agrupamiento (centroide).

- *Fuzzy ProbSOM*: Método descripto en este anexo y publicado en (Lanzarini et al., 2013).

Dado que la tasa de acierto varía en función de la cantidad de imágenes utilizadas para construir el modelo se realizaron en los 4 casos y para ambas bases, 9 cortes para formar el conjunto de imágenes de entrenamiento; estos son: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% y 90%. Para cada uno de ellos, se realizaron 30 ejecuciones independientes de cada método utilizando en cada caso la misma selección de imágenes para entrenar. La figura B.2 muestra, para cada uno de los métodos, la tasa de acierto promedio (de las 30 corridas) para cada corte utilizando la base YALE. Como puede observarse, el método propuesto es el que alcanza los mejores resultados. Dado que en todos los casos las desviaciones fueron equivalentes y de que se trata de una muestra grande, se realizó un test ANOVA junto con una prueba de Tukey, utilizando un nivel de significación de 0.05, pudiendo determinar que

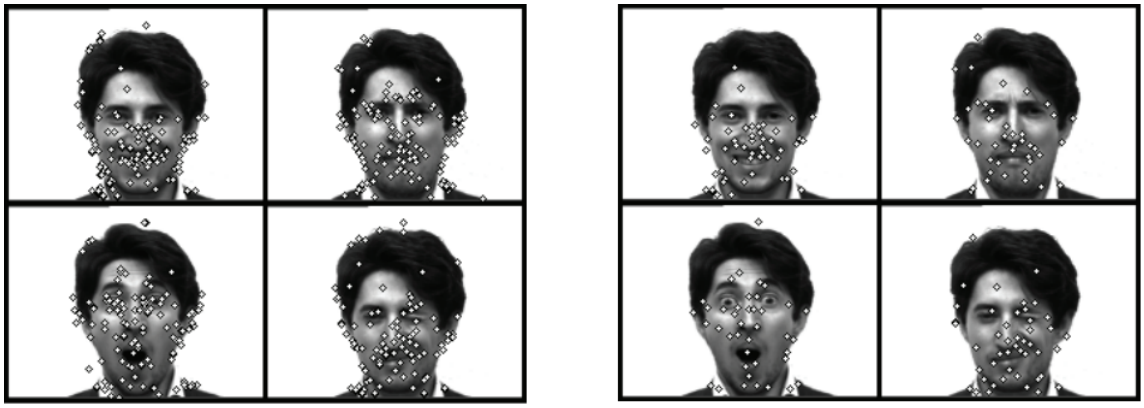


Figura B.1: Descriptores SIFT de una persona de la base YALE según a) el método de (Lowe, 2004), b) el método definido en (Lanzarini et al., 2013)

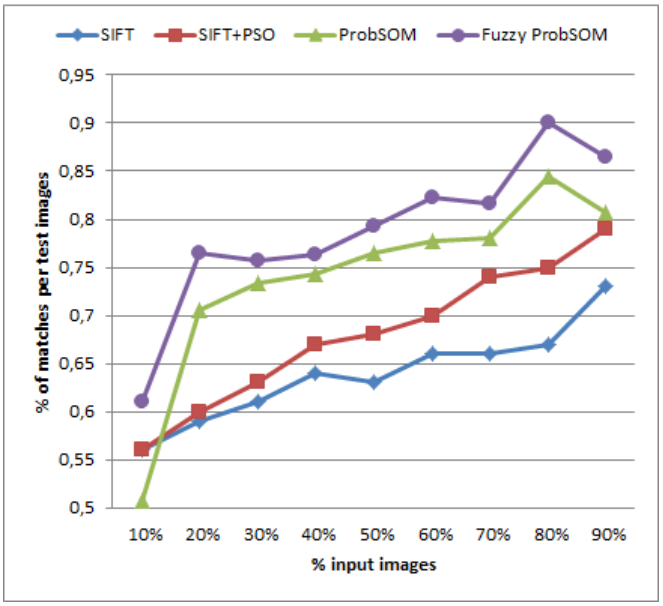


Figura B.2: Tasa de acierto de cada método para la base YALE. Cada valor indicado para cada porcentaje corresponde al promedio de 30 ejecuciones independientes.

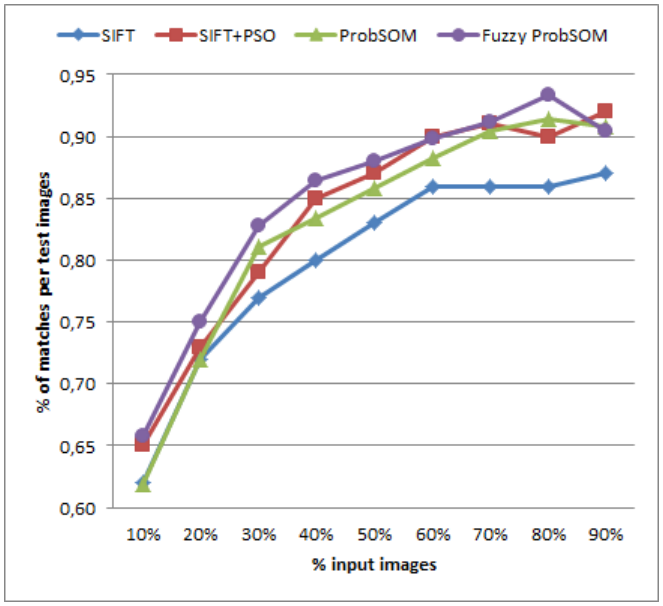


Figura B.3: Tasa de acierto de cada método para la base AT&T. Cada valor indicado para cada porcentaje corresponde al promedio de 30 ejecuciones independientes.

existen diferencias significativas en todos los casos salvo en el porcentaje más bajo donde los métodos SIFT y SIFT+PSO resultaron equivalentes.

Con respecto a la base AT&T se efectuaron las mismas pruebas llegando a la conclusión de que salvo el método SIFT, el resto ofrece resultados equivalentes, es decir que no se observan diferencias significativas. La figura B.3 ilustra los resultados obtenidos. En este caso es importante destacar que disponer de un modelo basado en una red neuronal permite realizar la clasificación en un tiempo computacional menor pese a la reducción en la cantidad de descriptores obtenida con PSO en Maulini and Lanzarini (2012).

B.5 Conclusiones

La combinación de una red neuronal competitiva difusa con un criterio de decisión probabilístico, tal como fuera publicado en (Lanzarini et al., 2013), al ser aplicada a dos bases de datos de prueba ha permitido obtener tasas de acierto superiores al método SIFT convencional definido por Lowe en Lowe (2004).

Las bases seleccionadas son muy diferentes; mientras que AT&T posee rostros similares, YALE incluye imágenes de un mismo individuo con cambios significativos (ej: diferentes expresiones o uso de anteojos) como se observa en la figura B.1. Estos factores dificultan el reconocimiento y por tal motivo las tasas de acierto sobre la base YALE son inferiores a las de AT&T. Como puede verse en la figura B.2, el método propuesto es el más robusto al

ofrecer los mejores resultados frente a grandes cambios en los vectores SIFT.

Como debilidad del método presentado debe destacarse su imposibilidad de operar sobre una clase de rechazo. Esta situación se repite también en los otros métodos. Nótese que en todos los casos, los cuatro métodos buscarán sobre la base al sujeto más parecido a la imagen de entrada. El problema se presenta cuando el sujeto a identificar no pertenece a la base. En este caso, el uso de un umbral para tomar una decisión no garantiza la obtención de una respuesta correcta. Por tal motivo se está incorporando a este modelo una segunda red neuronal que ayude en el proceso de reconocimiento.

Referencias

Aggarwal, C. (2015). *Data Mining: The Textbook*. Springer International Publishing.

Agrawal, R., Imieliński, T., and Swami, A. (1993a). Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216.

Agrawal, R., Imielinsky, T., and Swami, A. (1993b). Ais. Technical report, IBM Almaden Research Center, San Jose, California.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Al-Magaleh, B. and Shahbazkia, H. (2012). A genetic algorithm for discovering rules in data mining. *International Journal of Computer Applications (IJCA)*, 41(18):40–44.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23(4):589–609.

Azevedo, P. J. and Jorge, A. M. (2007). *Comparing Rule Measures for Predictive Association Rules*, pages 510–517. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bache, K. and Lichman, M. (2013). Uci-machine learning repository.

Baldino, G. and Lanzarini, L. (2016). Análisis del avance académico de alumnos universitarios. un estudio comparativo entre la un-frlp y la unlp. In *XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016)*, pages 589–596.

Bayardo, R. (1998). Efficiently mining long patterns from databases. *SIGMOD Rec.*, 27(2):85–93.

Charnelli, M. E., Lanzarini, L., Baldino, G., and Díaz, J. (2015). Determining the profiles of young people from buenos aires with a tendency to pursue computer science studies. In Feierherd, G., Pesado, P., and Spositto, O., editors,*Computer Science & Technology Series - Series - XX Argentine Congress of Computer Science*, chapter XII Information Technology Applied to Education Workshop, pages 1155 –1163. Red UNCI.

Chihli, H. and Huang, L. (2010). Extracting rules from optimal clusters of self-organizing maps. In *Computer Modeling and Simulation, 2010. ICCMS '10. Second International Conference on*, volume 1, pages 382–386.

Clerc, M. (2013). *Particle Swarm Optimization*. ISTE. Wiley.

Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation.*, 6(1):58–73.

De Jong, K. A. and Spears, W. M. (1991). Learning concept classification rules using genetic algorithms. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'91, pages 651–656, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ding An, C., Wei, C., YiFan, W., and Lain Jinn, H. (2001). Rules generation from the decision tree. *Journal of Information Science and Engineering*, 17(1):325–339.

Estrebou, C., Lanzarini, L., and Hasperué, W. (2010). Voice recognition based on probabilistic som. In *Conferencia Latinoamericana de Informática. CLEI 2010*.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996a). The kdd process for extracting useful knowledge from volumes of data. *Commun. ACM*, 39(11):27–34.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996b). Advances in knowledge discovery and data mining. chapter From Data Mining to Knowledge Discovery: An Overview, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA.

Fidelis, M. V., Lopes, H. S., and Freitas, A. A. (2000). Discovering comprehensible classification rules with a genetic algorithm. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, volume 1, pages 805–810 vol.1.

FitzPatrick, P. (1932). *A Comparison of the Ratios of Successful Industrial Enterprises with Those of Failed Companies*.

Formia, S. and Lanzarini, L. (2013). Caracterización de la deserción universitaria en la unrn utilizando minería de datos. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología (TE&ET)*, (11):92–98.

Frank, E. and Witten, I. H. (1998a). Generating accurate rule sets without global optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 144–151, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Frank, E. and Witten, I. H. (1998b). Generating accurate rule sets without global optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 144–151, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Fred, A. L. N. and Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(6):835–850.

Freitas, A. A. (2003). *A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery*, pages 819–845. Springer Berlin Heidelberg, Berlin, Heidelberg.

Futuyma, D. (2006). *Evolutionary Biology*. Sinauer Associates.

Gandhi, K., Karnan, M., and Kannan, S. (2010). Classification rule construction using particle swarm optimization algorithm for breast cancer data sets. In*2010 International Conference on Signal Acquisition and Processing*, pages 233–237.

Glover, F. (1989). Tabu search - part 1. *ORSA Journal on Computing*, 1(3):190–206.

Grgic, M. and Delac, K. (2017). Face recognition. `http://www.face-rec.org/databases/`. [Ultimo acceso 28/01/2017].

Hasperué, W. and Lanzarini, L. (2005). Dynamic self-organizing maps. a new strategy to upgrade topology preservation. In *XXXI Conferencia Latinoamericana de Informática. CLEI 2005*, pages 1081–1087.

Hasperué, W. and Lanzarini, L. (2006). Classification rules obtained from dynamic self-organizing maps. In *VII Workshop de Agentes y Sistemas Inteligentes. XII Congreso Argentino de Ciencias de la Computación*, pages 1220 –1230.

Hasperué, W. and Lanzarini, L. (2007). Classification rules obtained from evidence accumulation. *Information Technology Interfaces*, 13:167–172.

Hasperué, W. and Lanzarini, L. (2008). Obtaining a fuzzy classification rule system from a non-supervised clustering. *Information Technology Interfaces*, pages 341–346.

Holden, N. and Freitas, A. A. (2008). A hybrid pso/aco algorithm for discovering classification rules in data mining. *J. Artif. Evol. App.*, 2008:2:1–2:11.

Holland, J. H. and Reitman, J. S. (1977). Cognitive systems based on adaptive algorithms. *SIGART Bull.*, (63):49–49.

Houtsma, M. and Swami, A. (1993). Set-oriented mining of association rules. Technical Report RJ 9567, IBM Almaden Research Center, San Jose, California.

Inokuchi, A., Washio, T., and Motoda, H. (2000). An apriori-based algorithm for mining frequent substructures from graph data. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '00, pages 13–23, London, UK, UK. Springer-Verlag.

Jimbo, P., Villa Monte, A., Rucci, E., Lanzarini, L., and Bariviera, A. (2016). An exploratory analysis of methods for extracting credit risk rules. In *XVII Workshop Agentes y Sistemas Inteligentes (WASI). XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*, pages 834–841.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. *IEEE International Conference on Neural Networks*, IV:1942–1948.

Kennedy, J. and Eberhart, R. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers, Inc., San Francisco, CA.

Kennedy, J. and Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. *World Multiconference on Systemics, Cybernetics and Informatics (WMSCI)*, pages 4104–4109.

Khanesar, M., Tavakoli, H., Teshnehlab, M., and Shoorehdeli, M. (2007). A novel binary particle swarm optimization. *18th Mediterranean Conference on Control and Automation.*, pages 1–6.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Lanzarini, L. (2011). Análisis de controles de salud de alumnos de escuelas primarias bonaerenses. Informe técnico, III LIDI - UNLP. Programa Nacional de Sanidad Escolar del Ministerio de Educación.

Lanzarini, L., Charnelli, M. E., Baldino, G., and Díaz, J. (2015a). Selección de atributos representativos del avance académico de los alumnos universitarios usando técnicas de visualización. un caso de estudio. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología (TE&ET)*, (15):42–50.

Lanzarini, L., Charnelli, M. E., and Díaz, J. (2015b). Academic performance of university students and its relation with employment. In *Proceedings of the XLI Latin American Computing Conference (CLEI) - XXIII Simposio Iberoamericano de Educación Superior en Computación*.

Lanzarini, L., Leza, V., and De Giusti, A. (2008). Particle swarm optimization with variable population size. In Rutkowski, L., Tadeusiewicz, R., Zadeh, L. A., and Zurada, J. M., editors, *Artificial Intelligence and Soft Computing - ICAISC 2008*, volume 5097 of *Lecture Notes in Computer Science*, pages 438–449. Springer Berlin Heidelberg.

Lanzarini, L., López, J., Maulini, J. A., and De Giusti, A. (2011a). A new binary pso with velocity control. In Tan, Y., Shi, Y., Chai, Y., and Wang, G., editors, *Advances in Swarm Intelligence*, volume 6728 of *Lecture Notes in Computer Science*, pages 111–119. Springer Berlin Heidelberg.

Lanzarini, L., Ronchetti, F., Estrebou, C., Lens, L., and Bariviera, A. F. (2013). Face recognition based on fuzzy probabilistic som. In *IFSA World Congress - NAFIPS Annual Meeting. IEEE Catalog Nro.: CFP13750-USB*, pages 310–314.

Lanzarini, L., Sanz, C., Naiouf, M., and Romero, F. (2000). Mixed alternative in the assignment by classes vs. conventional methods for calculation of individuals lifetime in gavaps. In *Information Technology Interfaces, 2000. ITI 2000. Proceedings of the 22nd International Conference on*, pages 383–389.

Lanzarini, L., Villa Monte, A., Aquino, G., and De Giusti, A. (2015c). Obtaining classification rules using lvqpso. In Tan, Y., Shi, Y., Buarque, F., Gelbukh, A., Das, S., and Engelbrecht, A., editors, *Advances in Swarm and Computational Intelligence*, volume 9140 of *Lecture Notes in Computer Science*, pages 183–193. Springer International Publishing.

Lanzarini, L., Villa Monte, A., Bariviera, A., and Jimbo, P. (2017). Simplifying credit scoring rules using lvq+pso. *Kybernetes: The International Journal of Systems & Cybernetics*, 46(1).

Lanzarini, L., Villa Monte, A., and César, E. (2011b). E-mail processing with fuzzy soms and association rules. *Journal of Computer Science and Technology*, 11(1):41–46.

Lanzarini, L., Villa Monte, A., and Ronchetti, F. (2015d). Som+pso: A novel method to obtain classification rules. *Journal of Computer Science and Technology*, 15(1):15–22.

Liu, B., Abbas, H. A., and McKay, B. (2002). Density based heuristic for rule discovery with ant-miner. In *The 6th Australia-Japan Joint Workshop on Intelligent and Evolutionary System*, pages 180–184.

Liu, B., Abbas, H. A., and McKay, B. (2003). Classification rule discovery with ant colony optimization. In *IEEE/WIC International Conference on Intelligent Agent Technology, 2003. IAT 2003.*, pages 83–88.

López, J., Vela, J., and Mondejar, J. (2013). *Diseño y explotación de almacenes de datos*. Editorial Club Universitario.

López, J. H., Lanzarini, L., and De Giusti, A. (2009). Particle swarm optimization with oscillation control. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pages 1751–1752, New York, NY, USA. ACM.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

Maulini, J. and Lanzarini, L. (2012). Face recognition using sift descriptors and binary pso with velocity control. In De Giusti, A. and Díaz, J., editors, *Computer Science & Technology Series. XVII Argentine Congress of Computer Science Selected Papers*, pages 43–53. EDULP.

Medland, M., Otero, F. E., and Freitas, A. A. (2012). Improving the cant-minerpb classification algorithm. In Dorigo, M., Birattari, M., Blum, C., Christensen, A., Engelbrecht, A., Grob, R., and Stutzle, T., editors, *Swarm Intelligence*, volume 7461 of *Lecture Notes in Computer Science*, pages 73–84. Springer Berlin Heidelberg.

Molina Felix, L. C. (2001). Data mining: torturando a los datos hasta que confiesen. `http://www.uoc.edu/web/esp/art/uoc/molina1102/molina1102.html`. [Ultimo acceso 28/01/2017].

MYRA (2011). Myra. a collection of aco algorithms for the data mining classification task. `https://sourceforge.net/projects/myra/files/myra/3.5.0/`. [Ultimo acceso 25/01/2017].

Otero, F. E. B., Freitas, A. A., and Johnson, C. G. (2008). *cAnt-Miner: An Ant Colony Classification Algorithm to Cope with Continuous Attributes*, pages 48–59. Springer Berlin Heidelberg, Berlin, Heidelberg.

Pandya, R. and Pandya, J. (2015). C5.0 algorithm to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Application*, 117(16):18–21.

Parpinelli, R. S., Lopes, H. S., and Freitas, A. A. (2002). Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation* , 6(4):321–332.

Pateritsas, C., Modes, S., and Stafylopatis, A. (2007). Extracting rules from trained self-organizing maps. In Guimaraes, N. and Pedro, I., editors, *Proceedings of the IADIS International Conference on Applied Computing*, pages 183–190. ACM.

Pedersen, M. (2010). Good parameters for particle swarm optimization. Technical Report HL1001, Hvass Laboratories.

Piatetsky-Shapiro, G. (1991). Discovery, analysis and presentation of strong rules. In Piatetsky-Shapiro, G. and Frawley, W. J., editors, *Knowledge Discovery in Databases*, pages 229–248. AAAI Press.

Piatetsky-Shapiro, G. and Frawley, W. J., editors (1991). *Knowledge Discovery in Databases*. AAAI/MIT Press.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Quinlan, R. (2015). Data mining tools see5 and c5.0. `http://www.rulequest.com/see5-info.html`. [Ultimo acceso 26/01/2017].

RR, R. R. (2012). Is see5/c5.0 better than c4.5? `http://www.rulequest.com/see5-comparison.html`. [Ultimo acceso 26/01/2017].

Saravana Kumar, K. and Manicka Chezian, R. (2012). A survey on association rule mining using apriori algorithm. *International Journal of Computer Applications*, 45(5):47–50.

Shi, Y. and Eberhart, R. C. (1998). Parameter selection in particle swarm optimization. *7th International Conference on Evolutionary Programming.*, pages 591–600.

Singh, S., Garg, R., and Mishra, P. (2015). Performance analysis of apriori algorithm with different data structures on hadoop cluster. *International Journal of Computer Applications*, 128(9):45–51. Published by Foundation of Computer Science (FCS), NY, USA.

Skiena, S. (2010). *The Algorithm Design Manual (second edition)*. Computer Science: Algorithm Design. Springer-Verlag London Limited 2008.

Smith, S. F. (1980). *A learning system based on genetic adaptive algorithms*. PhD thesis, Pittsburgh, PA, USA. PhD thesis.

Vasoya, A. (2014). Novel approach to improve apriori algorithm using transaction reduction and clustering algorithm. *IJAIS Proceedings on International Conference and workshop on Advanced Computing 2014*, ICWAC 2014(1):37–44. Published by Foundation of Computer Science, New York, USA.

Vega Pons, S. and Ruiz Schulcloper, J. (2011). A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372.

Venturini, G. (1993). Sia: A supervised inductive algorithm with genetic search for learning attributes based concepts. In *Proceedings of the European Conference on Machine Learning*, ECML '93, pages 280–296, London, UK, UK. Springer-Verlag.

Wang, L. and Fu, X. (2006). *Data Mining with Computational Intelligence*. Advanced Information and Knowledge Processing. Springer Berlin Heidelberg.

Wang, Z., Sun, X., and Zhang, D. (2006). *Classification Rule Mining Based on Particle Swarm Optimization*, pages 436–441. Springer Berlin Heidelberg, Berlin, Heidelberg.

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.

Wu, H., Lu, Z., Pan, L., Xu, R., and Jiang, W. (2009). An improved apriori-based algorithm for association rules mining. In *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery - Volume 2*, FSKD'09, pages 51–55, Piscataway, NJ, USA. IEEE Press.