



OBTENCIÓN DE REGLAS DE CLASIFICACIÓN DIFUSAS UTILIZANDO TÉCNICAS DE OPTIMIZACIÓN - CASO DE ESTUDIO RIESGO CREDITICIO

Patricia Rosalía Jimbo Santana

ADVERTIMENT. L'accés als continguts d'aquesta tesi doctoral i la seva utilització ha de respectar els drets de la persona autora. Pot ser utilitzada per a consulta o estudi personal, així com en activitats o materials d'investigació i docència en els termes establerts a l'art. 32 del Text Refós de la Llei de Propietat Intel·lectual (RDL 1/1996). Per altres utilitzacions es requereix l'autorització prèvia i expressa de la persona autora. En qualsevol cas, en la utilització dels seus continguts caldrà indicar de forma clara el nom i cognoms de la persona autora i el títol de la tesi doctoral. No s'autoritza la seva reproducció o altres formes d'explotació efectuades amb finalitats de lucre ni la seva comunicació pública des d'un lloc aliè al servei TDX. Tampoc s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant als continguts de la tesi com als seus resums i índexs.

ADVERTENCIA. El acceso a los contenidos de esta tesis doctoral y su utilización debe respetar los derechos de la persona autora. Puede ser utilizada para consulta o estudio personal, así como en actividades o materiales de investigación y docencia en los términos establecidos en el art. 32 del Texto Refundido de la Ley de Propiedad Intelectual (RDL 1/1996). Para otros usos se requiere la autorización previa y expresa de la persona autora. En cualquier caso, en la utilización de sus contenidos se deberá indicar de forma clara el nombre y apellidos de la persona autora y el título de la tesis doctoral. No se autoriza su reproducción u otras formas de explotación efectuadas con fines lucrativos ni su comunicación pública desde un sitio ajeno al servicio TDR. Tampoco se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al contenido de la tesis como a sus resúmenes e índices.

WARNING. Access to the contents of this doctoral thesis and its use must respect the rights of the author. It can be used for reference or private study, as well as research and learning activities or materials in the terms established by the 32nd article of the Spanish Consolidated Copyright Act (RDL 1/1996). Express and previous authorization of the author is required for any other uses. In any case, when using its content, full name of the author and title of the thesis must be clearly indicated. Reproduction or other forms of for profit use or public communication from outside TDX service is not allowed. Presentation of its content in a window or frame external to TDX (framing) is not authorized either. These rights affect both the content of the thesis and its abstracts and indexes.



UNIVERSIDAD
NACIONAL
DE LA PLATA



UNIVERSITAT
ROVIRA I VIRGILI

Obtención de reglas de clasificación difusas utilizando técnicas de optimización – Caso de estudio Riesgo Crediticio

Tesis doctoral realizada por

Patricia Rosalía Jimbo Santana

dirigida en cotutela por

Dra. Laura C. Lanzarini
Facultad de Informática
Universidad Nacional de La Plata
La Plata (Argentina)

Dr. Aurelio Fernández Bariviera
Departamento de Gestión de Empresas
Universitat Rovira i Virgili
Reus (España)

2020



UNIVERSITAT
ROVIRA I VIRGILI



UNIVERSIDAD
NACIONAL
DE LA PLATA

FAIG CONSTAR que aquest treball, titulat "Obtención de Reglas de Clasificación Difusas utilizando Técnicas de Optimización - Caso de estudio Riesgo Crediticio", que presenta Patricia Jimbo Santana per a l'obtenció del títol de Doctor, ha estat realitzat sota la nostra direcció al Departament de Gestió d'Empreses de la Universitat Rovira i Virgili i a la Facultat de Informàtica de la Universidad Nacional de La Plata (Argentina).

HAGO CONSTAR que el presente trabajo, titulado "Obtención de Reglas de Clasificación Difusas utilizando Técnicas de Optimización - Caso de estudio Riesgo Crediticio", que presenta Patricia Jimbo Santana para la obtención del título de Doctor, ha sido realizado bajo nuestra dirección en el Departamento de Gestión de Empresas de la Universitat Rovira i Virgili y en la Faculta de Informàtica de la Universidad Nacional de La Plata (Argentina).

La Plata, 11 de abril de 2020
Reus, 11 de abril de 2020

Els directors de la tesi doctoral
Los directores de la tesis doctoral

Dra. Laura C. Lanzarini
Facultad de Informàtica
Universidad Nacional de La Plata

Dr. Aurelio Fernández Bariviera
Departamento de Gestión de Empresas
Universitat Rovira i Virgili

*Nunca abandones tus sueños
con constancia, disciplina y perseverancia
puedes lograr todo lo que te propongas
“La paciencia todo lo alcanza”*

“Confía en ti mismo nunca pierdas la FE y la ESPERANZA”

Agradecimientos

Dedico este trabajo a Dios, que me regalo la vida y me ha dado la oportunidad de cumplir mis metas, me ha enseñado que todo tiene un propósito en la vida aunque no lo entienda. A mis Padres que me han formado con principios y valores, han hecho de mí lo que soy. Ellos me enseñaron a luchar por mis sueños, que continúe a pesar de las adversidades, y que paso a paso, con perseverancia se llega a la meta. Son la bendición más grande que me ha dado Dios. A mis hermanas, a Manu por ser mi segunda mamá por estar conmigo en la oficina, por haberme apoyado en cada una de mis caídas y creer en mí, por sacrificarte y dejar muchas de las cosas que te gustaban por mí, por enviar tus ángeles para que me acompañen, gracias Niña Manu. A ti Nenita por tu apoyo incondicional y por brindarme tu mano cada vez que lo necesitaba, por ti empecé este sueño, nunca me dejaste sola, gracias por hacerme reír cuando todo parecía cuesta arriba, y recordarme que si uno ama lo que hace no es trabajar, que *“no puedo cambiar el mundo, pero si puedo cambiar mi mundo”*.

A alguien muy especial a Laura Lanzarini, que en estos años más que una directora de tesis se convirtió en mi amiga, por recordarme que *“no me dé por vencida ni aún vencida”*, a pesar de los mil y un problemas, que *“nunca pierda la FE”*. Gracias Laura por creer, confiar en mí, bríndame tu ayuda, tus conocimientos, y especialmente tus consejos para lograr este sueño. Me hiciste valorar lo que era realmente importante, siempre estuviste ahí cuando tenía una pregunta, una inquietud sin importar la hora, ni tus ocupaciones. Gracias por tus exigencias, y por brindarme tu amistad, por ellas pude terminar mi trabajo, que Dios te bendiga infinitamente por todo lo que hiciste por mí.

A Aurelio F. Bariviera, a quién pude conocer, me ayudo con sus conocimientos y su experiencia, estuvo conmigo durante todo el desarrollo de la tesis; gracias Aurelio por tu orientación, tus consejos, tu ayuda y guía en las publicaciones de este trabajo, sin ti esta tesis no hubiera sido posible.

A mis amigos Federico Zambrano y Ferney Correa, de cada uno de ustedes recibí un abrazo fraterno, y una palabra de aliento cuando más lo necesite, gracias. A todos aquellos con quienes no pude compartir por dedicar mi tiempo a este sueño, gracias por entenderme.

Por último, gracias tanto a la **Universidad Nacional de La Plata** como a la **Universidad Rovira i Virgili** por apoyar la realización de mi Doctorado en cotutela.

¡¡¡MUCHAS GRACIAS!!!

Patricia Rosalía Jimbo Santana

Índice general

Índice de figuras	v
Índice de tablas	vii
Abreviaturas	ix
1. Reglas de Clasificación	23
1.1. Minería de datos y el proceso de KDD	23
1.2. Etapas del proceso de KDD	24
1.2.1. Vista Minable	25
1.2.2. Minería de Datos	26
1.2.3. Interpretación, Difusión y uso de los resultados	28
1.3. Tareas de Minería de Datos	29
1.4. Técnicas de Minería de Datos	29
1.5. Reglas para extracción del conocimiento	31
1.5.1. Reglas de asociación	31
1.5.2. Reglas de clasificación	33
1.6. Métodos para extracción de Reglas de Clasificación	34
1.6.1. Árboles	34
1.6.2. Redes Neuronales	38
1.6.3. Máquinas de vectores de soporte	39
1.6.4. Técnicas de Optimización	40
Optimización por cumulo de partículas (Particle Swarm Opti- mization - PSO)	44
1.7. Conclusiones	45
2. Lógica Difusa	47
2.1. Introducción	47
2.2. Aplicación de la Lógica Difusa	48
2.3. Conceptos básicos de Lógica Difusa	49
2.3.1. Conjuntos Difusos y Función de Pertenencia	49
2.3.2. Variables Lingüísticas	53
2.3.3. Particiones Difusas	54
2.3.4. Operaciones Difusas	56
2.3.5. Representación de conjuntos difusos	57
2.3.6. Reglas Difusas	57

II

Sistemas basados en Reglas difusas	59
2.4. Representación difusa del conocimiento	63
2.5. Métodos para extracción de reglas difusas	64
2.6. Sistemas que combinan Redes Neuronales, Técnicas de optimización y Lógica Difusa	66
2.7. Conclusiones	67
3. Extracción de reglas utilizando PSO	69
3.1. Introducción	69
3.2. Representación de las Reglas	70
3.3. Estructura de la Partícula	71
3.3.1. Función Fitness	74
3.4. Construcción de Reglas utilizando PSO	75
3.5. Reglas de clasificación aplicadas a Riesgo Crediticio	77
3.5.1. Resultados Obtenidos	79
3.6. Conclusiones	79
4. Método propuesto FRvarPSO	83
4.1. Algoritmo de extracción de reglas propuesto	84
4.1.1. Variables lingüísticas. Etiquetas y conjuntos difusos	85
4.1.2. Estructura de una Partícula	87
Representación del Consecuente	90
4.2. Resultados obtenidos	92
4.2.1. Particionando el rango de cada atributo numérico en intervalos de igual longitud	94
4.2.2. Utilizando Fuzzy-C Means	99
4.2.3. Interpretación de las Reglas Difusas	102
4.3. Conclusiones	104
5. Caso de estudio: Riesgo Crediticio	105
5.1. Riesgo Crediticio	105
5.1.1. Técnicas Informáticas utilizadas en Riesgo Crediticio	110
5.2. FRvarPSO aplicado a Riesgo Crediticio	113
5.2.1. Datos de crédito de Instituciones Financieras	113
5.2.2. Fuzzificación de las variables	124
5.3. Resultados obtenidos	125
5.3.1. Utilizando el criterio del experto crediticio	127
5.3.2. Resultados comparativos con conjuntos equitativos, FCM y criterio del experto crediticio	128
5.4. Conclusiones	128

6. Conclusiones Finales y Líneas de Trabajo Futuras	137
6.1. Conclusiones	137
6.2. Líneas de trabajo futuras	139
A. Redes Neuronales para inicialización de PSO	141
A.1. SOM	144
A.2. LVQ	146
B. Versiones de PSO	149
B.1. PSO Básico	149
B.2. PSO Binario	150
B.3. PSO de Población Variable	152
Bibliografía	157

Índice de figuras

1.	Comparación de la Precisión de los métodos	5
2.	Comparación de la Cardinalidad de los métodos	6
3.	Comparación de la Precisión y Cardinalidad de los métodos	7
4.	Banco encargado de microcrédito	7
5.	Banco encargado de crédito de consumo y crédito productivo empresarial	8
6.	Cooperativa de Ahorro y Crédito del Ecuador que encuentra en el segmento 2 dentro de la Superintendencia de Economía Popular y Solidaria, activos mayor a 20'000.000,00 hasta 80'000.000,00	8
7.	Compararison of the accuracy of the different methods.	17
8.	Comparison of the cardinality of the methods.	18
9.	Relationship of accuracy and cardinality of each method under analysis.	18
10.	Bank dedicated to microcredits.	19
11.	Bank dedicated to consumption and production credits.	19
12.	Savings and credit cooperative from segment 2 of the Superintendency of Popular and Solidarity Economy, with assets between 20 and 80 USD millions.	20
1.1.	Etapas del proceso de Extracción de Conocimiento en Bases de Datos (KDD)	24
1.2.	Integración de datos	26
1.3.	Submuestreo	27
1.4.	Sobremuestreo	27
1.5.	Módulos para extracción de reglas utilizando SVM	41
1.6.	Clasificación de las Técnicas de Optimización	41
1.7.	Clasificación de las Metaheurísticas	42
1.8.	Ejemplos de Inteligencia de Cúmulo en la naturaleza	44
2.1.	Representación gráfica del grado de pertenencia de los elementos a un conjunto clásico (izquierda) y un conjunto difuso (derecha)	50
2.2.	Función de Pertenencia Trapezoidal	51
2.3.	Función de Pertenencia Singleton	51
2.4.	Función de Pertenencia Triangular	51
2.5.	Función de Pertenencia Tipo S	52
2.6.	Función de Pertenencia Tipo Pseudo-exponencial)	52
2.7.	Función de Pertenencia Tipo π (con forma de campana)	52

VI

2.8. Ejemplo de conjuntos difusos	54
2.9. Funciones de pertenencia a los conjuntos difusos Bajo, Medio y Alto de la variable Ingreso	55
2.10. Estructura genérica de un Sistema Basado en Reglas Difusas	59
2.11. Sistemas Difusos puros	61
2.12. Sistema Difuso: Takagi-Sugeno-Kang (TSK)	61
2.13. Sistemas difusos con Fuzzificador y Defuzzificador	61
2.14. Representación de un algoritmo para extracción de reglas difusas utilizando SVM	65
4.1. Comparación de la Precisión de los métodos	99
4.2. Comparación de la cardinalidad de los métodos	100
4.3. Comparación de la precisión y Cardinalidad de los métodos	100
5.1. Proceso de credit scoring	108
5.2. Proceso de credit scoring	114
5.3. Función de pertenencia triangular de la variable "Valor de la operación"	125
5.4. Función de pertenencia triangular de la variable "Disponible"	126
5.5. Función de pertenencia triangular de la variable "Ingresos totales"	126
5.6. Análisis comparativo de la precisión promedio de cada método en la Cooperativa de Ahorro y Crédito	131
5.7. Análisis comparativo de la precisión promedio de cada método en el Banco del Ecuador encargado de Microcrédito, especializado en colocación masiva de crédito	133
5.8. Análisis comparativo de la precisión promedio de cada método en Banco del Ecuador encargado de crédito de consumo y crédito productivo o empresarial	133
A.1. Arquitectura del Self Organizing Maps (SOM)	145

Índice de tablas

2.1. Grados de pertenencia de cada elemento de los conjuntos difusos que determinan los valores lingüísticos de la variable difusa "Salario"	55
3.1. Descripción de Bases de datos utilizadas	79
3.2. Resultados con base de datos Australiana del repositorio UCI	80
3.3. Resultados con base de datos Alemana del repositorio UCI	80
3.4. Resultados con base de datos de Cooperativa de Ahorro y Crédito del Ecuador	80
3.5. Resultados con base de datos de Institución Financiera Banco del Ecuador que se encarga de dar crédito de consumo productivo y no productivo	81
4.1. Antecedente de una regla difusa	90
4.2. Descripción de Bases de datos del repositorio UCI y de tres instituciones financieras del Ecuador	94
4.3. Resultados de la Precisión obtenidos al aplicar los métodos a un conjunto de bases del repositorio UCI y de tres instituciones financieras del Ecuador	95
4.4. Resultados de la Precisión obtenidos al aplicar los métodos difusos a un conjunto de bases del repositorio UCI y de tres instituciones financieras del Ecuador	96
4.5. Resultados del número de reglas obtenidos al aplicar los métodos a un conjunto de bases del repositorio UCI y de tres instituciones financieras del Ecuador	97
4.6. Resultados del número de reglas obtenidos al aplicar los métodos Difusos a un conjunto de bases del repositorio UCI y de tres instituciones financieras del Ecuador	98
4.7. Resultados de la Precisión obtenidos al aplicar los métodos a un conjunto de bases del repositorio UCI y tres instituciones financieras del Ecuador utilizando Agrupamiento Difuso (FCM)	101
4.8. Resultados del Número de Reglas obtenidos al aplicar los métodos a un conjunto de bases del repositorio UCI y tres instituciones financieras del Ecuador utilizando FCM	102
5.1. R1. Sujetos de crédito	115
5.2. R2. Operaciones concedidas	117

VIII

5.3. R3. Saldo de operaciones	121
5.4. Variables microeconómicas de análisis	123
5.5. Morosidad	124
5.6. Variables macroeconómicas de análisis	124
5.7. Cooperativa de Ahorro y Crédito del Ecuador	128
5.8. Cooperativa de Ahorro y Crédito del Ecuador	129
5.9. Resultado de Reglas Banco del Ecuador encargado de Microcrédito	129
5.10. Resultado de Reglas Difusas Banco del Ecuador encargado de Microcrédito	130
5.11. Resultado de Reglas Banco del Ecuador encargado de crédito de consumo y crédito productivo o empresarial	130
5.12. Resultado de reglas Difusas Banco del Ecuador encargado de crédito de consumo y crédito productivo o empresarial	131
5.13. Comparación de la precisión entre los modelos híbridos y aquellos que utilizan Lógica Difusa	132
5.14. Resultado de reglas Difusas Cooperativa de Ahorro y Crédito del Ecuador	132
5.15. Resultado de reglas Difusas Banco del Ecuador encargado de Microcrédito	134
5.16. Resultado de reglas Difusas Banco encargado de crédito de consumo y crédito productivo o empresarial	134

Abreviaturas

ACO Colonia de Hormigas

AG Algoritmos Genéticos

ANS Sistemas Neuronales Artificiales

CV Criterio de votación

DM Minería de Datos

FCM Agrupamiento Difuso

FRBS Sistema Basado en Reglas Difusas

FRL Aprendizaje de Reglas Difusas

FRvarPSO Fuzzy Rules Variable Particle Swarm Optimization

IRL Aprendizaje Iterativo de Reglas

KB Base de Conocimientos

KDD Extracción de Conocimiento en Bases de Datos

LD Lógica Difusa

LVQ Learning Vector Quantization

OLAP Procesamiento Analítico de Datos en Línea

OLTP Procesamiento Transaccional de Datos en Línea

PART Árboles de decisión Parciales

PG Programación Genética

PSO Optimización mediante Cúmulo de Partículas

SMOTE Synthetic Minority Over sampling

SOM Self Organizing Maps

X

SVM Máquinas de Vectores de Soporte

UCI UCI Machine Learning Repository

varPSO Variable Particle Swarm Optimization

Resumen

EN los últimos años gracias al avance de la tecnología, las organizaciones han almacenado gran cantidad de información. Esto las ha llevado a la necesidad de incorporar técnicas que permitan procesar y obtener información útil de los datos. El proceso de Extracción del Conocimiento, conocido como proceso de **KDD** (*Knowledge Discovery in Databases*), está formado por varias fases que van desde la recolección y transformación de los datos hasta la identificación e interpretación de patrones o relaciones subyacentes sumamente útiles en la toma de decisiones.

Dentro del proceso de **KDD**, la Minería de Datos (*Data Mining*) es considerada la fase más importante, ya que agrupa a las técnicas capaces de modelizar la información disponible. A partir del uso o comprensión del modelo generado es posible extraer conocimiento. Este conocimiento que se genera resulta de gran interés para las organizaciones, debido a que constituye una herramienta sumamente importante para la toma de decisiones tácticas y estratégicas, lo cual se convierte en una ventaja competitiva.

Según el tipo de tarea que se quiera resolver existe en la literatura una gran cantidad de técnicas disponibles entre las que podemos mencionar: las algebraicas y estadísticas, estocásticas y difusas, técnicas bayesianas, árboles de decisión, redes neuronales artificiales, máquinas de vectores de soporte y técnicas de optimización entre otras. Cada una de ellas tiene características diferentes.

Para poder seleccionar el tipo de técnica es necesario conocer el problema que se va a resolver teniendo en cuenta que no todas ellas pueden operar con datos faltantes, outliers o datos en cualquier formato. Por otra parte, los problemas que se abordan pueden ser de diferente tipología como por ejemplo: clasificación, categorización, regresión o agrupamiento.

Una característica deseable de los modelos construidos con las técnicas de la Minería de Datos es que el conocimiento que se extrae se exprese en términos comprensibles. En este sentido, las reglas de asociación son consideradas, por quienes deben tomar decisiones, como una de las formas más comprensibles que puede ser utilizada para representar el conocimiento, ya que tienen la capacidad de explicarse por sí mismas. Una regla de asociación es una expresión de la forma "SI *condic1* ENTONCES *condic2*" donde ambas condiciones son conjunciones de proposiciones de

la forma (atributo=valor) cuyos atributos/variables pueden ser cuantitativas o cualitativas. Por su estructura condicional, puede afirmarse que el objetivo principal de una regla es identificar relaciones entre los atributos. Cuando el conjunto de reglas de asociación presenta en el consecuente el mismo atributo, se dice que se trata de un conjunto de "reglas de clasificación". Si a esto se le suma que las reglas de clasificación empleen lógica difusa a través de conjuntos difusos para describir los valores de sus atributos, se obtienen "Reglas de Clasificación Difusas", facilitando aun más su comprensión, permitiendo además el manejo de la incertidumbre, aproximándonos cada vez más al razonamiento humano. Tal como se menciona en (Terano, Asai y Sugeno, 1992), "cuanto más humano debe ser un sistema, más lógica difusa contendrá".

La utilización de la lógica difusa para la generación de las reglas ha permitido que los investigadores obtengan formas de mejorar el desempeño de las metaheurísticas, básicamente para acelerar la convergencia y obtener una mejor calidad en las soluciones planteadas.

Las reglas de clasificación difusas son reglas que tienen sentencias lingüísticas que describen la manera en que los sistemas de inferencia difusa toman las decisiones sobre un conjunto de variables de entrada. Estas reglas de clasificación poseen un antecedente formado por variables nominales y/o condiciones difusas, y un consecuente conocido con anticipación denominado clase

El aporte central de esta tesis es la definición de un nuevo método capaz de generar un conjunto de reglas de clasificación difusas de fácil interpretación, baja cardinalidad y una buena precisión. Estas características ayudan a identificar y comprender las relaciones presentes en los datos facilitando de esta forma la toma de decisiones.

El nuevo método propuesto se denomina **FRvarPSO (Fuzzy Rules variable Particle Swarm Oprmization)** y combina una red neuronal competitiva con una técnica de optimización basada en cúmulo de partículas de población variable para la obtención de reglas de clasificación difusas, capaces de operar sobre atributos nominales y numéricos. Los antecedentes de las reglas están formados por atributos nominales y/o condiciones difusas. La conformación de estas últimas requiere conocer el grado de pertenencia a los conjuntos difusos que definen a cada variable lingüística. Esta tesis propone tres alternativas distintas para resolver este punto.

Con respecto a la forma de obtención de las reglas, el método propuesto utiliza un proceso iterativo por medio del cual se van cubriendo los ejemplos de una clase a la vez hasta lograr la cobertura deseada. Por lo tanto, el consecuente de la regla queda determinado por la clase seleccionada y es el antecedente el que se extrae a través de la técnica de optimización.

Cada partícula de la población utiliza una representación de longitud fija, donde sólo se codifica el antecedente de la regla y dado el enfoque adoptado, se efectuará un proceso iterativo asociando todos los individuos de la población con una clase predeterminada. El hecho de que todos los individuos pertenezcan a la misma clase

hace innecesaria la codificación del consecuente dentro de la partícula.

El proceso de extracción de cada regla inicia con la generación del cúmulo. Las posiciones iniciales de las partículas tienen una gran incidencia en la capacidad exploratoria de cada una de ellas. Si se encuentran demasiado próximas, su conocimiento social puede provocar la convergencia prematura evitando que se alcancen buenas soluciones. Por otro lado, si inician muy lejos del óptimo se requerirán un mayor número de iteraciones o pasos de búsqueda para alcanzarlo. Por todo lo anterior, se decidió utilizar una red neuronal competitiva para identificar las zonas más prometedoras. Esta información fue utilizada para inicializar el cúmulo.

En lo que se refiere al tamaño de la población, se optó por un cúmulo de tamaño variable con capacidad para incorporar individuos en las zonas menos pobladas. Como se dijo anteriormente, en PSO las partículas tienden a reunirse alrededor de las buenas posiciones encontradas por sus vecinos esto hace que el cúmulo pierda diversidad y varias zonas del espacio de búsqueda queden despobladas. La técnica para modificar el tamaño de la población trabaja sobre dos conceptos: por un lado detecta las zonas densas y elimina individuos muy similares y por otro identifica las zonas muy despobladas y agrega partículas adecuadamente. Esto permite iniciar con pocos individuos e ir modificando el tamaño de la población durante el proceso iterativo de búsqueda.

Volviendo a la representación de la partícula, a la información referida al movimiento del individuo se le agregan dos cuestiones referidas a la construcción del antecedente de la regla. En primer lugar, debe indicarse cuáles serán los atributos que formarán parte del antecedente y esto debe hacerse utilizando información binaria. Por otro lado, es preciso representar el grado de pertenencia de los atributos originalmente numéricos en cada conjunto difuso para conocer cómo operar con las correspondientes variables lingüísticas, a través de una representación difusa.

Uno de los aportes de esta tesis radica en la definición de la función de aptitud o fitness de cada partícula basada en un "*Criterio de Votación*" que pondera de manera difusa la participación de las condiciones difusas en la conformación del antecedente. Su valor se obtiene a partir de los grados de pertenencia de los ejemplos que cumplen con la regla y se utiliza para reforzar el movimiento de la partícula en la dirección donde se encuentra el valor más alto. Con la utilización de PSO las partículas compiten entre ellas para encontrar a la mejor regla de la clase seleccionada.

La eficiencia y eficacia de FRvarPSO se encuentran fuertemente condicionadas por la manera en que se determinen las funciones de pertenencia de los conjuntos difusos. En el marco de las investigaciones de esta tesis se han utilizado diferentes opciones. Uno de estas opciones fue particionar el rango de cada atributo numérico en intervalos de igual longitud, y centrando en cada uno de ellos una función

triangular con un solapamiento adecuado. Otra de las formas para obtener los conjuntos difusos ha sido utilizando el método Fuzzy C-Means. Adicionalmente, se utilizó también como técnica el conocimiento de un experto para la definición de los conjuntos difusos, y su correspondiente valor de pertenencia.

El desempeño del método propuesto FRvarPSO fue comparado con versiones previas del mismo como son *SOM + PSO*, *SOM + varPSO* (PSO con población variable), *LVQ + PSO*, así como otros métodos de extracción de reglas de clasificación como PART y C4.5. Adicionalmente se comparó FRvarPSO con las versiones *SOM + PSO Difuso*, *SOM + varPSO Difuso*, *LVQ + PSO Difuso*, que son las versiones anteriores a las que se les aplicó lógica difusa al momento de construir el antecedente junto con el criterio de votación utilizado en la función fitness.

La medición se realizó sobre doce bases de datos del repositorio UCI (Machine Learning Repository) y tres casos reales en el área de crédito del Sistema Financiero del Ecuador asociadas al riesgo crediticio considerando un conjunto de variables micro y macroeconómicas. Dentro de las bases de datos del Sistema Financiero del Ecuador se contó con la información de una cooperativa de ahorro y crédito, y las otras dos bases de datos correspondieron a bancos que otorgan crédito de consumo productivo, no productivo y microcrédito, que se encuentran en el mercado por más de 25 años.

FRvarPSO se empleó en bases de datos reales en el proceso de análisis para la concesión de crédito. Es importante remarcar la ardua tarea realizada en las tareas de limpieza y transformación de la información, antes de la aplicación de los métodos. Se analizó la función de distribución de cada atributo a fin de detectar valores anómalos, faltantes y/o fuera de rango. Se transformaron variables con la finalidad de tener indicadores aplicados al cliente como por ejemplo: la capacidad de pago, la capacidad de endeudamiento y la calificación crediticia. Esta última tuvo un análisis particular, debido a que se disponía de información de las características generales del cliente, así como del comportamiento de los créditos otorgados en un horizonte de tiempo, con lo cual se obtuvo el nivel de morosidad.

Otro de los aportes de esta tesis fue haber realizado una consideración especial en la morosidad del cliente teniendo en cuenta los días de vencimiento de la cartera otorgada; esto fue posible debido a que se tenía información del cliente en un horizonte de tiempo, una vez que el crédito se había concedido. De esta forma, se consideraron tres casos: la cartera vencida, la que no devenga intereses y la cartera castigada.

Se verificó que con este análisis las reglas difusas obtenidas a través de FRvarPSO permiten que el oficial de crédito de respuesta al cliente en menor tiempo, y principalmente disminuya el riesgo que representa el otorgamiento de crédito para las instituciones financieras. Lo anterior fue posible, debido a que al aplicar una regla difusa se toma el menor grado de pertenencia promedio de las condiciones difusas

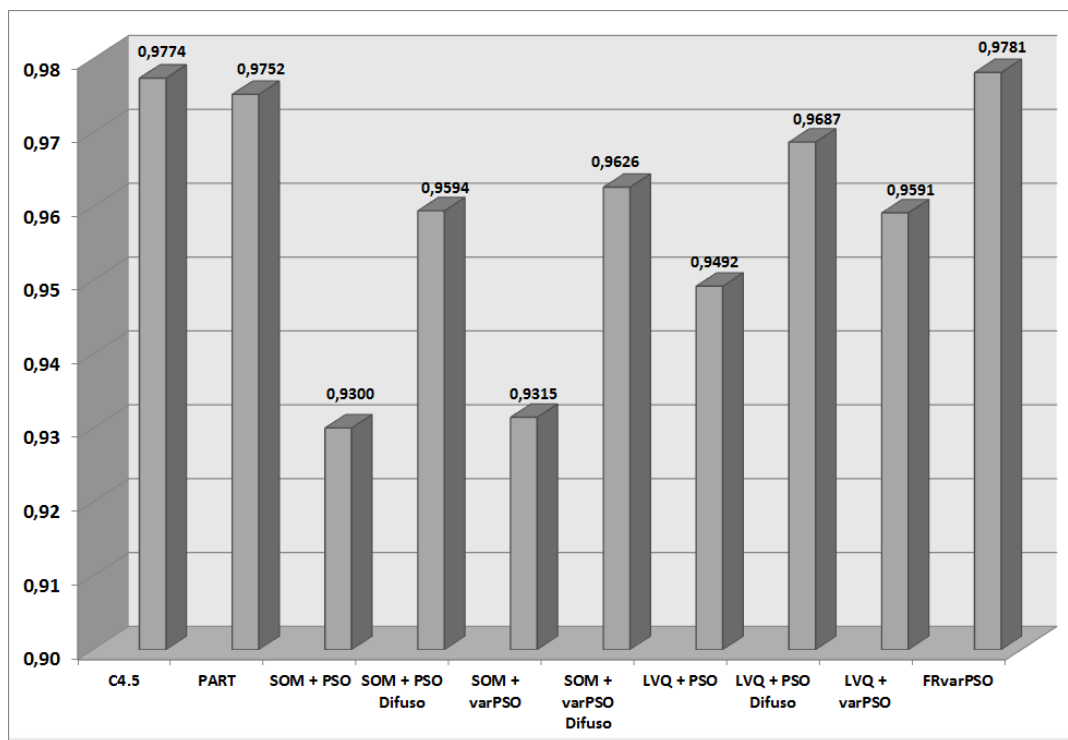


FIGURA 1: Comparación de la Precisión de los métodos

que forman el antecedente de la regla, con lo que se tiene una métrica proporcional al riesgo de su aplicación. Con esta observación el oficial de crédito puede tomar la decisión de conceder el crédito incrementando la tasa de interés, las garantías y/o colaterales (activo), con la finalidad de disminuir el riesgo asociado.

Los resultados obtenidos fueron comparados mediante tests de diferencia de medias, verificándose que los modelos difusos presentan en la mayor parte de los casos una precisión superior a la del método PART, pero ligeramente mayor a la alcanzada por el método C4.5. Sin embargo, si se observa la cantidad de reglas generadas para alcanzar dicha precisión, los métodos difusos utilizan una cantidad promedio de reglas mucho menor, que las reglas de C4.5 y PART. Esto último ratifica el *énfasis puesto en la sencillez del modelo* y en su facilidad de interpretación, gracias a la utilización de la lógica difusa.

En la figura 1 se ilustra un análisis comparativo de los resultados obtenidos al aplicar los métodos a las 12 bases de datos del repositorio UCI así como a las 3 bases de datos reales del Sistema Financiero Ecuatoriano. Aquí se puede observar la relación que existe entre la precisión de cada uno de los métodos con la mejor solución encontrada. Dicho valor corresponde al cociente entre la precisión promedio obtenida por el método y la mejor precisión obtenida, es decir, la mayor precisión de la base de datos analizada. Aquí se puede observar como el método propuesto tiene una mejor precisión, que los otros métodos analizados.

En la figura 2 se puede observar la relación que existe entre la cardinalidad (número de reglas) de cada uno de los métodos con la cardinalidad de la mejor solución

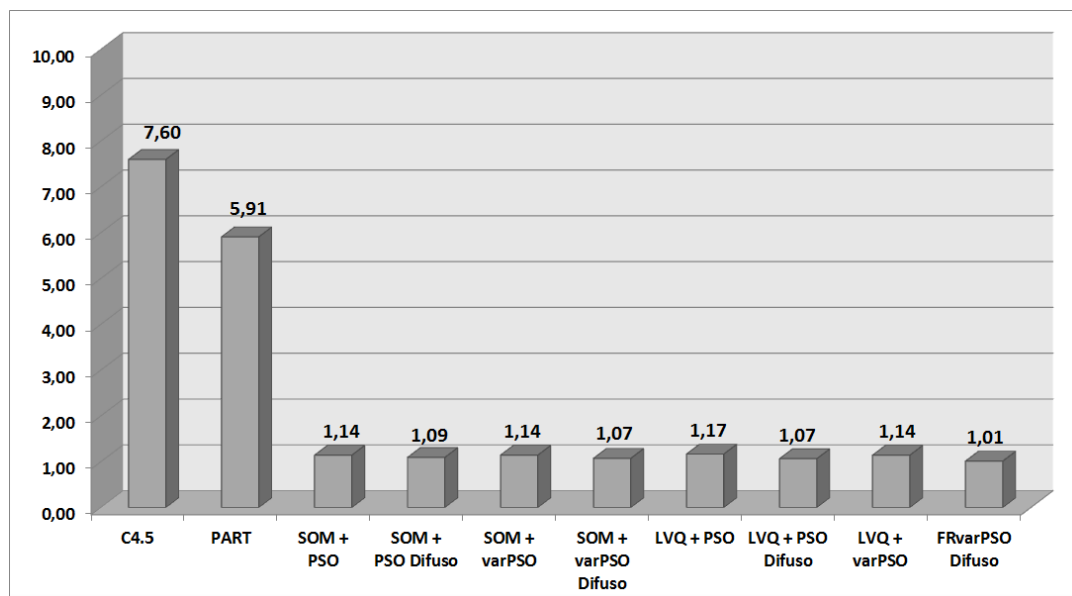


FIGURA 2: Comparación de la Cardinalidad de los métodos

encontrada. Dicha relación corresponde al cociente entre el promedio del valor del número de reglas difusas obtenidas por el método y la solución que tiene el menor número de reglas difusas de la base de datos analizada. Los resultados obtenidos permiten afirmar que el método propuesto es el que obtiene el menor número de reglas.

En la figura 3 se puede observar la relación entre la precisión y la cardinalidad de los métodos analizados.

En las figuras 4, 5, 6 se puede observar claramente la relación que existe entre la precisión y el número de reglas de las tres bases de datos reales del Sistema Financiero Ecuatoriano utilizando FRvarPSO y generando los conjuntos difusos por medio de particiones equitativas, Fuzzy C-Means y con conocimiento del experto.

Finalmente como resultado, se puede indicar que al utilizar FRvarPSO y no contar con el criterio del experto para la determinación de las funciones de pertenencia y el conjunto difuso definido por ella, el utilizar FCM, da como resultado un conjunto de reglas difusas con mayor precisión, menor cardinalidad y fácil interpretación, que cuando se utilizan conjuntos equitativos. A base de los resultados obtenidos se puede decir que el objetivo de la tesis se ha cumplido.

Palabras Claves: Reglas de Clasificación Difusas (Fuzzy Classification Rules), Optimización mediante Cúmulo de partículas tamaño variable (Variable Particle Swarm Optimization), Minería de datos.

Publicaciones

En el desarrollo de esta tesis se realizaron las siguientes publicaciones:

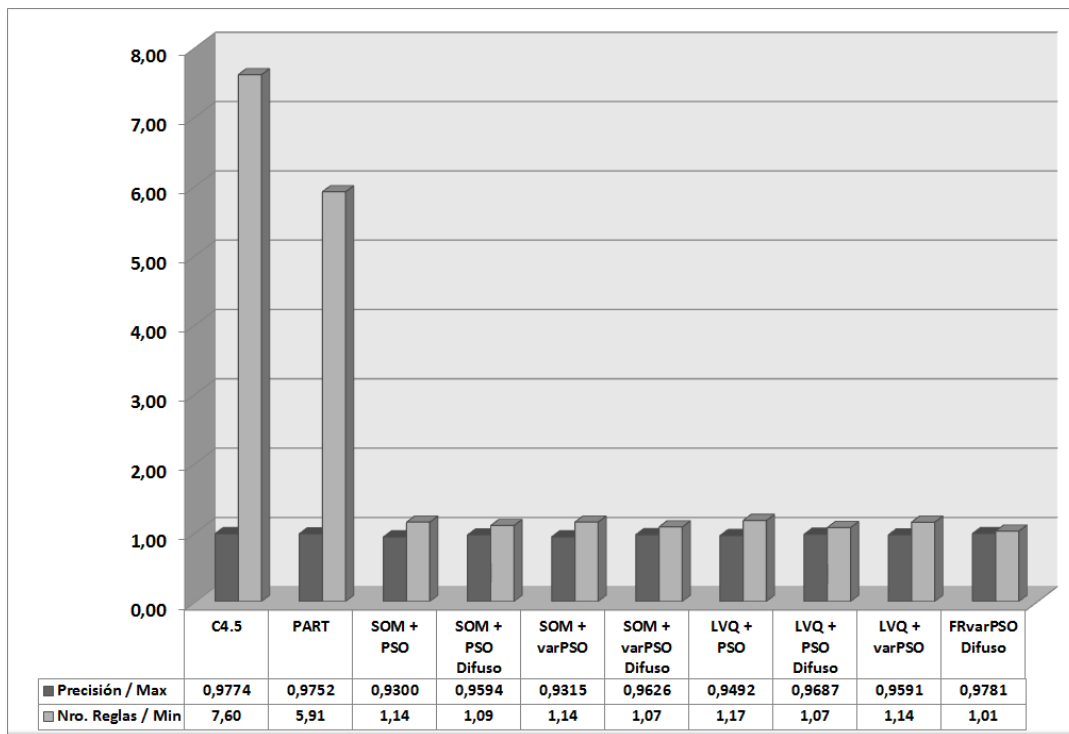


FIGURA 3: Comparación de la Precisión y Cardinalidad de los métodos

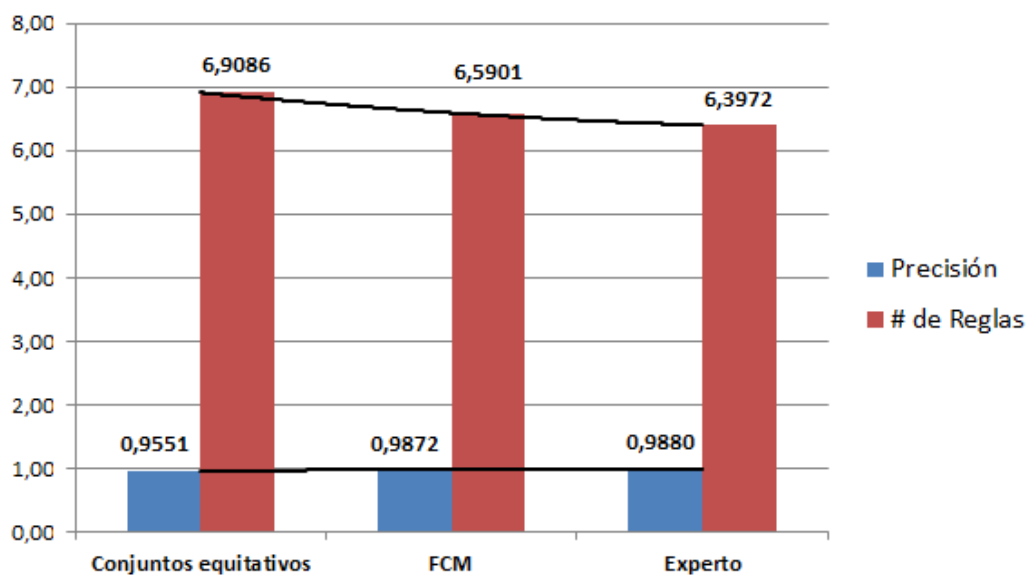


FIGURA 4: Banco encargado de microcrédito

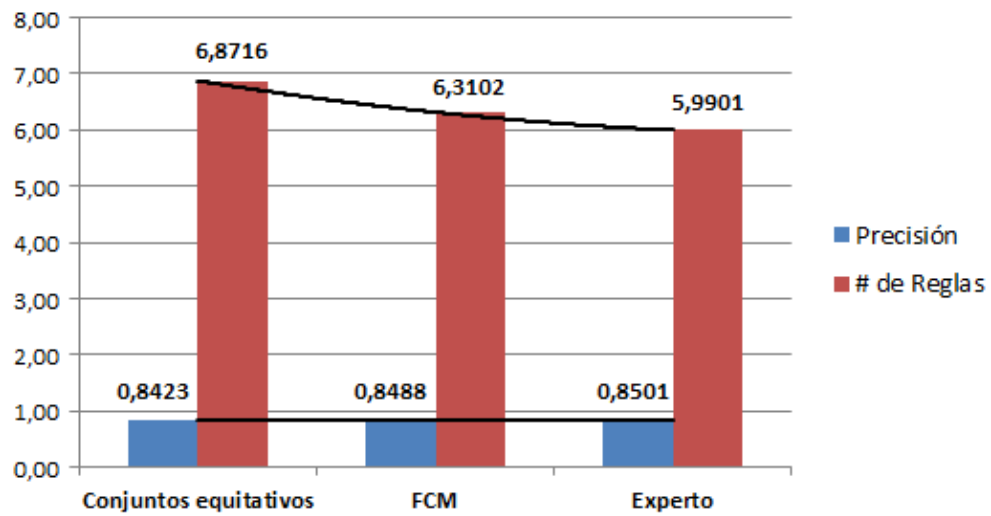


FIGURA 5: Banco encargado de crédito de consumo y crédito productivo empresarial

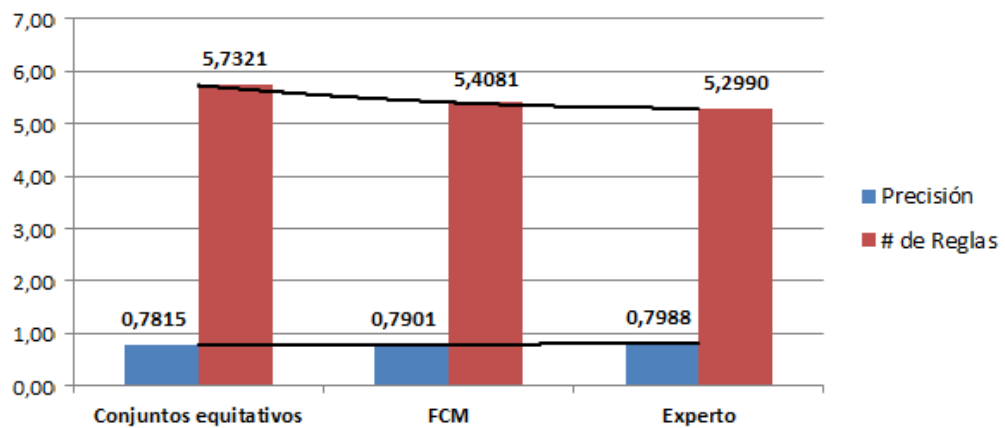


FIGURA 6: Cooperativa de Ahorro y Crédito del Ecuador que encuentra en el segmento 2 dentro de la Superintendencia de Economía Popular y Solidaria, activos mayor a 20'000.000,00 hasta 80'000.000,00

Publicaciones en Revistas

- **Simplifying Credit Scoring Rules using LVQ+PSO.** Lanzarini, Villa Monte, Fernández Bariviera, Jimbo Santana. *Kybernetes*. Vol.46 No 1. pp.8-16.
 - Impact Factor: 0.98. Cuartil 3 Computer Science, Cybernetics
 - SJR 0.29. Cuartil 2 Computer Science (miscellaneous).
 - Número de citas: 11 (Scopus). 7 (Web of Science)
- **Analysis of Methods for Generating Classification Rules Applicable to Credit Risk.** Jimbo Santana, Villa Monte, Rucci, Lanzarini, Fernández Bariviera. *Journal of Computer Science & Technology*; vol. 17, No 1. pp.20-28. 2017
 - Indexado en Emerging Sources Citation Index
 - Número de citas: 1 (Web of Science)
- **Data Mining Methods linked to Artificial Intelligence to Credit Risk.** Jimbo Santana, Lanzarini, Villa Monte, Fernández Bariviera. *FIGEMPA*. Universidad Central del Ecuador: Año V, volumen 2, No 8. 2017
 - Indexada en Latindex
- **Fuzzy Credit Risk Scoring Rules using FRvarPSO,** Jimbo Patricia, Lanzarini Laura, Bariviera Aurelio, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol 26, No 1. pp.39-57. 2018
 - Impact Factor: 1.286. Cuartil 3 Computer Science, Artificial Intelligence
 - SJR 0.39. Cuartil 2 Artificial Intelligence.
 - Número de citas: 2 (Scopus). 1 (Web of Science)
- **Variations of Particle Swarm Optimization for Obtaining Classification Rules Applied to Credit Risk in Financial Institutions of Ecuador.** Patricia Jimbo Santana, Laura Lanzarini, Aurelio Fernandez Bariviera. *Risks* Vol 8, No.2. 2020
 - Indexado en Emerging Sources Citation Index
 - Indexado en Scopus. CiteScoreTracker 2019: 0.89.

Capítulos de Libro

- **Obtaining Classification Rules Using LVQ+PSO: an application to Credit Risk.** Lanzarini, Villa Monte, Fernández Bariviera, Jimbo Santana. *Advances in Intelligent Systems and Computing*. Springer-Verlag Berlin Heidelberg. pp 383-391. ISBN 978-3-319-19704-3. DOI 10.1007/978-3-319-19704-3_31
 - Indexado en Scopus

- Número de citas 7 (Scopus).
- **Extraction of Knowledge with Population-Based Metaheuristics Fuzzy Rules Applied to Credit Risk.** Patricia Jimbo Santana, Laura Lanzarini, Aurelio Fernández-Bariviera. *Advances in Swarm Intelligence*. Springer. pp 153-163. 2018. ISBN 978-3-319-93817-2. DOI 10.1007/978-3-319-93818-9_15
 - Indexado en Scopus
 - Número de citas 4 (Scopus).
- **FRvarPSO: A Method for Obtaining Fuzzy Classification Rules Using Optimization Techniques,** Patricia Jimbo Santana, Laura Lanzarini, Aurelio Fernández-Bariviera, *Modelling and Simulation in Management Sciences*, Springer International Publishing, Cham, pp 112–126, 2020 DOI: 10.1007/978-3-030-15413-4_9
 - Indexado en Scopus

Congresos con referato internacional

- **An exploratory analysis of methods for extracting credit risk rules.** Jimbo Santana, Villa Monte, Rucci, Lanzarini, Bariviera. XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016). pp. 834-841.
- **Fuzzy Classification Rules with FRvarPSO Using Various Methods for Obtaining Fuzzy Sets .** Patricia Jimbo Santana, Laura Lanzarini, Aurelio Fernández Bariviera. The 12th International Conference on Computer Science and Information Technology (ICCSIT December 2019)

Estructura de la tesis

La tesis está desarrollada en seis capítulos según el siguiente detalle:

- **Capítulo 1 - Reglas de clasificación** En este capítulo la investigación se centra en la importancia de las reglas de clasificación, como una técnica de Minería de datos caracterizándose por la predicción de patrones de comportamientos futuros. Los mismos pueden ser utilizados para la toma de decisiones, permitiendo la disminución del riesgo (vulnerabilidades) frente a determinadas situaciones. Adicionalmente se indica el marco referencial en el cual se encuentra desarrollada la presente investigación.

- **Capítulo 2 - Lógica Difusa** Se realiza una revisión general del estado del arte relacionado con la lógica difusa introduciendo varias definiciones como conjuntos difusos, función de pertenencia a ser utilizados en las variables lingüísticas. Se realiza una revisión de los modelos híbridos como aquellos que utilizan redes neuronales, técnicas de optimización y lógica difusa utilizados en la cogeneración de reglas difusas.
- **Capítulo 3 - Extracción de reglas utilizando PSO** Se detallan varias soluciones propuestas con PSO, las mismas que sirven de comparación con el método **FRvarPSO** propuesto en el desarrollo de esta tesis.
- **Capítulo 4 - Método Propuesto FRvarPSO** En este capítulo se detalla el método propuesto en esta investigación **FRvarPSO**, el mismo que es un método adaptativo que utiliza lógica difusa. Este método es capaz de extraer un conjunto de reglas de clasificación cuyo antecedente está formado por atributos nominales y/o condiciones difusas, presentando una tasa de acierto aceptable. Se detallan los resultados obtenidos y se compara con versiones de métodos que no utilizan lógica difusa.
- **Capítulo 5 - Caso de estudio - Riesgo Crediticio**, En este capítulo se utiliza el método propuesto en un problema del mundo real aplicado a riesgo crediticio, Se realiza un análisis del proceso para gestionar el riesgo utilizando el proceso de puntuación de crédito (credit scoring). Se detallan las tareas de limpieza y transformación de los datos que se realizó. Se explica el análisis realizado para determinar la morosidad del cliente, considerando la cartera castigada, vencida y aquella que no devenga intereses. Posteriormente se analizan las variables micro y macro económicas que se utilizaron. Se indican los resultados obtenidos en la aplicación de los diferentes métodos haciendo énfasis en la precisión, finalmente se indican las conclusiones a las que se llegó.
- **Capítulo 6 - Conclusiones Finales y Líneas de Trabajo Futuro** Aquí se exponen las conclusiones obtenidas tras la realización de esta investigación a base de los resultados obtenidos. Finalmente se discuten futuras líneas de investigación.

Abstract

IN recent years thanks to the advancement of technology, organizations have stored a lot of information. This has led them to the need of incorporating techniques to process and obtain information from useful data. The Knowledge Extraction process, known as **KDD** (*Knowledge Discovery in Databases*), is composed by several phases ranging from the collection and transformation of data to the identification and interpretation of patterns or underlying relationships, useful to the decision-making process.

Within the **KDD** process, Data Mining is considered the most important phase, as it groups together the techniques capable of modeling the information. From the use or understanding of the model it is possible extract knowledge. This generated knowledge results of great interest to organizations, since it constitutes an extremely important tool for tactical and strategic decisions, which becomes a competitive advantage.

Depending on the type of task to be solved there are a lot of techniques available in the literature, among which we can mention: algebraic, statistics, stochastic, fuzzy and Bayesian techniques, decision trees, artificial neural networks, support vector machines, and optimization techniques, among others. Each of them has different characteristics.

In order to select the suitable technique, it is necessary to know the problem to be solved. Not all techniques can be used with missing data, outliers or heterogeneous data format. On the other hand, the problems to be addressed could be of different types: classification, categorization, regression or grouping.

A desirable feature of the models is to extract knowledge in understandable terms. In this sense, association rules are considered by decision makers, as one of the most understandable forms that can be used to represent knowledge, since they have the ability to explain themselves. An association rule is an expression of the form "*IF condition1 THEN condition2*" or where both conditions are conjunctions (attribute=value) whose attributes/variables can be quantitatives or qualitative. Because of its conditional structure, it may be stated that the primary purpose of a rule is to identify relationships between attributes. When the association rule set presents in the consequent the same attribute, it is said to be "*a classification rule*". If you add to this that classification rules use fuzzy logic through fuzzy sets to describe the values of their attributes, you get "*Fuzzy Classification Rules*", making your understanding

even easier, bringing us ever closer to human reasoning. As mentioned in (Terano, Asai y Sugeno, 1992), *the more human a system is, the more fuzzy logic it will contain*.

The use of fuzzy logic for the generation of rules has allowed researchers to improve the performance of metaheuristics, basically for convergence, obtaining a better quality in solutions.

Fuzzy classification rules are rules that have linguistic statements that describe how fuzzy inference systems make decisions about a set of input variables. These classification rules have an antecedent formed by nominal variables and/or fuzzy conditions, and a consequent (known beforehand) called "class".

The central contribution of this thesis is the definition of a new method able of generating a set of fuzzy classification rules of easy interpretation, low cardinality and good accuracy. These features help to identify and understand the relationships presented in data, facilitating decision making.

The new proposed method is called **FRvarPSO** (**Fuzzy Rules variable Particle Swarm Optimization**). It combines a competitive neural network with an optimization technique based on variable population particles clusters to obtain fuzzy classification rules. It is capable of working with nominal and numerical attributes. The antecedents of the rules are made up of nominal attributes and / or fuzzy conditions. The conformation of the latter requires knowing the membership degree to the fuzzy sets of each linguistic variable. This thesis proposes three different alternatives to solve this point.

Regarding to obtaining the rules, the proposed method uses an iterative process where the examples of a class are covered until the desired coverage is achieved. Therefore, the rule's consequent is determined by the selected class and the antecedent is extracted through the optimization technique.

Each particle in the population uses a fixed-length representation, where only the antecedent of the rule is coded. By means of an iterative process all individuals in the population will be associated with a predetermined class. The fact that all individuals belong to the same class, makes it unnecessary to code the consequent within the particle.

The process of extracting each rule begins with the generation of the cluster. The initial positions of the particles have a high impact on the exploratory capacity of each particle. If they are too close, the social knowledge can lead to premature convergence, preventing good solutions. On the other hand, starting far from the optimal requires a greater number of iterations to achieve it. For all of the above, it was decided to use a competitive neural network to identify the most promising areas. This information was used to initialize the cluster.

Regarding population size, we advocate for a variable size cluster with the capacity to incorporate individuals into the least populated areas. As stated above,

when using PSO particles tend to gather around the good positions found by their neighbors. This situation cause the cluster to lose diversity and several areas of the search space become unexplored. The technique to modify the size of the population works on two concepts: on the one hand it detects dense areas and eliminates very similar individuals and on the other hand identifies unoccupied areas and adds new particles. This allows initialize the process with a few individuals and change the population size during the iterative search process.

Returning to the representation of the particle, the information regarding the movement of the individual is complemented with two characteristics regarding the construction of the antecedent of the rule. First, the attributes that will be part of the antecedent should be indicated. This should be done using binary information. Second, it is necessary to represent the membership degree of the original numeric attributes with the corresponding linguistic variables, through a fuzzy representation.

One of the contributions of this thesis lies in the definition of the fitness function of each particle based on a "*Voting criterion*", that weights the participation of the fuzzy conditions in the formation of the antecedent. Its value is obtained from the degrees of membership of the examples that abide by the rule and is used to reinforce the movement of the particle in the direction where the highest value is located. With the use of PSO the particles compete among them to find the best rule of the selected class.

The efficiency and efficacy of FRvarPSO are strongly driven by the way fuzzy sets membership functions are determined. During the research process of this thesis different options have been used. One of these options was to partition the range of each numeric attribute into intervals of equal length, and centering on each of them a triangular function with an appropriate overlapping. Another way of obtaining the fuzzy sets has been using the Fuzzy C-Means method. In addition, we used also the knowledge of an expert for the definition of fuzzy sets and their corresponding membership value.

The performance of the proposed FRvarPSO method was compared with previous versions of the method such as *SOM + PSO*, *SOM + varPSO* (PSO with variable population), *LVQ + PSO*, as well as other classification methods such as PART and C4.5. In addition, FRvarPSO was compared with *SOM + fuzzy PSO*, *SOM + fuzzy var-PSO*, *LVQ + fuzzy PSO*, which are the previous versions that used fuzzy logic when constructing the antecedent together with the voting procedure used in the fitness function.

The experiments were carried out on twelve databases of the UCI repository and three real cases in the credit institutions of Ecuador. In the latter case, the focus was on credit risk. Regarding the databases of the financial system of Ecuador, we obtained anonymized information of a savings and credit cooperative, and the other

databases of banks that provide loans to productive and non-productive consumption, and microcredits. The financial institutions considered are in the market by market since at least 25 years.

FRvarPSO was used in actual databases to produce rules regarding the credit granting process. It is important to highlight the arduous task of cleaning and transforming information, before applying the methods. The distribution function of each attribute was analyzed to detect abnormal, missing, and/or outlier values. Variables were transformed in order to obtain suitable indicators. For example: the capacity to repay the loan, the debt capacity and the credit scoring. The latter had a particular analysis, because we had information about the general characteristics of the customer, as well as the previous credits over a time horizon, obtaining the level of delinquency.

Another contribution of this thesis was to have made a special consideration in the customer's delinquency taking into account the days of maturity of the portfolio; this was possible because the database included information of the customer's subsequent behavior on the credit that had already granted. In this way, three cases were considered: the overdue portfolio, the non-interest portfolio and portfolio at risk .

It was verified that with this analysis the fuzzy rules obtained through FRvarPSO allows the credit officer to solve the customer's request in a shorter time period, and to reduce the risk of the financial institutions. This was possible, because applying a fuzzy rule takes the least average degree of membership of the conditions that make up the antecedent of the rule, thus having a metric that is proportional to the risk. With this information, the credit officer can decide to grant the credit at an increased interest rate or to require collateral, in order to control the associated risk.

The results of FRvarPSO were benchmarked with the other methods, verifying that the fuzzy models have in most cases higher accuracy than PART method, but slightly higher than that achieved by C4.5 method. However, for a similar level of accuracy, the number of rules to achieve of fuzzy methods is significantly smaller than the number of rules generated by C4.5 and PART methods. This confirms our *emphasis on the model's simplicity* and its straightforward interpretation, thanks to the use of fuzzy logic.

Figure 7 shows a comparison of the results of the different methods applied to 12 databases in the UCI repository and the 3 financial databases from Ecuador. We can observe the relationship between the accuracy of each of the methods with the best solution. This value corresponds to the ratio between the average accuracy reached by a method and the best accuracy. It shows that the proposed method offers better precision than the other methods under analysis.

Figure 8 displays the relationship between the cardinality (number of rules) of each of the methods, with the best cardinality of each database. Such relationship is

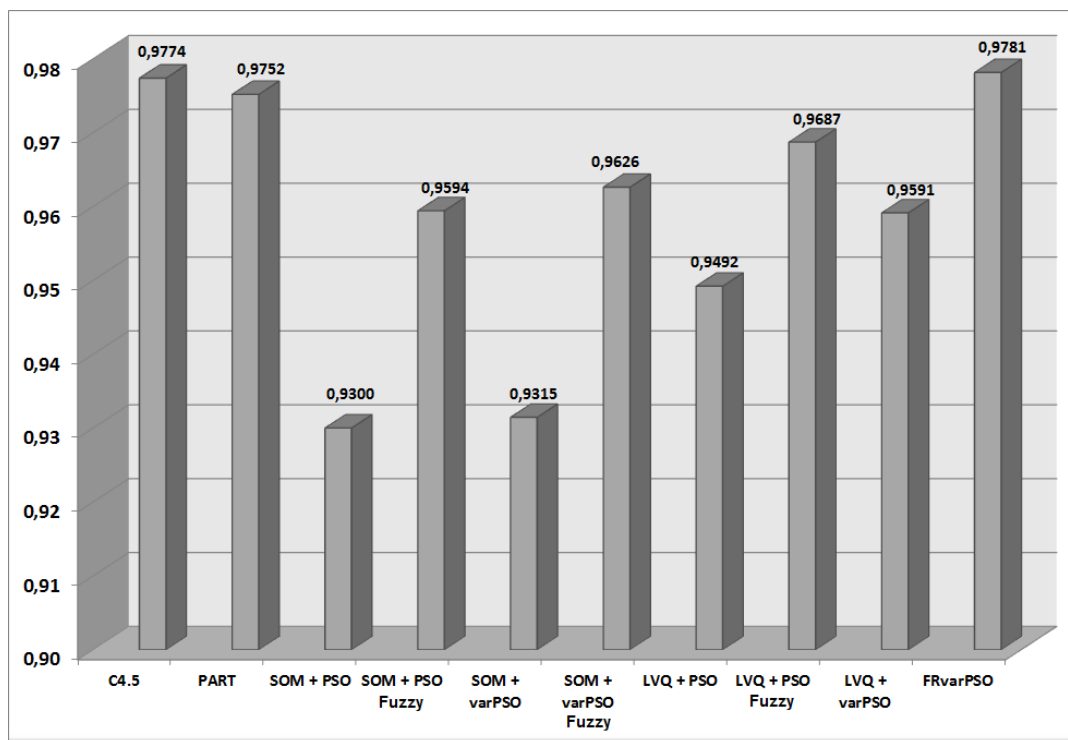


FIGURA 7: Compararion of the accuracy of the different methods.

defined as the quotient between the average number of fuzzy rules and the rules generated by the best method. Our results allow us to affirm that the proposed method is the one with the lowest number of rules.

Figure 9 displays the relationship between accuracy and cardinality of each method under examination.

Figures 10, 11, and 12 shows us the relationship between accuracy and number of rules in the three databases of financial institutions from Ecuador, using FRvarPSO, and generating fuzzy sets by means of equitable partitions, fuzzy C-means, and expert knowledge.

Finally, we find that the use of FRvarPSO, with FCM to determine the fuzzy sets, generates fuzzy rules with greater precision, less cardinality and easy interpretation, than when using equitable sets. Considering the global results obtained, we can say that the goals of the thesis has been fulfilled.

Keywords: Fuzzy Classification Rules, Variable Particle Swarm Optimization, Data Mining.

Publications

In the development of this thesis, the following publications were produced:

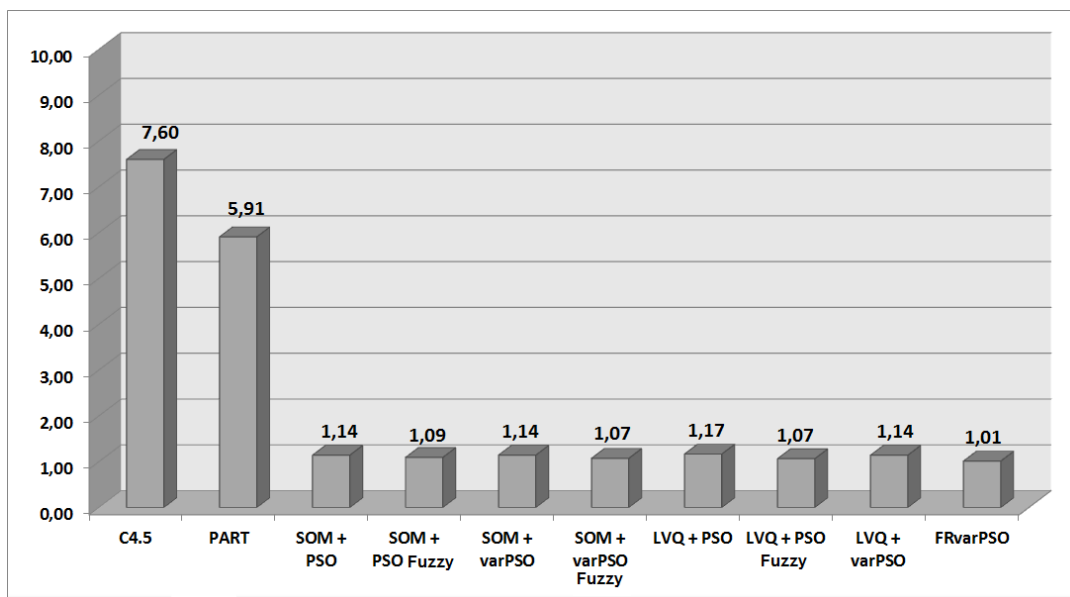


FIGURA 8: Comparison of the cardinality of the methods.

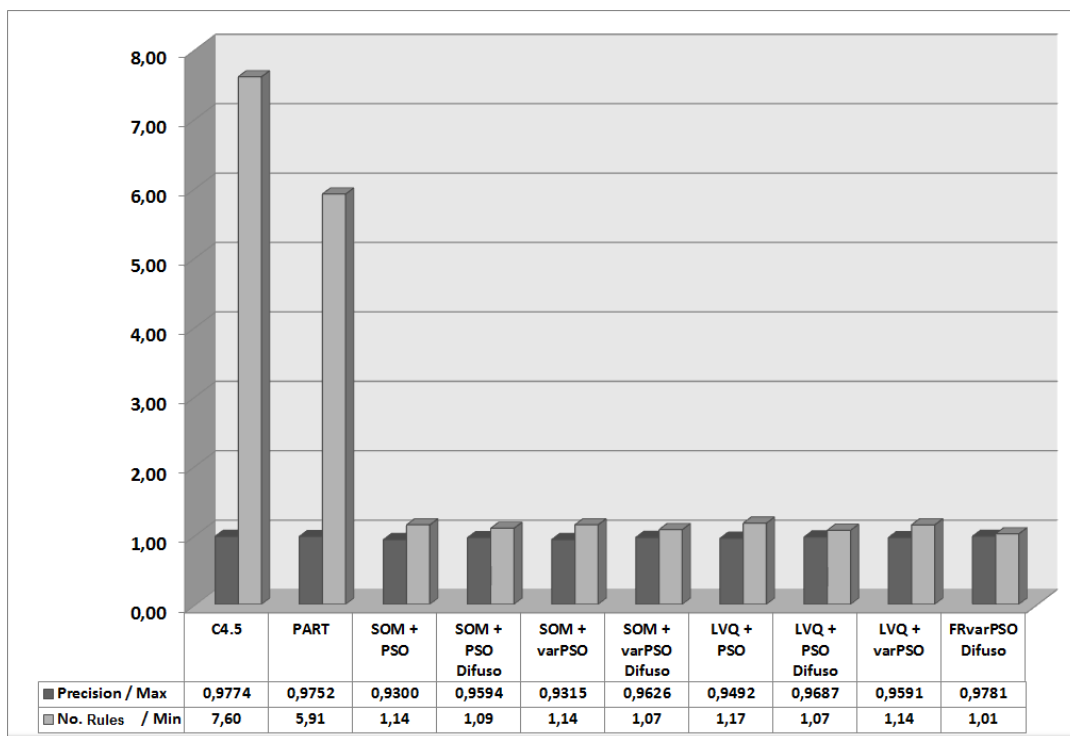


FIGURA 9: Relationship of accuracy and cardinality of each method under analysis.

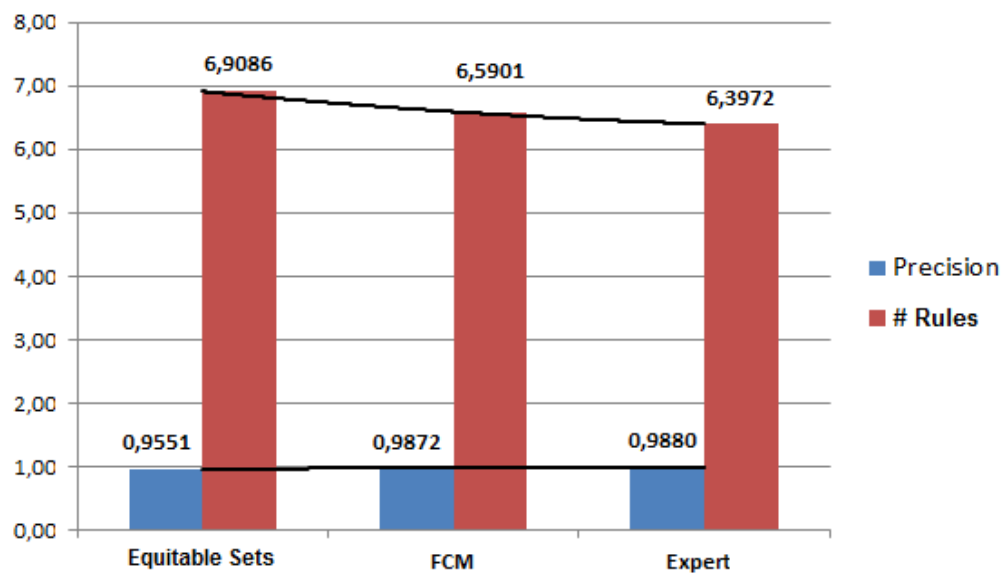


FIGURA 10: Bank dedicated to microcredits.

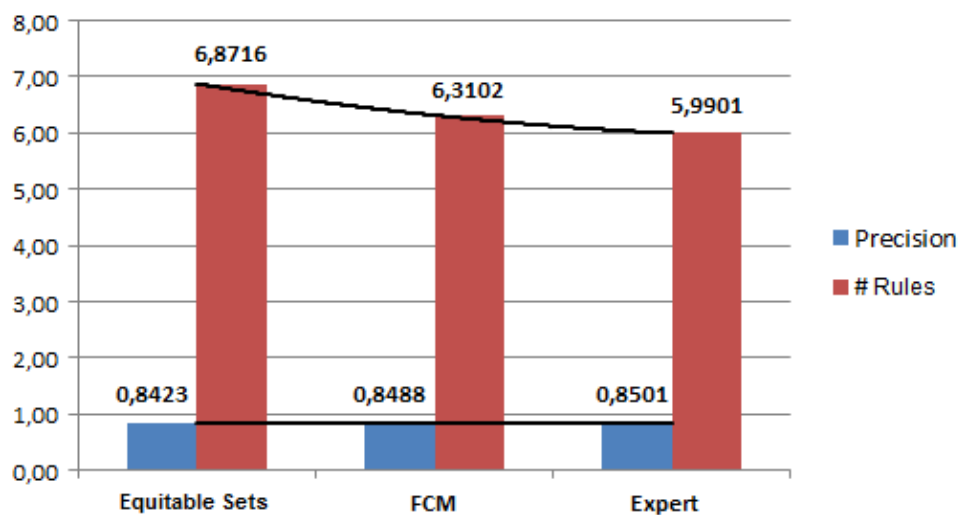


FIGURA 11: Bank dedicated to consumption and production credits.

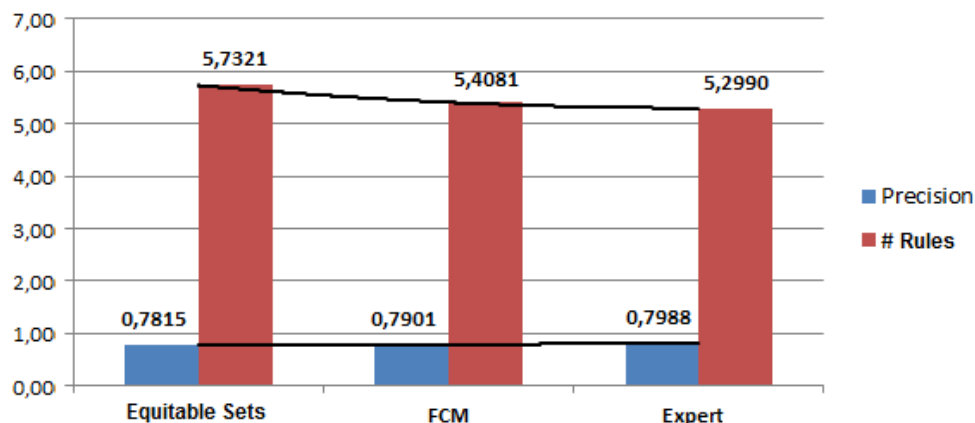


FIGURA 12: Savings and credit cooperative from segment 2 of the Superintendency of Popular and Solidarity Economy, with assets between 20 and 80 USD millions.

Publications in refereed journals

- **Simplifying Credit Scoring Rules using LVQ+PSO.** Lanzarini, Villa Monte, Fernández Bariviera, Jimbo Santana. *Kybernetes*. Vol.46 No 1. pp.8-16.
 - Impact Factor: 0.98. Quartile 3 Computer Science, Cybernetics
 - SJR 0.29. Quartile 2 Computer Science (miscellaneous).
 - Number of citations: 11 (Scopus). 7 (Web of Science)
- **Analysis of Methods for Generating Classification Rules Applicable to Credit Risk.** Jimbo Santana, Villa Monte, Rucci, Lanzarini, Fernández Bariviera. *Journal of Computer Science & Technology*; vol. 17, No 1. pp.20-28. 2017
 - Indexed in Emerging Sources Citation Index
 - Number of citations: 1 (Web of Science)
- **Data Mining Methods linked to Artificial Intelligence to Credit Risk.** Jimbo Santana, Lanzarini, Villa Monte, Fernández Bariviera. *FIGEMPA*. Universidad Central del Ecuador: Año V, volumen 2, No 8. 2017
 - Indexed in Latindex
- **Fuzzy Credit Risk Scoring Rules using FRvarPSO,** Jimbo Patricia, Lanzarini Laura, Bariviera Aurelio, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol 26, No 1. pp.39-57. 2018
 - Impact Factor: 1.286. Quartile 3 Computer Science, Artificial Intelligence
 - SJR 0.39. Quartile 2 Artificial Intelligence.
 - Number of citations: 2 (Scopus). 1 (Web of Science)

- **Variations of Particle Swarm Optimization for Obtaining Classification Rules Applied to Credit Risk in Financial Institutions of Ecuador.** Patricia Jimbo Santana, Laura Lanzarini, Aurelio Fernandez Bariviera. *Risks* Vol 8, No.2. 2020
 - Indexed in Emerging Sources Citation Index
 - Indexed in Scopus. CiteScoreTracker 2019: 0.89.

Book chapters

- **Obtaining Classification Rules Using LVQ+PSO: an application to Credit Risk.** Lanzarini, Villa Monte, Fernández Bariviera, Jimbo Santana. *Advances in Intelligent Systems and Computing*. Springer-Verlag Berlin Heidelberg. pp 383-391. ISBN 978-3-319-19704-3. DOI 10.1007/978-3-319-19704-3_31
 - Indexed in Scopus
 - Number of citations: 7 (Scopus).
- **Extraction of Knowledge with Population-Based Metaheuristics Fuzzy Rules Applied to Credit Risk.** Patricia Jimbo Santana, Laura Lanzarini, Aurelio Fernandez-Bariviera. *Advances in Swarm Intelligence*. Springer. pp 153-163. 2018. ISBN 978-3-319-93817-2. DOI 10.1007/978-3-319-93818-9_15
 - Indexado in Scopus
 - Number of citations: 4 (Scopus).
- **FRvarPSO: A Method for Obtaining Fuzzy Classification Rules Using Optimization Techniques,** Patricia Jimbo Santana, Laura Lanzarini, Aurelio Fernández-Bariviera, *Modelling and Simulation in Management Sciences*, Springer International Publishing, Cham, pp 112–126, 2020 DOI: 10.1007/978-3-030-15413-4_9
 - Indexed in Scopus

International conferences

- **An exploratory analysis of methods for extracting credit risk rules.** Jimbo Santana, Villa Monte, Rucci, Lanzarini, Bariviera. XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016). pp. 834-841.
- **Fuzzy Classification Rules with FRvarPSO Using Various Methods for Obtaining Fuzzy Sets .** Patricia Jimbo Santana, Laura Lanzarini, Aurelio Fernandez Bariviera. The 12th International Conference on Computer Science and Information Technology (ICCSIT December 2019)

Estructure of the thesis

The thesis is developed in six chapters according to the following detail:

- **Chapter 1 - Classification rules.** This chapter focuses on the importance of classification rules, as a data mining technique to characterize patterns of future behavior. This patterns can be used for decision making, allowing the reduction of risk regarding some situations. In addition, we include the theoretical framework in which this research is developed.
- **Chapter 2 - Fuzzy logic.** We conduct a literature review on fuzzy logic, introducing the definitions of fuzzy sets and membership functions to be used with linguistic variables. We perform a review of hybrid models that used neural networks, optimization techniques and fuzzy logic for the cogeneration of fuzzy rules.
- **Chapter 3 - Extraction of rules using PSO.** We detail the different solutions proposed with PSO, which serve as benchmarks to the proposed **FRvarPSO** method.
- **Chapter 4 - Proposed method: FRvarPSO.** This chapter details the proposed method. Our proposal is able to extract a set of classification rules, whose antecedent is formed by nominal attributes and/or fuzzy conditions, presenting an acceptable accuracy rate. We detail the results obtained from the application of this method to different repository databases. We compare our results with the crisp version of the method.
- **Chapter 5 - Case study - Credit Risk.** This chapter applies the proposed method to a real world problem: credit risk. We analyze the process of risk management under the perspective of credit scoring. We detail the cleaning and transformation of data from the original raw data. We explain the analysis to determine the customer's delinquency, considering the portfolio at risk, overdue portfolio, and the non-interest portfolio. Subsequently, we analyze micro and macroeconomic variables used in our model. We detail the results obtained with the application of the different methods, measuring the accuracy in the classification. Finally, we draw the main conclusions.
- **Chapter 6 - Conclusions and future research lines.** This chapter exposes the main conclusions from our research, based on the obtained results. Finally, we discuss some future research lines.

Capítulo 1

Reglas de Clasificación

EN la actualidad, son numerosas las áreas interesadas en extraer conocimiento útil y novedoso a partir de información almacenada. Disponer de herramientas capaces de representar de una manera descriptiva las decisiones tomadas, resultan sumamente útiles para explicar lo ocurrido y elegir el camino a seguir ante nuevas situaciones. Cuanto más simple de comprender sea dicha representación, más fácil será decidir cómo y cuando utilizarla. Entre las diferentes técnicas existentes en la literatura, las reglas de clasificación son las preferidas a la hora de establecer relaciones ya que poseen un formato fácil de comprender e interpretar.

Este capítulo se centra en las reglas de clasificación con el objetivo de brindar un marco, desde el estado del arte, al método propuesto en esta tesis. Finalmente se definen distintas técnicas relacionadas con la extracción de reglas de clasificación.

1.1. Minería de datos y el proceso de KDD

El proceso de Extracción del Conocimiento, conocido como proceso de **KDD** es empleado para describir el proceso total de descubrimiento y extracción de conocimiento nuevo, que no resulta obvio a partir de un conjunto de datos. Este se encuentra conformado por relaciones y patrones entre los elementos que conforman los datos. El proceso de **KDD** es el conjunto de tareas no triviales capaces de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia comprensible a partir de los datos (Fayyad, Piatetsky-Shapiro y Smyth., 1996). Es decir que su objetivo es analizar grandes cantidades de datos, encontrar relaciones o patrones, e interpretar los resultados. Contar con técnicas que faciliten la detección de asociaciones entre patrones y ayuden a visualizar la manera en la que se organizan los datos constituye una herramienta de suma utilidad para la toma de decisiones.

El proceso de **KDD** abarca varias etapas en su realización que van desde la selección de los datos hasta la visualización y análisis de los resultados obtenidos.

La **Minería de Datos (DM)** es la etapa más relevante del proceso de **KDD**. Se encarga de reunir un conjunto de técnicas capaces de construir modelos, que procesan

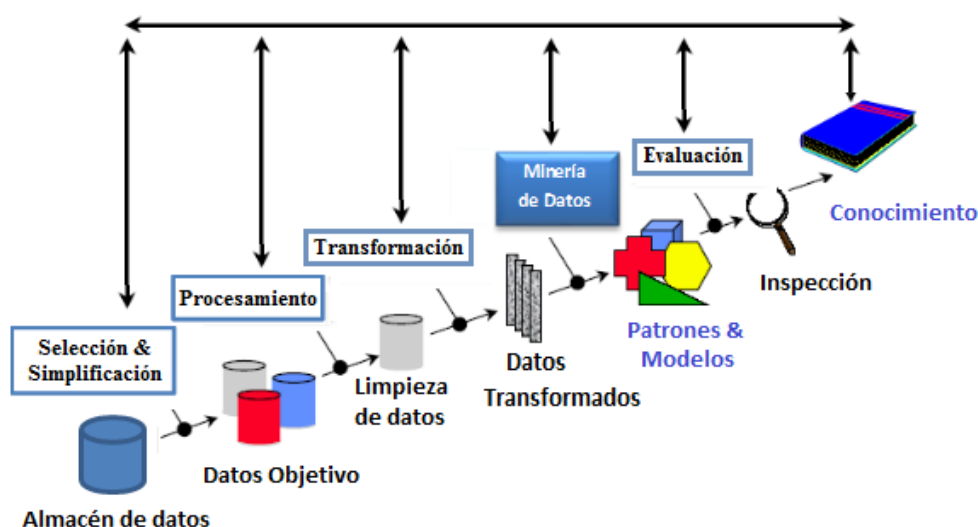


FIGURA 1.1: Etapas del proceso de **KDD**

la información, con el objetivo de identificar relaciones o patrones sin necesidad de una hipótesis previa. Su importancia es tal que en la literatura se le da su nombre al proceso de **KDD** completo. Es por eso que cuando se extrae conocimiento de los datos se suele hablar de que se está llevando adelante un proceso de **DM**.

Las técnicas de **DM** utilizadas tienen la capacidad de descubrir relaciones y/o patrones significativos de información que ayudan tanto en el diagnóstico correcto de un problema como en la formulación de las estrategias para la solución. (Fayyad, Piatetsky-Shapiro y Smyth., 1996), (Hernandez Orallo, Ramirez Quintana y Ramirez, 2004).

Sin embargo, no debe dejarse de lado el valor que tienen las etapas anteriores a la aplicación de la técnica ya que las transformaciones que se efectúen sobre los datos tendrán un efecto decisivo en la eficiencia y eficacia de la solución obtenida. De mismo modo, las etapas finales del proceso de **KDD** son las relacionadas con la interpretación y el análisis de las relaciones encontradas. Estas últimas son fundamentales para acercar el conocimiento adquirido a quienes son los responsables de tomar las decisiones.

En la Figura 1.1 se indican las etapas del proceso de **KDD** (Fayyad, Piatetsky-Shapiro y Smyth., 1996).

1.2. Etapas del proceso de KDD

Como se pudo observar en la Figura 1.1, el proceso de **KDD** tiene una secuencia de etapas. Las primeras se relacionan con la preparación de los datos a base de las necesidades de la técnica a utilizar. Luego se aplica dicha técnica y finalmente se

analizan y comunican los resultados. A continuación se describen cada una de ellas con mayor detalle.

1.2.1. Vista Minable

Se denomina Vista Minable al conjunto de datos sobre los que se aplicará la técnica de **DM**. Su obtención inicia con la recolección de datos relevantes y tipo de información obtenida de diferentes fuentes, para ser utilizadas en el preprocesado. Se la conoce también como la "preparación" de los datos (Devedzic, 2000). Inicialmente debe determinarse qué datos de la base de datos serán seleccionados para el proceso de extracción de conocimiento. Luego que se ha obtenido la información, se procede a familiarizarse con ella e identificar su procedencia.

En la tarea de recopilación puede decirse que, las bases de datos y las aplicaciones basadas en el **Procesamiento Transaccional de Datos en Línea (OLTP)** se encargan de cubrir las necesidades diarias de información de una organización. Sin embargo, no siempre son suficientes para poder desarrollar funciones como es el caso del análisis, planificación y la predicción. Es por esta razón que, en algunos casos, es necesario obtener datos de otras áreas de la organización. Incluso, es posible que algunos datos no se hayan recolectado aún y sea necesario extraerlos desde otras bases de datos ya sean estas públicas o privadas. Cuando esto ocurre surgen otros problemas entre los que se encuentran, por ejemplo, diferentes formatos de registro, distintos grados de agregación en los datos, claves primarias no coincidentes entre otros.

Es por esto necesario, integrar todos los datos en un almacén de datos (Data Warehouse), que se encarga de recopilar información de diferentes fuentes, unificándola en un único repositorio como se puede observar en la Figura 1.2. El tener un almacén de datos, se puede modelar con una estructura de base de datos multidimensional en la que cada una de las dimensiones corresponde a un atributo o a un conjunto de atributos. Esta representación es la adecuada para el **Procesamiento Analítico de Datos en Línea (OLAP)**.

Es importante tener en cuenta que la calidad del conocimiento descubierto no depende únicamente del algoritmo de **DM** que se utilice si no también de los datos que se involucran. Es por eso que después de la recopilación de la información, viene la preparación de los datos que se van a minar. El resultado de todas las etapas del proceso de **KDD**, previos a la aplicación de la técnica de minería de datos, da como resultado una vista minable.

En esta fase, se incluye la selección de las tablas o archivos, registros y atributos, así como la limpieza y transformación de los datos. Es aquí donde se deben analizar las funciones de distribución de cada atributo a fin de detectar valores anómalos, faltantes y/o fuera de rango. Luego dependerá del tipo de problema a resolver y de

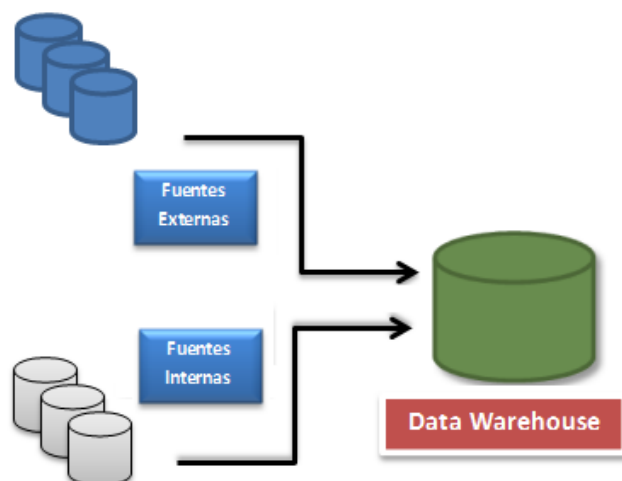


FIGURA 1.2: Integración de datos

la técnica de **DM** a aplicar si dichos valores son ignorados, descartados o transformados.

Por último, se debe tener en cuenta que se deben escoger aquellos atributos que sean relevantes para la tarea de **DM**. Incluso puede mejorarse la capacidad representativa de los datos construyendo nuevos atributos a partir de los existentes.

1.2.2. Minería de Datos

Una vez generada la vista minable se pasa a aplicar la técnica de **DM**. El resultado de esta etapa es la obtención del modelo a través del cual será posible identificar patrones y relaciones presentes en los datos. Esta información permitirá que el analista empresarial ejercite su habilidad y experiencia en la materia, para separar los hechos útiles de los inútiles. (Tan, Steinbach y Kumar, 2005).

Esta etapa es la más característica del proceso de **KDD**. Existen varias técnicas aplicables a los problemas de **DM**, algunas de ellas tienen requerimientos específicos en el formato de los datos, por lo que se establece una estrecha relación entre el preprocesamiento utilizado para construir la vista minable y la técnica de **DM** a emplear.

Es necesario tener en cuenta que, antes de la construcción del modelo, se debe reservar una parte de los datos para verificar su efectividad. Eso da lugar a dos subconjuntos de datos: el *conjunto de entrenamiento* utilizado para construir y entrenar el modelo y el *conjunto de test* con el que se va a evaluar dicho modelo verificando su efectividad.

Cuando se trata de construir un modelo predictivo, como es el caso del propuesto en esta tesis, el desbalance de clases puede llegar a ser un problema. Para ajustar la

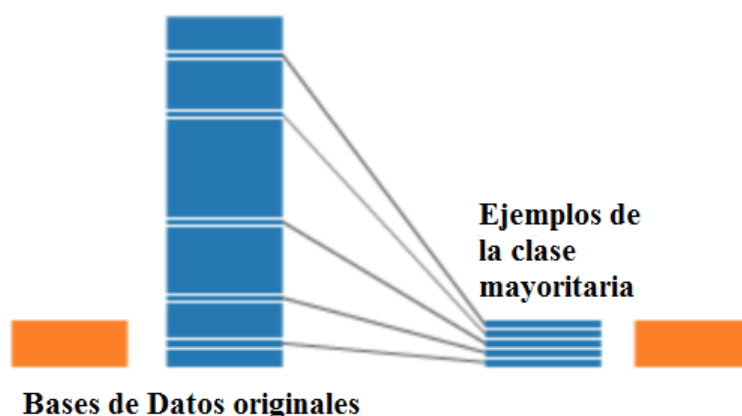


FIGURA 1.3: Submuestreo

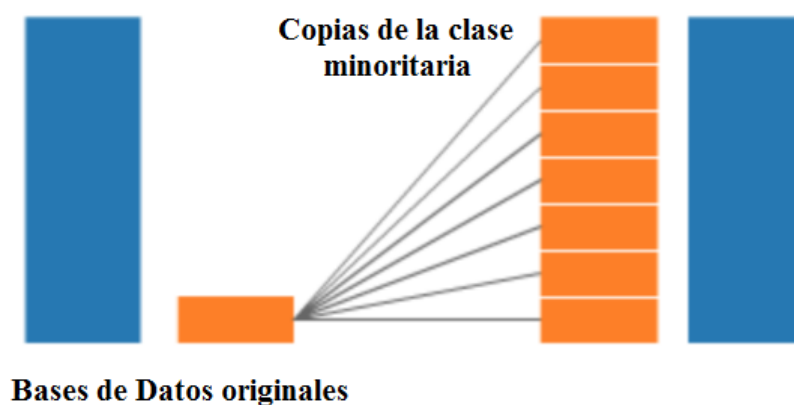


FIGURA 1.4: Sobremuestreo

distribución de un conjunto de datos suele utilizarse el submuestreo o el sobremuestreo. En ambos casos, se trata de introducir un sesgo para seleccionar más muestras de una clase que de otra, a fin de compensar un desequilibrio que ya está presente en los datos, o que probablemente se desarrollará si se toma una muestra puramente aleatoria.

- Submuestreo (undersampling): se caracteriza por eliminar al azar algunos de los casos de la clase mayoritaria, de tal forma que coincida el número de casos que se tiene de la clase minoritaria, como se observa en la Figura 1.3
- Sobremuestreo (oversampling): Se agregan ejemplos de la clase minoritaria hasta tener una cantidad similar a la clase mayoritaria, como se observa en la Figura 1.4

Tal es el caso de de **Synthetic Minority Over sampling (SMOTE)**, que consiste en aplicar un paso de preprocesamiento para volver a muestrear los datos de entrenamiento, que forma nuevos ejemplos de clases minoritarias al interpolar entre varios ejemplos de clases de minorías vecinas (Fernández y col., 2018).

Llegado el momento de aplicar la técnica de **DM** existen en la literatura distintas opciones dentro de un mismo tipo de problema a resolver. Por ejemplo, si se va a emplear una técnica de optimización se deberá decidir si se trabajará con algoritmos genéticos, colonia de hormigas, cúmulo de partículas o enjambre de abejas, entre otros. Dentro de un mismo tipo de técnica pueden presentarse variaciones y no existe una estrategia determinística establecida a priori aplicable a cualquier tipo de problema. Será el analista de datos quien decidirá cuál es la más conveniente para la situación a resolver. Incluso, luego de obtenido el modelo podrían surgir observaciones que lleven a la necesidad de introducir modificaciones en las etapas previas. Es por esto que suele ser en este punto donde se ve reflejado, con mayor claridad, el carácter iterativo del proceso de **KDD**.

En lo que se refiere a la evaluación del modelo generado, existen básicamente dos tipos de métricas a emplear según la clase de tarea a resolver:

- **Precisión predictiva:** Se utiliza en modelos predictivos y busca establecer la cantidad de coincidencia entre el valor predicho y el esperado ya sea sobre el conjunto total de datos de prueba o para cada valor de clase por separado
- **Medidas de distancia:** Se emplea en tareas descriptivas. En este caso, se utiliza alguna medida de distancia calculada entre los elementos de un mismo grupo y entre pares de grupos. Se espera que un ejemplo se encuentre más próximo (según la medida utilizada) a los que pertenecen a su mismo grupo que a los que pertenecen a otro.

El modelo se considerará adecuado si cumple con el criterio establecido para medir su eficacia. Esto dependerá del problema a resolver.

1.2.3. Interpretación, Difusión y uso de los resultados

Finalizada la construcción del modelo, el conocimiento que se obtiene a partir de él debe ser organizado y presentado de manera que pueda ser entendido y comprendido por el usuario final para la toma de decisiones. Esta etapa puede ser tan simple como la generación de un informe o tan compleja como la aplicación del modelo a otro grupo de datos, llegando a implementar procesos iterativos de **DM**, tantas veces como sea necesario hasta llegar al cumplimiento de los objetivos.

El usuario debe entender los resultados, y utilizar el modelo creado; es aquí donde las investigaciones realizadas en el marco de esta tesis presentan su aporte, debido a que el método propuesto genera un modelo sencillo y de fácil interpretación.

1.3. Tareas de Minería de Datos

Básicamente, puede decirse que existen dos tipos de tareas: predictivas y descriptivas. Resulta fundamental identificar en cuál de ellas se enmarca el problema a resolver antes de iniciar el proceso de **KDD** ya que esta decisión condiciona la elección de la técnica a aplicar y por consiguiente, las operaciones de preprocesamiento que deben efectuarse sobre los datos a la hora de construir la vista minable. A continuación se brinda un mayor detalle de cada una de ellas.

Tareas Predictivas

Son las tareas que se relacionan con la necesidad de predecir un comportamiento; es decir, aquellas que requieren de la obtención de un modelo que permita dar una respuesta, en una etapa posterior, ante la presencia de información nueva. La tarea predictiva puede ser una regresión o una clasificación según si la respuesta a obtener es continua o discreta, respectivamente.

Las técnicas capaces de resolver tareas predictivas requieren de ejemplos o situaciones históricas etiquetadas. Es decir que, se necesita conocer para cada ejemplo o caso anterior cuál fue la respuesta dada en ese momento. Luego, la técnica a aplicar utilizará estos ejemplos etiquetados y tendrá por objetivo reducir la brecha entre las respuestas obtenidas a través del modelo y las respuestas esperadas (valores de respuesta reales indicados en los ejemplos).

Tareas Descriptivas

El objetivo de las tareas descriptivas es identificar relaciones en los datos que ayuden a explicar su organización. Estas relaciones son sumamente útiles para hallar características distintivas entre grupos de casos o ejemplos de la vista minable.

A diferencia de las tareas predictivas, si lo que se busca es una descripción de cómo están organizados los datos disponibles, no será necesario contar con ejemplos etiquetados. Es decir que el objetivo en este caso es medir similitud entre ejemplos en lugar de pertenencia a una clase dada.

1.4. Técnicas de Minería de Datos

Según el tipo de tarea que se quiera resolver, existe en la literatura una gran cantidad de técnicas disponibles. A continuación se mencionan brevemente las más utilizadas:

- **Algebraicas y estadísticas:** Expresan patrones y modelos mediante funciones lineales, no lineales, fórmulas algebraicas, entre otros. Dentro de este grupo se

puede mencionar a la regresión lineal, regresión logarítmica, regresión logística, discriminante lineal y no lineal como las más importantes. Son utilizadas básicamente cuando los atributos originales son en su mayoría cuantitativos, en el caso de variables cualitativas es necesario su transformación en binarias.

- **Bayesianas:** Estiman la probabilidad de pertenencia a una clase o un grupo utilizando el teorema de Bayes.
- **Conteo de frecuencias y tablas de contingencia:** Se basan en la frecuencia en que dos o más sucesos se presentan de forma conjunta. Un ejemplo de este tipo de técnicas es el algoritmo Apriori (Agrawal y Srikant, 1994)
- **Arboles de decisión:** Se basan en el mecanismo de inducción: "*Divide y Vencerás*", utilizan aprendizaje supervisado para identificar la manera de separar recursivamente los ejemplos hasta lograr subconjuntos suficientemente homogéneos. Dentro de esta categoría, los algoritmos clásicos de la literatura son el ID3 (Quinlan, 1986) o el C4.5 (Quinlan, 1993) entre otros.
- **Redes neuronales:** Son modelos no lineales que operan únicamente sobre información numérica y que poseen la capacidad de aprender relaciones complejas. Los patrones o relaciones aprendidos dependerán de la arquitectura y el proceso adaptativo utilizados.
- **Máquinas de vectores de soporte:** Al igual que las redes neuronales, operan sobre información numérica y son capaces de realizar un mapeo no lineal entre la entrada y la salida esperada. Sin embargo, a diferencia de las redes neuronales, proponen una estrategia para efectuar las distintas transformaciones del espacio de entrada de una manera óptima.
- **Estocásticas y difusas:** En esta categoría se engloban todas aquellas técnicas que junto a las redes neuronales, forman lo que se conoce como computación flexible. Son técnicas en las que los componentes aleatorios son fundamentales como es el caso de los métodos evolutivos y genéticos. Una característica de las técnicas estocásticas es su costo computacional, en general requieren bastante tiempo y memoria para llegar a una convergencia. En el caso de las técnicas difusas una de sus características es la expresividad, utilizan funciones de pertenencia, lo que hace que sus fronteras no sean rígidas.

No todas las técnicas mencionadas anteriormente generan modelos con iguales características. Todo depende del objetivo de base. Los modelos predictivos hacen hincapié en la performance mientras que los descriptivos ofrecen la posibilidad de explicar el resultado obtenido.

El objetivo de esta tesis es reunir ambas características al proponer una nueva estrategia para generar un modelo predictivo que sea fácil de interpretar. Dado que dicho modelo se encuentra basado en reglas, en la próxima sección se realizará una

breve revisión de los aspectos básicos. Las Reglas son una de las técnicas más utilizadas por su facilidad en la interpretación del conocimiento. Esta es una de las razones por las cuales fueron consideradas como centro en el desarrollo de esta investigación.

1.5. Reglas para extracción del conocimiento

Tal como se dijo anteriormente, las investigaciones realizadas a lo largo del desarrollo de esta tesis se enmarcan en la resolución de problemas predictivos buscando ofrecer una solución que, por sí misma, posea una buena capacidad explicativa. Entre los distintos modelos existentes, se ha trabajado en la generación de un conjunto de reglas de clasificación, ya que si se las compara con otras técnicas como las redes neuronales o las máquinas de vectores de soporte, que son consideradas cajas negras y difíciles de explicar, las reglas son de fácil interpretación.

1.5.1. Reglas de asociación

Se denomina regla de asociación a una expresión condicional de la forma:

$$SI (condicion_1) ENTONCES (condicion_2)$$

donde $condicion_1$ y $condicion_2$ son conjunciones de proposiciones de la forma "atributo = valor" donde la única restricción es, que los atributos que intervienen en el antecedente de la regla no formen parte del consecuente. Los atributos pueden ser cualitativos, o cuantitativos, siendo necesario indicar para estos últimos el rango a utilizar, ya sea a través de un intervalo o de una cota superior o inferior.

La definición de (Agrawal, Imieliński y Swami, 1993) de una regla de asociación es una implicación de la forma $X \Rightarrow Y$, donde $X, Y \subset I$ y $X \cap Y = \emptyset$, entendiéndose con ello que toda transacción que satisface a X también satisface a Y . Cada par de la forma (atributo, valor) recibe el nombre de *ítem*. Dado el gran volumen de datos de los problemas con el que los algoritmos de aprendizaje de reglas de asociación trabajan, la tarea de buscar patrones que cumplan estos requisitos puede parecer muy costosa, ya que los conjuntos de ítems a ser analizados crece exponencialmente con respecto al número de variables de los datos. Sin embargo, por fortuna, en los casos reales existen realmente pocos conjuntos frecuentes y los métodos que exigen una confianza y/o cobertura mínimas se benefician de este hecho. Por ejemplo, la mayoría de los clientes de un supermercado suelen comprar un número bastante limitado de productos.

Para poder comparar las reglas obtenidas, se utilizan métricas. Entre las más habituales se encuentran el soporte y la confianza.

El *soporte* es el número de instancias a las que la regla se aplica y predice correctamente y se calcula según la ecuación (1.1):

$$Sup(X \Rightarrow Y) = \frac{\|X \cup Y\|}{\|D\|} \quad (1.1)$$

siendo D el conjunto total de ejemplos de la vista minable y $\|D\|$ su cardinalidad.

La *confianza* es la proporción de instancias que la regla predice correctamente. Es decir que, su valor se calcula como el cociente entre el soporte de la regla y el soporte del antecedente tal como se indica en la ecuación (1.2).

$$Conf(X \Rightarrow Y) = \frac{SupX \Rightarrow Y}{SupX} = \frac{\|X \cup Y\|}{\|X\|} \quad (1.2)$$

El problema de extraer reglas de asociación se puede dividir en dos subproblemas:

- Generar combinaciones de ítems que cumplan con un soporte mínimo establecido a priori. Estas combinaciones suelen encontrarse en la literatura con el nombre de *ítemsets frecuentes*.
- Dado un ítemset frecuente $I = \{i_1; i_2; \dots; i_k\}$; $k \geq 2$, generar todas las reglas que contengan todos los ítems de ese ítemset. Para ello, se toman todos los subconjuntos no vacíos X de Y y se generan las reglas $X \Rightarrow Y - X$ que cumplan con la métrica de calidad establecida a priori. Por ejemplos, podría considerarse importante que la confianza de las reglas generadas supere un cierto umbral.

Un algoritmo de aprendizaje de reglas de asociación muy simple y popular es el algoritmo *Apriori* (Agrawal y Srikant, 1994). El funcionamiento de este algoritmo, se basa en la búsqueda de los conjuntos de ítems con determinada cobertura. Para ello, en primer lugar se construyen los conjuntos formados por sólo un ítem que superan la cobertura mínima. Este conjunto de conjuntos se utiliza para construir el conjunto de conjuntos de dos ítems, y así sucesivamente hasta que se llegue a un tamaño en el cual no existan conjuntos de ítems con la cobertura requerida. Una vez que se han seleccionado los conjuntos de ítems, que cumplen con la cobertura mínima, el siguiente paso consiste en extraer de estos conjuntos de reglas las que cumplan con el criterio establecido; por ejemplo, que tengan un nivel de confianza mínimo.

Por lo tanto, se puede decir que el aprendizaje de reglas de asociación se divide normalmente en dos fases: la extracción de los conjuntos de ítems que cumplan con la cobertura requerida desde los datos, y la generación de las reglas a partir de estos conjuntos.

Con el objetivo de mejorar la búsqueda de reglas de asociación, así como ampliar el campo de aplicación de estas estrategias, se han definido diferentes variantes del

algoritmo básico. Estas variantes se han basado en el perfeccionamiento de los puntos clave que constituyen un mayor consumo de recursos. Por ejemplo, existen en la literatura variantes que tienen por objetivo minimizar el acceso a la base de datos para recabar información sobre ítems. Una posible opción es la utilización de tablas Hash (Sit y Morris, 2002) o la minimización del número de ejemplos a ser inspeccionados para el aprendizaje de las reglas (Han, Pei y Yin, 2000), entre otras. Dado que las reglas de asociación buscan expresar relaciones entre ítems frecuentes, parece claro que trabajar sobre una muestra de ejemplos representativos de la base de datos, permitiría obtener resultados parecidos a trabajar sobre todos los datos, y al mismo tiempo se reduciría de manera considerable el tiempo de aprendizaje. Un desarrollo de esta técnica de muestreo para el aprendizaje de reglas de asociación, se puede encontrar en (Toivonen, 1996). Otra técnica similar a la utilización de muestreo sobre los ejemplos o registros, es la de realizar un filtro sobre los atributos, para rechazar aquellos que se consideren poco relevantes. A esta técnica se le llama selección de atributos (feature selection).

Una manera mejor de abordar bases de datos con gran volumen de datos, es la utilización de técnicas de paralelización. Estas técnicas consisten en modificar los algoritmos, de manera que puedan ser ejecutados de manera concurrente en máquinas con varios procesadores. Sin embargo las máquinas multi-procesador, son por lo general muy caras, por lo que su utilización está bastante más restringida que las computadoras con único procesador. Los algoritmos que se desarrollaron en esta tesis, se ejecutaron en un servidor con dos procesadores Xeon 5060, realizándose varias ejecuciones.

1.5.2. Reglas de clasificación

Las reglas de clasificación, si bien poseen la misma estructura condicional que las reglas de asociación, se caracterizan por que su consecuente, también llamado conclusión de la regla, identifica al valor de la clase que se aplica a los ejemplos cubiertos por dichas reglas. Por lo tanto, el consecuente de todas las reglas que forman el conjunto de reglas deberá referirse a un mismo atributo o característica, el que corresponde a la respuesta esperada.

Un ejemplo de regla de clasificación podría ser el siguiente:

SI (*Ingreso es alto*) **Y** (*Egreso es bajo*) **ENTONCES** (*conceder credito*)

La obtención de reglas de clasificación es un problema clásico del aprendizaje automático. A partir de ejemplos etiquetados se seleccionan los ítems que conformarán el antecedente de la regla así como el valor de clase a utilizar en el consecuente. A lo largo de todo el proceso el énfasis siempre estará puesto en generar reglas con un soporte y una confianza mínimos, establecidos a priori.

Para la extracción de las reglas de clasificación se pueden considerar dos métodos:

- **Directos:** Se encargan de extraer las reglas, directamente a partir de los datos, aquí se puede citar métodos como ZeroR, OneR, PRISM (Witten y col., 2016)
- **Indirectos:** Se caracterizan por construir un modelo con la información que disponen, luego de lo cual extraen las reglas. Existen métodos capaces de obtener un conjunto de reglas a partir de otros métodos como es el caso de los árboles, o redes neuronales. El método desarrollado en esta investigación se encuentra en esta categoría.

1.6. Métodos para extracción de Reglas de Clasificación

En el estado del arte se encuentran varios métodos para la extracción de Reglas de Clasificación. A continuación se describen aquellos que en el capítulo 4 son compararlos con el método desarrollado en esta tesis.

1.6.1. Árboles

Los árboles son modelos de predicción muy utilizados ya que no sólo cumplen con esta tarea sino que además, por su forma jerárquica ayudan a visualizar la organización de los atributos. Por otro lado, sus ramas determinan el conjunto de reglas que permiten la toma de decisiones. Cada una de las ramas del árbol puede interpretarse como una regla distinta, donde todos los nodos internos en el camino que va desde la raíz a la hoja, definen los términos de la conjunción que constituye el antecedente, y el valor de clase que más veces aparece en la hoja es el utilizado en el consecuente.

Uno de los primeros métodos de construcción de árboles de clasificación fue ID3, definido por (Quinlan, 1986). Posteriormente, su autor lo mejoró convirtiéndolo en el algoritmo C4.5 (Quinlan, 1993), el que permite derivar, a partir de un árbol de decisión no pospodado, un conjunto de reglas. Ambos algoritmos utilizan, durante su construcción, el concepto de entropía de la información (*Última versión del código fuente del C4.5 publicada por su autor*).

Algoritmo C4.5

C4.5 incorpora, entre otras características, la capacidad de trabajar con atributos numéricos. La presencia de atributos con dominio continuo introduce una complejidad mayor en el proceso de descubrimiento de las reglas. También incluye un criterio de poda que permite simplificar el árbol según el nivel de precisión indicado.

C4.5 fue considerado entre los 10 algoritmos que más han influido y mayor uso han tenido en **DM** (Filiberto y col., 2011).

La construcción del árbol es un proceso recursivo que va seleccionando los atributos y separando los ejemplos utilizando los siguientes casos base:

- Si todos los ejemplos pertenecen a la misma clase, se crea una hoja para el árbol eligiendo esa clase.
- Si los atributos proporcionan la misma ganancia de información o esta es igual a cero, se crea un nodo hacia arriba del árbol con el valor esperado. Es decir, se detiene la construcción del árbol a partir de este nodo.

El Algoritmo 1 describe el pseudocódigo de C4.5 (Kotsiantis, 2007). En (Quinlan, 1993) se puede encontrar una descripción más detallada del método. La implementación del método C4.5 ha sido suministrada por el autor y es posible descargarla de forma gratuita. En el capítulo 4, el conjunto de reglas de clasificación obtenido a partir de las ramas del árbol generado con el algoritmo C4.5 es comparado con el método propuesto en esta tesis.

Algoritmo 1: Pseudocódigo del algoritmo C4.5

C4.5 (D)

Input: un conjunto de ejemplos D

begin

$Arbol = \{ \}$

if algún caso base se alcanza **then**

 └ Termina

for cada atributo $a \in D$ **do**

 └ Calcular el valor del criterio al dividir D usando a

a_{mejor} es el mejor atributo según el criterio calculado anteriormente

$Arbol =$ Crear un nodo de decisión que tenga a_{mejor} en la raíz

$D_v =$ subconjuntos de D de cada rama del nodo de decisión anterior

for cada D_v **do**

 └ $Arbol_v = C4.5(D_v)$

 └ Conectar $Arbol_v$ a la correspondiente rama de $Arbol$

 retornar($Arbol$)

La construcción del árbol empieza seleccionando el atributo con mayor **Tasa de Ganancia (Gain Ratio)**. Esta métrica tiene en cuenta no sólo la capacidad del atributo para separar los ejemplos, sino también la cantidad de valores posibles que puede tomar. Si se considera D como el conjunto de datos que se desea evaluar y C la cantidad de clases distintas a las que pueden pertenecer los ejemplos de D . Se denomina a $p(D, j)$ a la proporción de ejemplos de D que pertenecen a la j -ésima clase. La incertidumbre referida a la clase a la cual pertenece un ejemplo de D puede expresarse

como la ecuación (1.3)

$$Info(D) = - \sum_{j=1}^c p(D, j) * \log_2(p(D, j)) \quad (1.3)$$

Un atributo T con opciones mutuamente excluyentes T_1, T_2, \dots, T_k al ser aplicado sobre los ejemplos de D dará lugar a los subconjuntos D_1, D_2, \dots, D_k donde D_i contiene aquellos ejemplos que verifican la condición T_i del atributo. La información ganada por un atributo T que posee k opciones se define de acuerdo a la ecuación (1.4)

$$Gain(D, T) = Info(D) - \sum_{i=1}^k \frac{|D_i|}{|D|} * Info(D_i) \quad (1.4)$$

La ganancia definida en (1.4) se encuentra fuertemente influenciada por la cantidad de opciones del atributo T y toma su valor máximo cuando cada subconjunto D_i está formado por un único elemento. Es necesario tener una métrica que permita penalizar a aquellos atributos que posean muchas opciones. Por lo que se utiliza la entropía, relacionada con la cantidad de particiones que un atributo efectúa sobre los datos de entrada. Esta métrica es *Split Information* que se calcula de acuerdo a la ecuación (1.5)

$$Split(D, T) = - \sum_{i=1}^k \frac{|D_i|}{|D|} * \log_2 \left(\frac{|D_i|}{|D|} \right) \quad (1.5)$$

El incremento del valor de $Split(D, T)$ es proporcional a la cantidad de opciones que tiene el atributo para separar los datos. Por lo tanto la *Tasa de Ganancia* de un atributo T para un conjunto de datos D se calcula como (1.6)

$$GainRatio(D, T) = \frac{Gain(D, T)}{Split(D, T)} \quad (1.6)$$

De tal forma que cuando se trata de seleccionar un atributo de un conjunto de atributos posibles, se evalúa la ecuación (1.6) para cada uno de ellos, seleccionando aquel que posea el mayor valor.

Algoritmo Árboles de decisión Parciales (PART)

Otra de las técnicas que permite la obtención de reglas de clasificación, a partir de árboles de clasificación construidos parcialmente, es el algoritmo **PART** definido por (Frank y Witten, 1998). Este algoritmo genera una lista de reglas de clasificación, es decir, una secuencia ordenada de reglas que a la hora de aplicarse a un ejemplo, deben ejecutarse en el orden en que fueron generadas hasta encontrar la primera que

cubra el ejemplo en cuestión. Dicha regla tendrá en su consecuente la clase a asignar. El resultado obtenido es una lista y no un conjunto porque luego de obtener cada regla se retira del conjunto de ejemplos los correctamente cubiertos por dicha regla.

Es importante tener en cuenta la diferencia entre un conjunto de reglas independientes y una lista de clasificación. Si se trata de reglas independientes, podrán superponerse en el espacio de entrada permitiendo que un mismo ejemplo pertenezca a dos clases distintas. Cuando se trata de listas de clasificación, las reglas se analizan una por una comenzando por la que se generó primero hasta encontrar aquella que cubra el ejemplo. La lista incluye una clase por defecto, la misma que es asignada al conjunto de ejemplos que no cumplen con ninguna regla. En otras palabras al utilizar una lista de clasificación, un ejemplo dado sólo puede ser asignado a una clase.

El método **PART** requiere de la definición previa de un árbol de clasificación parcialmente construido. Esto implica que en cierto punto del proceso, el árbol deja de construirse y a partir de las ramas que terminan en hoja se decide la regla que se va a generar. Cuando la construcción se detiene, se agrega a la lista de reglas la rama con mayor cobertura y se repite el proceso hasta que la cantidad de ejemplos por cubrir se encuentre por debajo de un cierto umbral (o se hayan cubierto todos).

La expansión del árbol se realiza en orden comenzando por el subconjunto de menor valor de entropía, repitiéndose recursivamente hasta que todos los subconjuntos expandidos sean hojas. Este proceso se puede observar en el Algoritmo 2. Luego que se ha finalizado la construcción del árbol parcial, se selecciona la rama con mayor cobertura, que finaliza en una hoja. Después de obtener una regla, los ejemplos correctamente cubiertos por ella son eliminados de los datos de entrada, y el árbol se descarta. Este proceso se realiza de forma iterativa hasta lograr llegar hasta la cobertura deseada.

Algoritmo 2: Construcción parcial del árbol de clasificación

Expandir_Subconjunto(S):

begin

 Elegir una métrica y usarla para dividir el conjunto de ejemplos en subconjuntos;

 Ordenar los subconjuntos en orden creciente de entropía promedio;

while (*Exista un subconjunto X que aún no se ha expandido*) y (*Todos los subconjuntos expandidos hasta ahora sean hojas*) **do**

 Expandir_Subconjunto(X);

if (*Todos los subconjuntos expandidos son hojas*) y (*el error estimado del subárbol es mayor al estimado en el nodo*) **then**

 Deshacer la expansión en subconjuntos y convertir el nodo en hoja;

1.6.2. Redes Neuronales

Los métodos de extracción de reglas de clasificación convencionales, independientemente de si se basan en árboles o no, van incorporando condiciones al antecedente de la regla, con el objetivo de cubrir los ejemplos en forma adecuada. A medida que la longitud del antecedente crece, el soporte de la regla disminuye pero su confianza aumenta.

La extracción de reglas utilizando Redes Neuronales tiene un funcionamiento diferente ya que opera sobre todos los atributos a la vez. En este sentido, las redes neuronales competitivas, muy utilizadas para resolver problemas de agrupamiento (clustering), han sido empleadas en muchas ocasiones como punto de partida para la obtención de reglas de clasificación.

En el estado del arte se pueden encontrar varias de estas arquitecturas que han sido utilizadas para la extracción de reglas (Setiono, Baesens y Mues, 2008) (Reyes, Morales-Esteban y Martínez-Álvarez, 2013).

Dentro de este tipo de red se puede considerar a la red neuronal SOM que fue definida por (Kohonen, 1990). Su aplicación principal es el agrupamiento de la información disponible y se caracteriza por su capacidad para preservar la topología de los datos de entrada. Puede ser representada como una estructura de dos capas: la capa de entrada cuya función es sólo permitir el ingreso de la información a la red y la capa competitiva que es la encargada de realizar el agrupamiento. Las neuronas que forman esta segunda capa se encuentran conectadas y poseen la capacidad de identificar la cantidad de “saltos” o conexiones que la separan de cada una de las restantes dentro de este nivel. Cada neurona competitiva lleva asociado un vector de pesos o centroide representado por los valores de los arcos que llegan a ella desde la capa de entrada. De esta forma, la red SOM maneja dos estructuras de información: una referida a los centroides asociados a las neuronas competitivas y otra encargada de determinar la proximidad entre neuronas. Esto le permite brindar información adicional con respecto a los agrupamientos ya que neuronas cercanas dentro de la arquitectura representarán agrupamientos similares en el espacio de los datos de entrada. Un detalle importante a tener en cuenta para la red SOM es que su eficiencia y eficacia depende de la cantidad de neuronas competitivas a utilizar. Pueden hallarse en la literatura soluciones a este problema a través de los mapas dinámicos denominados GSOM (Growing Self-Organizing Maps) los cuales presentan una variante del modelo SOM que considera la posibilidad de expandir o contraer la arquitectura de la red durante el proceso adaptativo (Hsu, 2000).

En el estado del arte pueden encontrarse trabajos referidos a la extracción de reglas, utilizando redes neuronales (Andrews, Diederich y Tickle, 1995; Baesens y col., 2003; Zhu y Azar, 2014). Sin embargo, si bien este tipo de trabajos extraen reglas simbólicas de las redes formadas, siempre existen inconvenientes en cuanto a la expresividad que presentan dichas reglas. Estos algoritmos obtienen directamente, a

partir de la red entrenada, reglas que relacionan las entradas y salidas de dicha red. Estas técnicas suelen utilizar la red entrenada para clasificar ejemplos y generar adicionalmente ejemplos denominados "artificiales", que después son utilizados por un algoritmo de aprendizaje, para finalmente inferir las reglas, estos trabajos discuten tres tipos de algoritmos Neurorule, Trepan, Nefclass (Baesens y col., 2003), (Zhu y Azar, 2014).

En (Malone y col., 2006) en lugar de encontrar los grupos y las reglas de extracción que los describen, se propone una técnica que extrae reglas a partir de SOM entrenados. La técnica realiza un análisis de una matriz-U, que se obtiene como resultado del SOM entrenado, el mismo que tiene límites que se relacionan como componentes individuales. De tal forma que la técnica extrae los componentes importantes que son responsables de la agrupación, así como los valores importantes de estos componentes que pueden ser utilizados para describir los clusters formados, logrando la extracción de reglas que representan estos límites. Sin embargo, la deficiencia que existe es que en los casos en los que el SOM no logra una buena agrupación de los datos, las reglas resultantes son inexactas.

En (Pateritsas y Modes, 2000) se presentan dos enfoques diferentes para la representación del conocimiento a través de reglas, el primer enfoque apunta a definir límites en el mapa entrenado, que luego se transformará en reglas, mientras que el segundo enfoque intenta agrupar las unidades del mapa en clusters, y luego se encarga de extraer las reglas que describen estos clusters. Una de las limitantes que se presenta en este trabajo radica en la escasa cantidad de datos utilizados para el entrenamiento. Otros trabajos como el desarrollado por (Blachnik y Duch, 2011) utilizan Learning Vector Quantization (LVQ) (Kohonen, 1990), que es una versión supervisada del método de Kohonen, obteniendo un conjunto de reglas basadas en similitud de prototipos, considerando que la precisión de LVQ depende en gran medida de la inicialización adecuada de los prototipos, así como del mecanismo de optimización utilizado. Utiliza dos algoritmos para inicializar los prototipos de LVQ, uno basado en la similitud entre las clases y otro criterio utilizando el vecino más cercano para la formación de la regla.

En el Anexo A se indica en detalle las características de las redes neuronales, haciendo énfasis en SOM y LVQ utilizados como punto de partida en el método propuesto en esta tesis.

1.6.3. Máquinas de vectores de soporte

Las Máquinas de Vectores de Soporte (SVM) han participado dentro de los métodos de extracción de reglas. Son sistemas de aprendizaje basados en la teoría del aprendizaje estadístico, que muestran una buena capacidad de generalización en conjuntos de datos reales. Sin embargo, una posible limitación de SVM es que generan modelos de caja negra. Existen trabajos como el de (Haydemar, Cecilio y Andreu,

2002), (Barakat y Bradley, 2010), donde una vez determinada la función de decisión por medio de un SVM, se utiliza un algoritmo de agrupamiento para determinar los vectores prototipo para cada clase. Luego, estos vectores se combinan con los vectores de soporte, que utilizan métodos geométricos para definir elipsoides en el espacio de entrada, dando como resultado un conjunto de reglas.

La información proporcionada por los vectores de soporte, se utiliza para determinar los límites de las regiones definidas en el espacio de entrada (elipsoides o hiper rectángulos). Estas regiones se obtienen de la combinación de vectores prototipo y vectores de soporte. Los vectores prototipo, se calculan a través de un algoritmo de agrupamiento. Cada región define una regla con su sintaxis correspondiente, la misma que corresponde a ecuaciones matemáticas de los elipsoides. Esto hace que las reglas obtenidas sean complejas, y no sean fáciles de entender. Otro de los inconvenientes es la determinación de los algoritmos de agrupamiento para mejorar la generación de prototipos ya que la selección de buenos prototipos, determinará el número y la calidad de las regiones. En este caso las reglas son diseñadas a partir de los modelos SVM, y no directamente de los datos como ocurre en otras técnicas. Esta es una de las razones por las cuales se debe detallar claramente de qué forma los patrones fueron aprendidos e incorporados en la estructura de la SVM.

Estos métodos se agrupan en cuatro categorías, basadas en los componentes SVM utilizados para la extracción de las reglas. La primera categoría de algoritmos trata a la SVM como una caja cerrada, la segunda extrae las reglas directamente de los vectores de soporte, la tercera utiliza una función de decisión y los vectores de soporte, mientras que la cuarta utiliza los Vectores de soporte, la función de decisión y los datos de entrenamiento. En la Figura 1.5 se puede observar la representación gráfica de los módulos para extracción de reglas utilizando SVM.

1.6.4. Técnicas de Optimización

Al momento de resolver un problema, hallar una solución óptima implica encontrar la mejor solución, o al menos una solución que sea lo suficientemente buena para el caso en cuestión.

Existen distintas técnicas que pueden utilizarse para hallar un solución óptima que se dividen en dos grandes categorías: Exactas y Aproximadas (ver figura 1.6).

En el caso de las técnicas exactas se caracterizan por encontrar la solución óptima, sin embargo una desventaja es que el tiempo necesario crece exponencialmente con el tamaño del problema. Los algoritmos aproximados, en cambio, se caracterizan por encontrar una buena solución en un tiempo razonable, es por eso que estas técnicas cobran una singular importancia.

Entre las técnicas aproximadas, las estrategias constructivas son métodos fáciles, considerados rápidos, sin embargo las soluciones que ofrecen son de baja calidad.

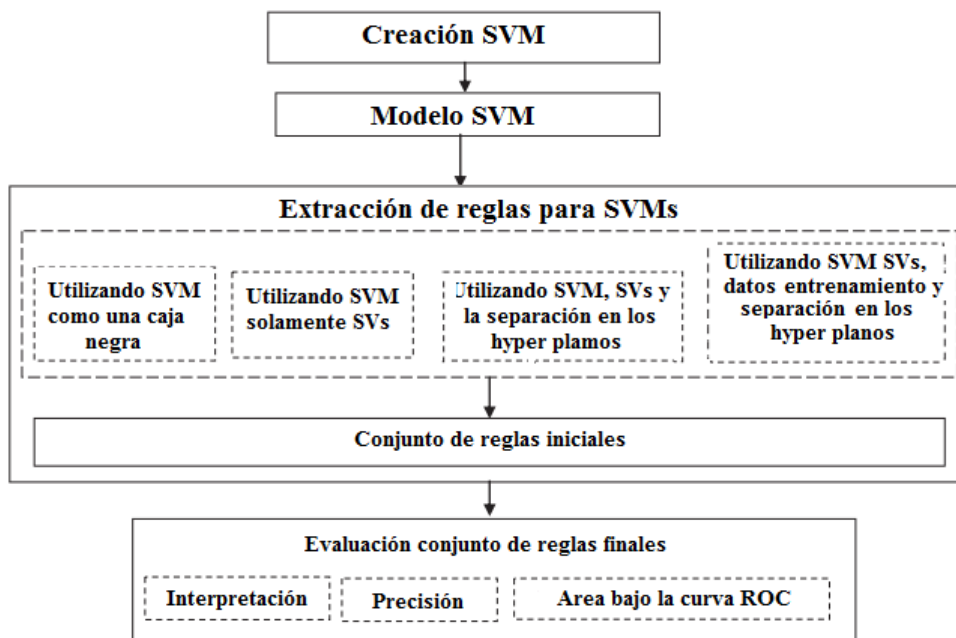


FIGURA 1.5: Módulos para extracción de reglas utilizando SVM

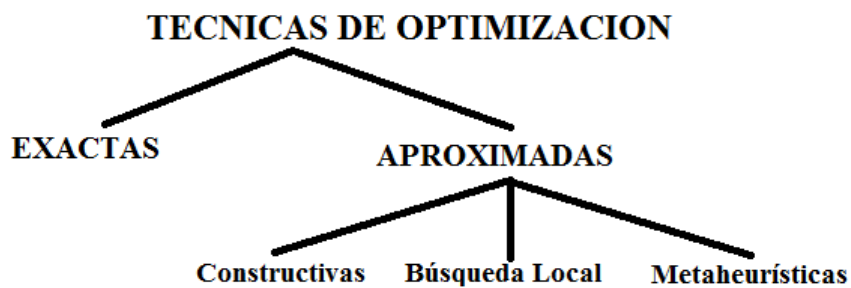


FIGURA 1.6: Clasificación de las Técnicas de Optimización

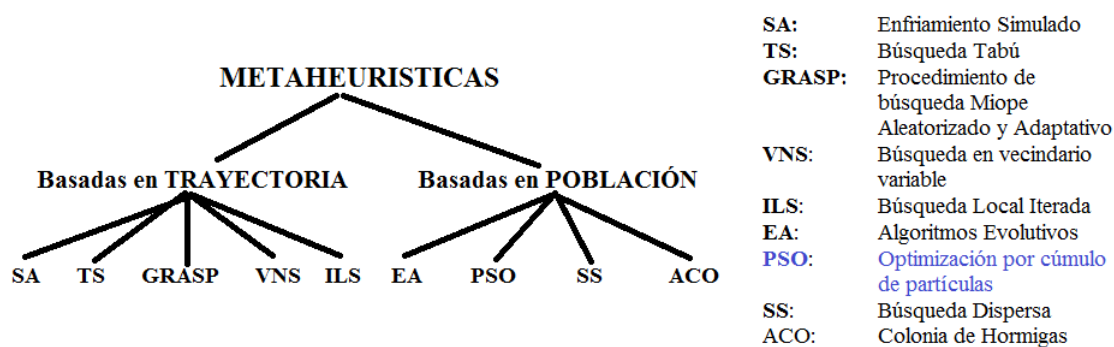


FIGURA 1.7: Clasificación de las Metaheurísticas

En cuanto a los métodos de búsqueda local, también conocidos como métodos de seguimiento del gradiente, se caracterizan porque parten de una solución completa y, utilizando el concepto del vecindario, recorren parte del espacio de búsqueda hasta encontrar el óptimo local. El vecindario de una solución s , que se determina como $N(s)$, es el conjunto de soluciones que se pueden construir a partir de s , aplicando un determinado movimiento. El óptimo local es una solución mejor o igual que cualquier otra solución de su vecindario. Los métodos de búsqueda local parten de una solución inicial, examinan su vecindario y eligen el mejor vecino, y así sucesivamente hasta que encuentran un óptimo local. Las técnicas metaheurísticas (Reeves, 1993), se caracterizan porque combinan varios métodos heurísticos, de tal forma que se pueda explorar el espacio de búsqueda de forma eficiente y efectiva.

Existen diferentes clasificaciones de las técnicas metaheurísticas (Crainic y Toulouse, 2003). Una de las clasificaciones que se utiliza más en el campo científico son aquellas que se dividen en metaheurísticas basadas en trayectoria y basadas en población. La Figura 1.7 menciona varias técnicas de cada uno de estos tipos.

Las metaheurísticas basadas en **Trayectoria** se caracterizan por partir de un punto y mediante la exploración del vecindario, van actualizando la solución actual, formando una trayectoria. Estos algoritmos parten de los métodos de búsqueda local simple, a los que se les incorpora alguna característica para no caer en los mínimos locales. La búsqueda se termina cuando se alcanza un número máximo de iteraciones que se encuentra predefinido. En el caso de las metaheurísticas basadas en **Población** son, básicamente, métodos de búsqueda que se caracterizan por mejorar iterativamente un conjunto de soluciones denominada población. El enfoque con el que realiza esta mejora varía de una técnica a otra:

- **EA: Algoritmos Evolutivos:** Se basan en postulados de la evolución biológica de supervivencia y reproducción, evolucionando hacia mejores soluciones. Dentro de este tipo de algoritmos se puede considerar:

- * Estrategias evolutivas

- * Algoritmos genéticos
 - * Programación genética
 - * Programación evolutiva
- **PSO: Optimización por cúmulo de partículas:** Mejoran la población utilizando estrategias inspiradas en el comportamiento social del vuelo de las bandadas de aves, o de los bancos de peces.
 - **SS: Búsqueda Dispersa:** Se caracteriza porque tiene un conjunto pequeño de soluciones buenas y diversas denominado conjunto de referencia. Su funcionamiento se basa en combinar las soluciones que forman el conjunto de referencia para obtener nuevas soluciones que mejoren a las que las originaron.
 - **ACO: Colonia de Hormigas:** Se basa en el comportamiento de las hormigas cuando buscan comida, dejando la feromona cuando regresan al nido con la comida, en esta técnica el rastro de la feromona se simula mediante un modelo probabilístico.

Es importante considerar que todos estos métodos deben realizar transformaciones para trabajar con diferente tipo de atributos, ya que en el caso de **Algoritmos Genéticos (AG)** y **Optimización mediante Cúmulo de Partículas (PSO)** presentan un mejor desempeño cuando se trabaja con atributos cuantitativos o numéricos mientras que **Colonia de Hormigas (ACO)** presenta mejores resultados cuando trabaja con atributos cualitativos o categóricos. Sin embargo los problemas del mundo real, presentan los dos tipos de atributos.

En el estado del arte, se pueden encontrar trabajos que utilizan programación genética para la extracción de reglas, entre los que pueden mencionarse (Langdon y col., 2008) el cual "hereda" las ventajas de los algoritmos genéticos en cuanto a la rapidez de exploración del espacio de soluciones y elimina la representación estática de los individuos, los cuales pueden tener tamaño y formas variables (como las reglas de clasificación) (Carvajal, 2015).

En (Luna, Romero y Ventura, 2012) se utiliza la **Programación Genética (PG)** para la extracción de reglas de asociación. Tiene la capacidad de trabajar con atributos categóricos y numéricos, eliminando los problemas de discretización. Durante el proceso evolutivo se seleccionan aquellos individuos que presentan un cierto nivel de soporte y confianza; sin embargo uno de los inconvenientes es que trabajan con una población fija, no permiten individuos con tamaño ni formas dinámicas, lo que limita su campo de acción. Otro trabajo es el de (Wong y Hamouda, 2000) que utiliza **AG** para la obtención de reglas difusas.

Es importante considerar que **ACO** trabaja mejor con atributos cualitativos mientras que **AG** y **PSO** lo hacen con atributos numéricos por lo que es importante considerar que se debe realizar transformaciones en los datos de entrada, de tal forma



FIGURA 1.8: Ejemplos de Inteligencia de Cúmulo en la naturaleza

que permitan trabajar con estos dos tipos de atributos. En el estado del arte también se puede encontrar una combinación de estas técnicas.

Las técnicas de optimización permiten relacionar ítems, lo que faculta la extracción de reglas.

Usando la estructura dinámica de los individuos de la **PG**, es posible representar las reglas de clasificación para la evolución de las mismas y hallar el mejor individuo entre un conjunto de reglas que estimen un valor de la clase. No obstante, los problemas de clasificación cuentan con muchos valores en la clase, lo que implica un número igual o mayor de reglas (Carvajal, 2015). Por lo tanto, si se obtiene como resultado final de la evolución a la mejor regla, esta regla es válida únicamente para un único valor de la clase, si no se conoce que sucede con los otros valores de la clase, la solución del problema es incompleta.

Se pueden encontrar en la literatura trabajos que combinan varias de estas técnicas, como es el caso de (Holden y Freitas, 2008), que combina dos técnicas de optimización como **ACO** y **PSO** para la obtención de reglas de clasificación. Este algoritmo híbrido trabaja con atributos tanto continuos como nominales, y al compararlo con **PART** obtiene una mejor precisión.

Optimización por cúmulo de partículas (Particle Swarm Optimization - PSO)

La técnica de optimización por cúmulo de partículas o **PSO**, definida en (Kennedy y Eberhart, 1997) y , se basa en el comportamiento de una bandada de aves o peces (Figura 1.8) (Kennedy, 2010). **PSO** utiliza lo que se conoce como *Inteligencia de Cúmulo* y plantea una "metáfora social" que se puede resumir de la siguiente forma: los individuos que conviven en una sociedad tienen una opinión que es parte de un conjunto de creencias"(el espacio de búsqueda) compartido por todos los posibles individuos.

A diferencia de otras estrategias poblacionales que reemplazan los individuos de la población durante el proceso adaptativo, **PSO** sólo modifica la posición de cada solución dentro del espacio de búsqueda. Este movimiento está controlado por la

función de aptitud o función de *Fitness* con la que se mide el desempeño de cada solución en dicho espacio y el conocimiento del cúmulo.

Luego, un individuo se adapta siguiendo tres factores:

- Su conocimiento actual (su capacidad para resolver el problema, valor de aptitud o fitness)
- Su conocimiento histórico o experiencias anteriores (su memoria o conocimiento cognitivo)
- El conocimiento histórico o experiencias anteriores de los individuos situados en su vecindario (su conocimiento social)

En **PSO**, los individuos de la población reciben el nombre de partículas. Cada partícula está compuesta por tres vectores y dos valores de fitness:

- Vector $X_i = (X_{i1}, X_{i2}, \dots, X_{in})$ almacena la posición actual de la partícula en el espacio de búsqueda
- Vector $pBest_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{in})$ almacena la mejor solución encontrada por la partícula hasta el momento.
- Vectores Velocidad $V_i = (V_{i1}, V_{i2}, \dots, V_{in})$ almacena la dirección sobre la cual se moverá la partícula.
- $FitnessX_i$ almacena el valor de aptitud de la solución actual. Vector X_i
- $FitnesspBest_i$ almacena el valor de aptitud de la mejor solución local encontrada hasta el momento. Vector $pBest_i$

La principal diferencia entre **PSO**, y otras técnicas evolutivas radica en que el primero mantiene información tanto de posiciones como del cambio de su posición es decir la velocidad, mientras que las otras técnicas evolutivas de manera general sólo mantienen información únicamente de las ubicaciones. En el Anexo **B** se describen en detalle el proceso de búsqueda y optimización llevado a cabo por varias variantes de **PSO** que fueron utilizadas en el desarrollo del método propuesto en esta investigación.

En trabajos como el de (Holden y Freitas, 2008), se puede observar que en métodos para obtención de reglas, que utilizan la versión reducida de **PSO** para atributos continuos proporcionan un ligero aumento en la precisión.

1.7. Conclusiones

En la primera parte de este capítulo se han dado nociones generales que hacen referencia a la extracción del conocimiento. Esto constituye un marco de referencia

para las investigaciones realizadas a lo largo de esta tesis. El foco estuvo puesto en establecer la importancia que debe darse al reconocimiento inicial de la tarea a resolver ya que condiciona el tipo de técnica de **DM** a utilizar y por consiguiente también establece cuáles serán las operaciones y transformaciones que deben efectuarse sobre los datos.

Luego, una vez sentada esta base, se ha realizado una revisión teórica sobre las diferentes alternativas que existen para la obtención de las reglas de clasificación, como son los árboles, las redes neuronales, las máquinas de vectores de soporte y las técnicas de optimización, haciendo énfasis en **PSO** por ser la técnica base de la estrategia propuesta en esta tesis.

Capítulo 2

Lógica Difusa

Este capítulo hace una revisión de la (**Lógica Difusa (LD)**) comparándola con la lógica clásica, y las aplicaciones que se tienen en el mundo real. Se describen los sistemas difusos que toman como pilar la aplicación de la **LD**, detallando cada uno de sus componentes. Finalmente se revisa en qué consisten las reglas difusas, y los sistemas que consideran dichas reglas.

2.1. Introducción

El concepto de **LD** fue creado por Lofti A. Zadeh, catedrático de la Universidad de Berkeley (California) (Zadeh, 1965). Básicamente la **LD** es una lógica multivaluada que permite representar matemáticamente la incertidumbre y la vaguedad, proporcionando herramientas formales para su tratamiento.

La **LD** fue diseñada precisamente para imitar el razonamiento del ser humano. Permite estructurar de mejor manera, todo aquello que está separado por fronteras mal definidas o imprecisas, como son el pensamiento, el lenguaje y la percepción. La **LD** proporciona un mecanismo de inferencia, que permite simular los procedimientos del razonamiento humano, en sistemas basados en el conocimiento, los mismos que siempre son ambiguos. Se encarga de representar el comportamiento de fenómenos que son difíciles de predecir con modelos matemáticos, utilizando un lenguaje claro y natural. Por lo tanto, se puede decir que la **LD** *"constituye una herramienta de representación del conocimiento que permite modelar incertidumbre e imprecisión de una forma sencilla y directamente interpretable por el usuario"* (Hernandez Orallo, Ramirez Quintana y Ramirez, 2004).

En 1922 Lukasiewicz cuestionaba la lógica clásica bivaluada de valores cierto y falso (Giles, 1976) (Boicescu y col., 1991). Lukasiewicz suponía una lógica de valores ciertos en el intervalo unidad como generalización de su lógica tri-valuada [verdadero, falso, indeterminada] (en (Kosko, Isaka y Satoru, 1993) se resume esta parte de la historia de la (**LD**)).

El primer formato de reglas fue propuesto por (Mamdani, 1974), con la finalidad de realizar un controlador difuso que estabiliza un sistema en torno a su punto de

trabajo. A partir de 1975 se empezaron a desarrollar aplicaciones con éxito, de tal forma que además de Zadeh han existido otros aportes como los de (Mamdani, Østergaard y Lembessis, 1983) aplicando la LD a un sistema de control de vapor. También ha sido utilizada por (Tsukamoto, 1993) quién utilizó la aplicación del control difuso al proceso de purificación del agua. Muchas de las definiciones de la LD se exponen en (Wang y col., 1995), (Jang, Sun y Mizutani, 1997), (Passino y Yurkovich, 1998) y (Lin y Lee, 2019).

La LD hace referencia a que *los elementos claves en el pensamiento humano no son números, sino etiquetas*. De esta forma, los objetos pueden pertenecer a varias clases, sin existir una rigidez en los límites de las clases. La LD incluye:

- Predicados borrosos (caro, bajo, alto,)
- Cuantificadores borrosos (poco, casi todo, mucho,)
- Valores ciertos borrosos (muy cierto, mas o menos cierto,)

2.2. Aplicación de la Lógica Difusa

La LD es aplicable en procesos complejos para los que no existe un modelo de solución simple, o un modelo matemático preciso. De la misma manera se puede aplicar, cuando ciertas partes de un sistema a controlar son desconocidas y no pueden medirse de forma confiable y cuando el ajuste de una variable puede producir el desajuste de otras. No es recomendable utilizar la LD cuando algún modelo matemático ya soluciona eficientemente el problema, cuando los problemas son lineales o cuando no tienen solución. Es considerada una herramienta importante para el control de sistemas (Dubois y Prade, 1997). (Vaidyanathan y Azar, 2016), y procesos industriales complejos, así como también para la electrónica de entretenimiento y hogar, sistemas de diagnóstico y sistemas expertos.

Actualmente existe una gran cantidad de aplicaciones de la LD, que afectan nuestra vida de manera cotidiana. La LD se ha desarrollado en diferentes áreas, algunas de ellas son:

- **Control de sistemas:** Control de tráfico, control de vehículos, control de compuertas en plantas hidroeléctricas así como en centrales térmicas, control en máquinas lavadoras, control de metros (mejora de su conducción, precisión en las paradas y ahorro de energía), ascensores, entre otras aplicaciones. Existen en la literatura implementaciones de distinto tipo de controladores basados en la combinación de LD y técnicas de optimización como por ejemplo los que se detallan en (Bingül y Karahan, 2011), (Melin y col., 2013), (Collotta, Pau y Marniscalco, 2017), (Borni y col., 2017).
- **Predicción** de terremotos, optimización de horarios.

- **Reconocimiento** de patrones y Visión por ordenador: Seguimiento de objetos con cámara reconocimiento de escritura manuscrita, reconocimiento de objetos, compensación de vibraciones en la cámara, sistemas de enfoque automático.
- **Sistemas de información o conocimiento:** Bases de datos, sistemas expertos o como base de los lenguajes para la web semántica (Straccia, 2014).

La lógica difusa es un pilar importante en varias investigaciones, así en el estado del arte podemos encontrar que en (Hüllermeier, 2015), se analiza la importancia de la aplicación de la lógica difusa en machine learning, considerándola un elemento clave en el desarrollo de la inteligencia artificial. La representación y el manejo de la incertidumbre es uno de los temas a los que la lógica difusa puede contribuir de mejor manera.

En general podemos decir que el campo de las aplicaciones de la LD está en pleno desarrollo, siendo esta una de las razones por las que esta investigación puede ser considerada como un avance técnico.

2.3. Conceptos básicos de Lógica Difusa

Esta sección contiene algunas definiciones básicas que resultan necesarias a la hora de trabajar con LD.

2.3.1. Conjuntos Difusos y Función de Pertenencia

Los conjuntos clásicos surgen por la necesidad del ser humano de clasificar objetos y conceptos. Estos conjuntos pueden definirse o bien enumerando directamente los elementos que lo forman o bien mediante una función de pertenencia μ que asigna valores 1 ó 0 a cada elemento de un universo en discurso indicando si pueden pertenecer o no a dicho conjunto, respectivamente. El proceso por el que un individuo del conjunto universal A , es o no miembro de un conjunto se puede definir como una función característica. De tal forma que para un conjunto A , la función asigna un valor $\mu_A(x)$, a cada $x \in X$, de tal forma que se expresa de acuerdo a (2.1), donde se asigna el valor 1 a todos los elementos incluidos en el conjunto, y el valor 0 a los no incluidos.

$$u_A(x) = \begin{cases} 0 & \text{si y solo si } x \notin A \\ 1 & \text{si y solo si } x \in A \end{cases} \quad (2.1)$$

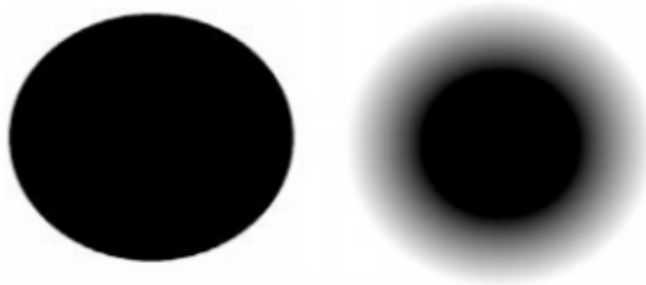


FIGURA 2.1: Representación gráfica del grado de pertenencia de los elementos a un conjunto clásico (izquierda) y un conjunto difuso (derecha)

Entonces la función característica μ_A asocia elementos del conjunto universal al conjunto $\{0,1\}$ teniendo que $\mu_A : X \rightarrow \{0,1\}$. A la función característica, que indica el grado de pertenencia de los elementos al conjunto se le denomina *función de inclusión o función de pertenencia (membership function)*.

Cuando se presentan problemas donde la imprecisión aparece, es decir existen conceptos sin límites claros, los criterios de pertenencia no se encuentran definidos nítidamente y es aquí cuando aparece la necesidad de trabajar con *Conjuntos Difusos*.

En la Figura 2.1 se puede observar la representación gráfica del grado de pertenencia de los elementos a un conjunto clásico y un conjunto difuso. El color negro corresponde al valor 1 y el blanco al 0. Nótese que en el círculo de la derecha, los elementos ubicados en la periferia poseen un grado de pertenencia menor a los centrales.

Un conjunto difuso A en U queda caracterizado por una función de pertenencia μ_A donde:

$$\mu_A : U \rightarrow [0,1]$$

Si x es un elemento del conjunto universal U , la función de pertenencia asocia a cada x un número real del intervalo $[0,1]$, donde $\mu_A(x)$ representa el "grado de pertenencia de x en A ". Cuanto más se aproxima $\mu_A(x)$ a 1, mayor será el grado de pertenencia.

Para definir las funciones de pertenencia se utilizan funciones de tipo: trapezoidal (figura 2.2), singleton (figura 2.3), triangular (figura 2.4), S (figura 2.5), pseudo-exponencial (figura 2.6) y tipo π (figura 2.7).

A continuación se revisan tres conceptos básicos relacionados con conjuntos difusos:

- El **Soporte** de un conjunto difuso A es el conjunto clásico o conjunto *crisp* que contiene todos los elementos del dominio que tienen un grado de pertenencia

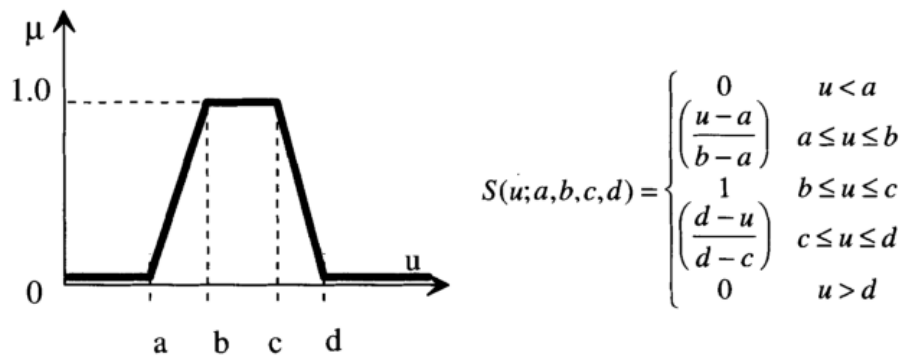


FIGURA 2.2: Función de Pertenencia Trapezoidal

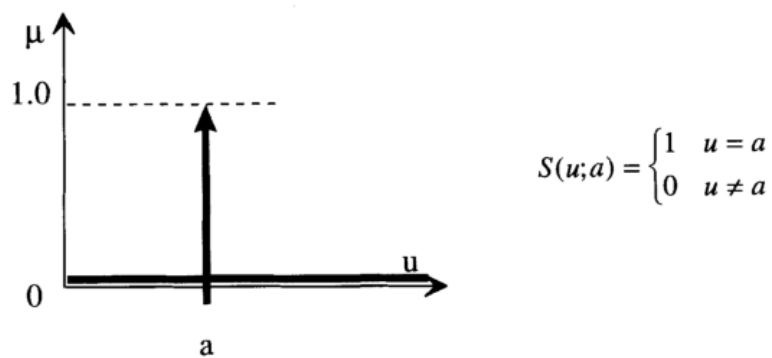


FIGURA 2.3: Función de Pertenencia Singleton

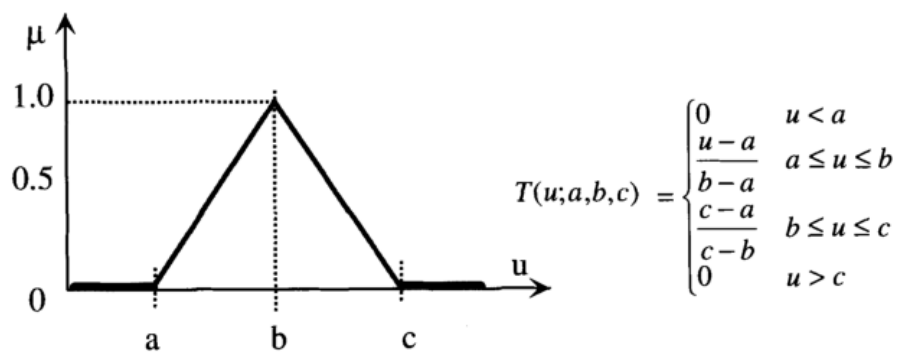


FIGURA 2.4: Función de Pertenencia Triangular

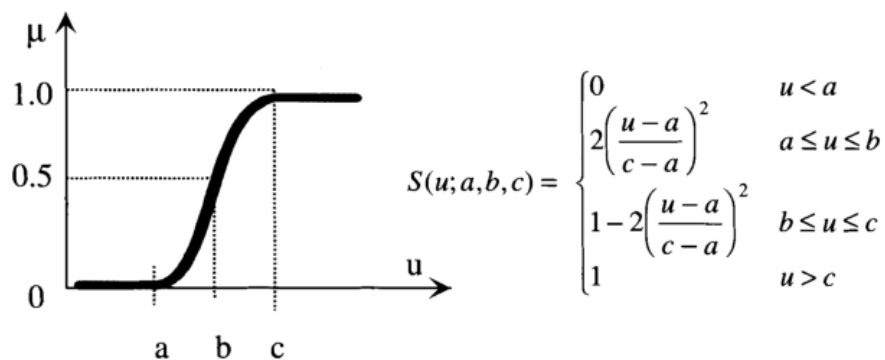


FIGURA 2.5: Función de Pertenencia Tipo S

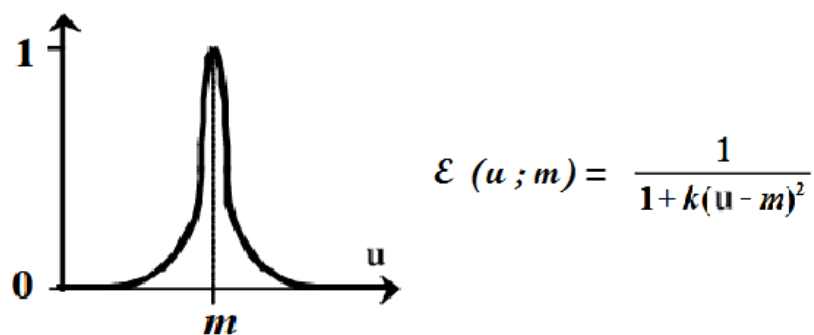


FIGURA 2.6: Función de Pertenencia Tipo Pseudo-exponencial)

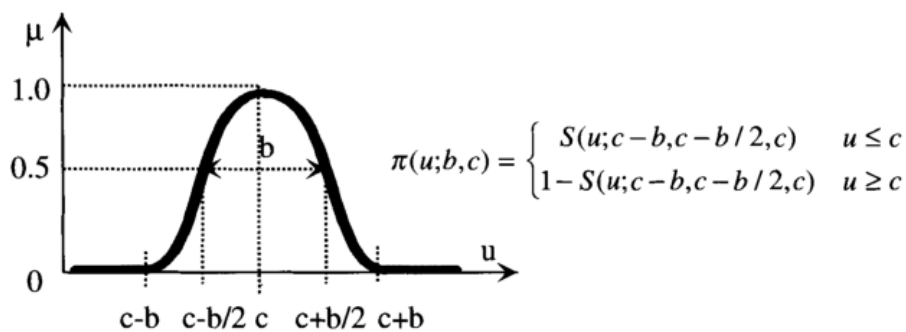


FIGURA 2.7: Función de Pertenencia Tipo π (con forma de campana)

no nulo en A .

$$Sop(A) = \{x \in U \mid \mu_A(x) > 0\} \quad (2.2)$$

Luego, el conjunto difuso A se puede escribir de la siguiente forma

$$A = \{ (x, \mu_A(x)) \mid x \in Sop(A) \} \quad (2.3)$$

- Se conoce como la **Altura** de un conjunto difuso al mayor grado de pertenencia que tiene cualquiera de los elementos que lo conforman. Se denomina **Núcleo** del conjunto difuso al conjunto de elementos del soporte que tienen el máximo grado de pertenencia a dicho conjunto difuso. Por último se debe indicar que el conjunto difuso es **Normalizado**, cuando por lo menos uno de sus elementos alcanza el grado de pertenencia 1, es decir que el elemento pertenece totalmente al conjunto difuso.
- α -**corte** de un conjunto difuso A es un conjunto crisp o clásico, que contiene todos los elementos del conjunto universal U que tienen un grado de pertenencia en A mayor o igual que el valor de α :

$$A_\alpha = \{x \in U \mid \mu_A(x) \geq \alpha\} \quad (2.4)$$

2.3.2. Variables Lingüísticas

Una variable lingüística es aquella que puede tomar por valor términos del lenguaje natural, el mismo que está definido por una palabra, etiqueta lingüística o adjetivo como por ejemplo: alto, medio, bajo, mucho, poco, etc, que son las palabras que desempeñan el papel de etiquetas en un conjunto difuso (Brío y Sanz, 2006). En una expresión como *el Ingreso es alto*, la variable Ingreso debe ser entendida como una variable lingüística, debido a que se le asigna como valor la etiqueta lingüística *alto*. Es importante considerar que, en la vida real esta variable toma valores numéricos específicos, como por ejemplo, *el Ingreso es USD 4500*.

La variable lingüística está definida por una tupla $x \rightarrow (Var, T(Var), Z)$ donde Var es el nombre de la variable, $T(Var)$ es el conjunto de términos que nombran los valores x que puede tomar Var , valores que son conjuntos difusos en Z . Si se asocia con el ejemplo anterior se tiene $Var=Ingreso$ es una variable lingüística, $T(Ingreso)$ es el conjunto de todos los términos que puede tener ingreso como es *muy alto*, *alto*, *medio*, *bajo*, *muy bajo*, o también definiciones como *mucho*, *poco*, se puede considerar que el Universo Z de esta variable en dólares puede ir desde 0 hasta 100.000.

Por ejemplo, los términos lingüísticos de la variable "Salario"

$$T(Salario) = \{Bajo, Medio, Alto\}$$

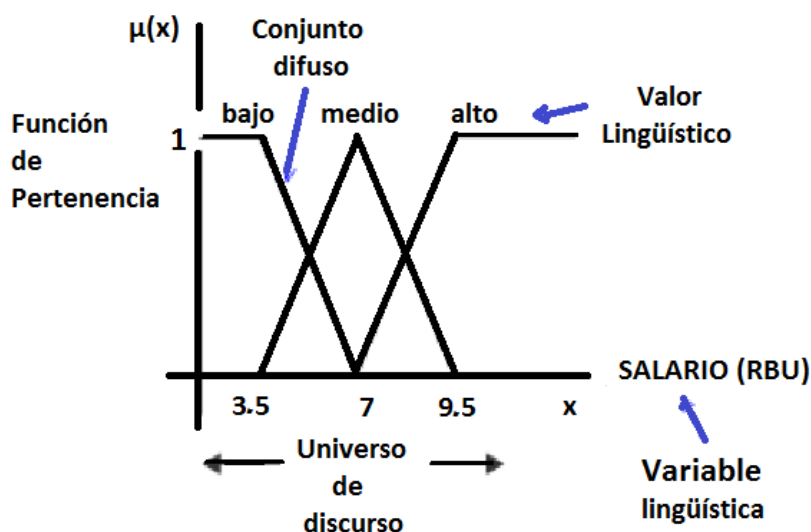


FIGURA 2.8: Ejemplo de conjuntos difusos

se pueden definir mediante tres conjuntos difusos, por lo que

$$Z_{Salario}(x) = \{Z_{Bajo}(x), Z_{Medio}(x), Z_{Alto}(x)\}$$

En la Figura 2.8 se puede observar un ejemplo de la representación de los conjuntos difusos, con la variable lingüística "Salario", el mismo que se encuentra en función del salario básico o remuneraciones básicas unificadas.

Si se considera un universo X para la variable lingüística "Salario" de personas en dólares en función de las Remuneraciones Básicas Unificadas.

$$X = \{3,5; 7; 9,5\}$$

se definen en X los conjuntos difusos *Bajo*, *Medio* y *Alto*, con su respectivas funciones de pertenencia como se muestra en la tabla 2.1.

En la Figura 2.9 se puede observar otro ejemplo de la variable Ingreso con sus respectivas funciones de pertenencia a cada uno de los conjuntos difusos en un Universo X :

$$X = \{200, 500, 1000, 2500, 3000, 4000, 5000, 6500, 7200\}$$

2.3.3. Particiones Difusas

Aunque la definición de partición difusa difiere según el autor (Bezdek, 1981; Bouchon y Cohen, 1986; Butnariu, 1983; Ruspini, 1969), en general la idea es dividir al universo de discurso en una serie de clases difusas normalmente asociadas a etiquetas lingüísticas. Existen multitud de formas de definir particiones difusas sobre

TABLA 2.1: Grados de pertenencia de cada elemento de los conjuntos difusos que determinan los valores lingüísticos de la variable difusa "Salario"

Salario en dolares			
Salario	Bajo	Medio	Alto
200	1	0	0
500	1	0	0
1000	0,6	0,6	0
2500	0,1	0,8	0,3
3000	0	1	0,6
4000	0	1	0,8
5000	0	1	1
6500	0	1	1
7200	0	1	1

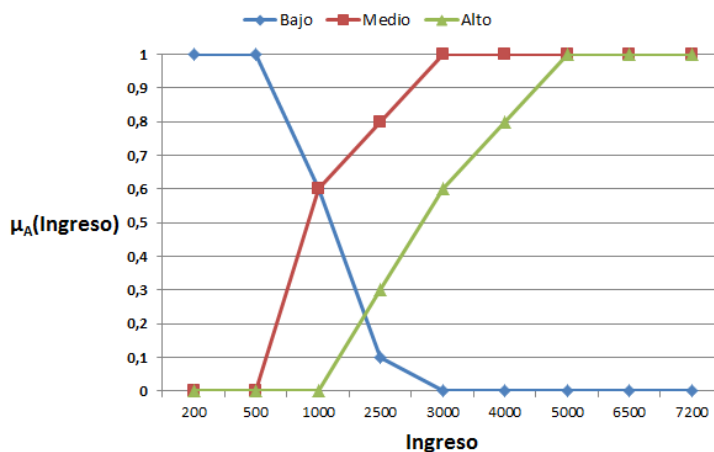


FIGURA 2.9: Funciones de pertenencia a los conjuntos difusos Bajo, Medio y Alto de la variable Ingreso

un universo U . La más conocida que no necesariamente es la más conveniente, es la familia de particiones dadas por expertos. Este tipo de particiones es útil cuando una partición viene dada por conceptos lingüísticos propios del ser humano sin sustento científico ni definición formal, como puede ser la división del día en intervalos de tiempo vagos como la mañana, tarde, noche, etc; la división de vegetales en fruta o verdura o la clasificación de animales en mascotas. (Madrid y Díaz, 2019)

Así para la variable Ingreso se pueden tener tres conjuntos difusos cada uno identificado por: $\{Bajo, Medio, Alto\}$ con la respectiva función de pertenencia:

$$\{\mu_{Bajo}(t), \mu_{Medio}(t), \mu_{Alto}(t)\}$$

Se dice que una partición es *completa* cuando los conjuntos difusos definidos cubren todo U . Cuando la intersección entre dos conjuntos difusos no es nula se dice que los conjuntos están *solapados*, de este modo, el solapamiento de un conjunto difuso es la relación del número de elementos que comparte con otros conjuntos de la misma partición, respecto del número total de elementos que la conforman.

Para la definición de las particiones es necesario considerar la complejidad de las variables. Se recomienda que las particiones sean completas, con un solapamiento del 20% al 50%, y en número impar. Normalmente se emplean particiones de 3 o 7 conjuntos, ya que en este caso la complejidad no es excesiva y permiten una precisión suficiente en la descripción de los valores de la variable. También para simplificar el análisis se recomienda definir conjuntos de tipo **T** Triangular (Starczewski, 2009), (Barua, Snigdha y Kosheleva, 2014).

Por otro lado, la imprecisión de un conjunto difuso se puede medir con la **Entropía difusa** (Luca y Termini, 1972). Sea A un conjunto difuso en el Universo U , con una función de pertenencia $\mu_A(x)$, para todo x en U . Si U es un conjunto discreto $U = \{x_1, x_2, \dots, x_n\}$, la imprecisión del conjunto difuso A se mide por:

$$f(A) = -\frac{1}{n} \sum_{i=1}^m \{\mu_A(x) \ln \mu_A(x) + [1 - \mu_A(x)] \ln[1 - \mu_A(x)]\} \quad (2.5)$$

2.3.4. Operaciones Difusas

En la **LD** se consideran tres operaciones:

- **Complemento:**

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad \forall x \in U \quad (2.6)$$

- **Unión:** de dos conjuntos difusos A y B es otro conjunto difuso $A \cup B$ que cumple:

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad \forall x \in U \quad (2.7)$$

- **Intersección:** de dos conjuntos difusos A y B es otro conjunto difuso $A \cap B$ que cumple:

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad \forall x \in U \quad (2.8)$$

Las *normas y conormas triangulares* establecen modelos genéricos para las operaciones de unión e intersección, las cuales deben cumplir ciertas propiedades básicas (conmutativa, asociativa, monotonicidad y condiciones frontera)

La función mínimo (\wedge) es una *t-norma*, que corresponde a la operación de intersección en conjuntos clásicos cuyos grados de pertenencia están en $\{0,1\}$. Por esto, esta función es la extensión natural de la intersección en conjuntos difusos.

La función máximo (\vee) es una *t-conorma*, que corresponde a la operación de unión en conjuntos clásicos cuyos grados de pertenencia están en $\{0,1\}$. Por esto, esta función es la extensión natural de la unión en conjuntos difusos.

2.3.5. Representación de conjuntos difusos

Para definir un conjunto difuso es necesario definir su función de pertenencia. Entre las alternativas para la representación de conjuntos difusos se puede mencionar:

- Preguntar a un experto del área del conocimiento.
- Particionando el rango de cada atributo numérico.
- Utilizar una técnica de minería de datos (Yager y Zadeh, 2012), (Gottwald, 2013).

En el marco de esta investigación se consideraron estas tres alternativas, en el primer caso se definieron los conjuntos difusos con el experto en el área del conocimiento, utilizando para la representación del conocimiento funciones triangulares y trapezoidales. En el segundo caso, se particiono el rango de cada atributo numérico en intervalos de igual longitud, centrando, en cada uno de ellos, una función triangular con un solapamiento adecuado. Por último se utilizó FCM (Hanyu y col., 2019), (Shi y col., 2019) como técnica para la distribución de los conjuntos difusos.

2.3.6. Reglas Difusas

Como se dijo anteriormente la LD trata información como es el caso de *ingresos altos, riesgo bajo, velocidad media*, los mismos que pueden ser utilizados en la construcción de reglas, haciendo más fácil su interpretación.

Las reglas difusas combinan uno o más conjuntos difusos de entrada, llamados antecedentes o premisas; asociado a un conjunto rígido (crisp) o difuso de salida

denominado consecuente. Los conjuntos difusos de la premisa se asocian mediante conjunciones lógicas del tipo "and". Las reglas difusas son del tipo **IF-THEN SENTENCES**. Un ejemplo de una regla de clasificación difusa es "Si el ingreso es alto y el egreso es bajo, entonces conceder el crédito".

Para definir las reglas caben dos vías:

- Diseñarlas a base de la opinión de los expertos
- Utilizar un sistema de aprendizaje como las redes neuronales, técnicas de optimización por cúmulo de partículas, algoritmos genéticos, entre otros para extraer las reglas a base de los datos disponibles.

Según (Sik y Lee, 2013) para resolver la problemática que presenta la extracción del conocimiento, se debería definir en conjunto con el experto las etiquetas lingüísticas, obteniendo como resultado una reducción de la granularidad de los dominios de los atributos. La utilización de las reglas difusas permite obtener reglas de mayor calidad, debido a que la utilización de las etiquetas lingüísticas ayudan a reducir la segmentación del dominio, otorgando reglas más intuitivas por su proximidad a la forma de razonar del experto humano.

Se necesita de un conjunto de reglas para representar las distintas relaciones que existen en los datos. Esto último da lugar a lo que se denomina **base de reglas difusas**.

Una base de reglas difusas es una colección de reglas, $R^{(l)}$, que tienen el siguiente formato:

$$R^{(l)} : IF x_1 is F_1^{(l)} and \dots and x_n is F_n^{(l)} THEN y is G^{(l)} \quad (2.9)$$

donde $F_i^{(l)}$ y $G^{(l)}$ son conjuntos difusos contenidos en los universos $U_i \subset \mathbb{R}$ y $V \subset \mathbb{R}$ y $x = (x_1, \dots, x_n)^T \in U_1 \times \dots \times U_n$ son variables lingüísticas. A este formato de reglas se le conoce como de **tipo Mamdani**.

Otro formato para las reglas son las de **tipo Sugeno** donde la función de salida es una combinación lineal de las variables de entrada.

$$R^{(l)} : IF x_1 is F_1^{(l)} and \dots and x_n is F_n^{(l)} THEN y^l = f^l(x) \quad (2.10)$$

donde el vector x representa el conjunto de las entradas mientras que y es la salida del sistema difuso.

Los sistemas difusos que tienen múltiples entradas y una sola salida se llaman **MISO (Multiple Input Single Output)**, mientras que aquellos que tienen múltiples entradas y múltiples salidas toman el nombre de **MIMO (Multiple Input Multiple Output)**

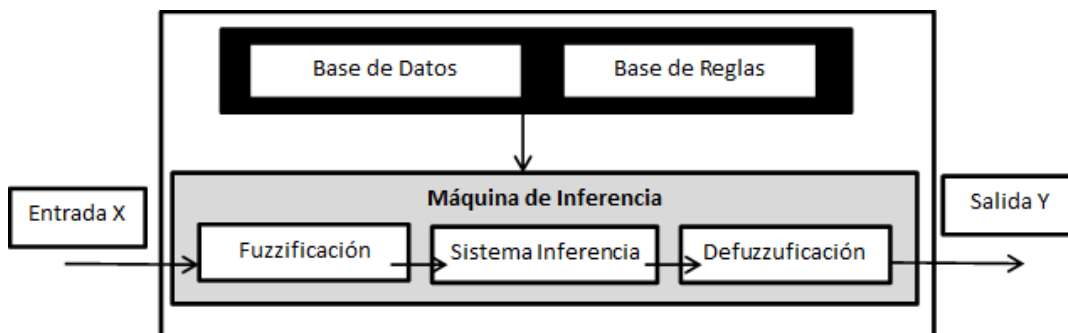


FIGURA 2.10: Estructura genérica de un Sistema Basado en Reglas Difusas

Debido a que las variables que provienen del exterior serán, en general, valores no difusos, es necesario fuzzificarlas previamente, es decir establecer una relación entre los valores de la variable de entrada no difusa al sistema $x = (x_1, x_2, \dots, x_n)^T$, y sus correspondientes conjuntos difusos A en su respectivo universo U . Se pueden utilizar las siguientes estrategias de fuzzificación:

- **Fuzzificador Singleton:** en este método los valores discretos de las variables son considerados como conjuntos difusos. Consiste en que para cada valor de la variable de entrada x se define un conjunto A' que lo soporta, con una función de pertenencia $\mu_{A'}(x')$, de modo que $\mu_{A'}(x) = 1$, ($x' = x$) y $\mu_{A'}(x') = 0$, para todos los otros $x' \in U$ en los que $x' \neq x$
- **Fuzzificador No Singleton:** utiliza una función exponencial del tipo:

$$\mu_{A'}(x') = a.exp[-((x' - x)/\sigma)^2] \quad (2.11)$$

Se trata de una función con forma de campana, con el valor x' de entrada y a y σ , son los parámetros que le dan la forma concreta de la gaussiana (altura y amplitud respectivamente)

Sistemas basados en Reglas difusas

Un **Sistema Basado en Reglas Difusas (FRBS)** presenta dos componentes principales:

- La **Base de Conocimientos (KB)** representando el conocimiento sobre el problema que se resuelve en forma de reglas lingüísticas difusas *SI-ENTONCES*.
- El motor de inferencia que pone en efecto el proceso de inferencia difuso necesario para obtener una salida desde el **FRBS** cuando se especifica una entrada.

La estructura de un **FRBS** lingüístico se muestra en la Figura 2.10.

La **KB** está compuesta por la Base de Reglas, formada por la colección de reglas lingüísticas, y por la Base de Datos, que contiene los conjuntos de términos y las funciones de pertenencia.

El motor de inferencia está compuesto por tres componentes:

- Fuzzificación: Tiene el efecto de transformar datos de entrada nítidos en conjuntos difusos.
- Un Sistema de Inferencia: que los utiliza junto con la **KB** para realizar el proceso de inferencia difusa.
- Defuzzificación: Obtiene la salida nítida final de las salidas difusas individuales. En el caso particular del método propuesto en esta tesis, esta etapa no fue considerada dado que la identificación de la clase que realiza cada regla no es difusa.

Para diseñar un **FRBS** se deben realizar varias tareas. Una de las más importantes y difíciles es obtener una base de conocimientos adecuada que permita la resolución del problema. En esta etapa, la mayor dificultad radica en lograr que los expertos indiquen su conocimiento a través de reglas. Esto último ha hecho que se desarrollen una serie de técnicas que permitan la obtención de las reglas difusas de forma automática.

Existe una gran cantidad de entornos de programación, que permiten desarrollar Sistemas Difusos, entre estos se tiene **Fuzzy Tech** utilizado especialmente como herramienta para desarrollo de aplicaciones de control basadas en **LD**. Otra de las herramientas es **Fuzzy Inference Development Environment - FIDE** basado en un lenguaje de descripción de controladores FIL (Fuzzy Inference Language). Otros entornos de programación son TILShell (Togai Infralagic), o Cubicalc (HyperLogic) que también utilizan hardware para realizar interacciones directas entre el entorno del programa con el sistema basado en reglas. Estas herramientas utilizan C como lenguaje base y tienen interfaces amigables al usuario permitiendo el desarrollo de los sistemas de control difusos.

Los sistemas difusos utilizados comúnmente son:

- Sistemas difusos puros (Figura 2.11)
- Sistemas difusos Takagi-Sukeno-Kang (TSK) (Figura 2.12)
- Sistemas difusos con Fuzzificador y Defuzzificador (Figura 2.13)

Para la construcción de un sistema difuso se deben considerar los siguientes pasos:

- Determinar aquellas variables que van a ser objeto de análisis, considerando tanto las dependientes como las independientes.

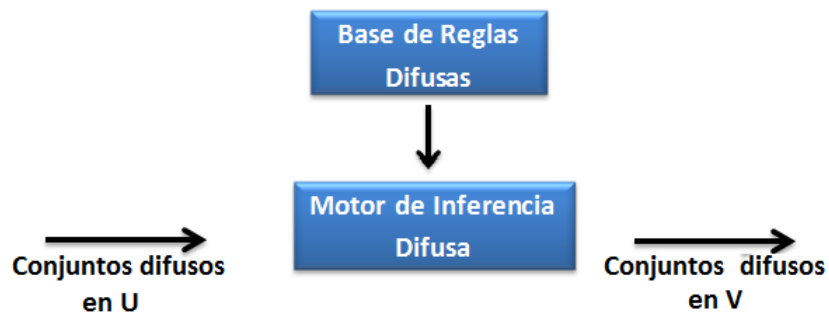


FIGURA 2.11: Sistemas Difusos puros



FIGURA 2.12: Sistema Difuso: Takagi-Sugeno-Kang (TSK)

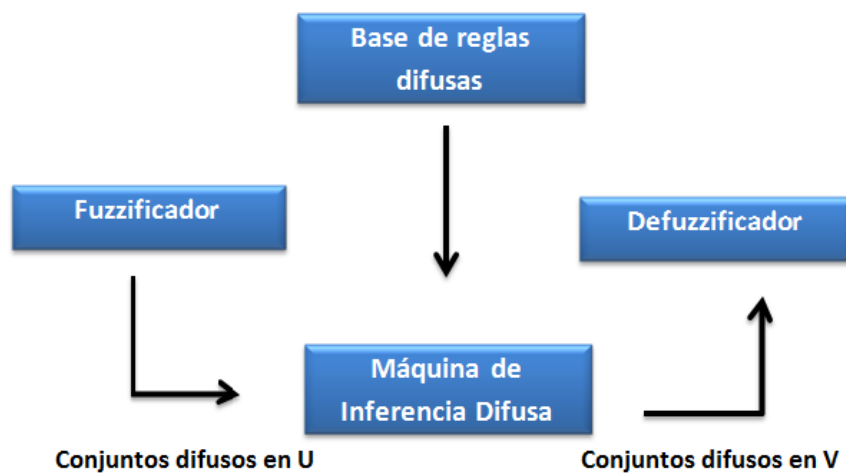


FIGURA 2.13: Sistemas difusos con Fuzzificador y Defuzzificador

- Crear los conjuntos difusos para las variables objeto de estudio.
- Integrar las reglas difusas al sistema. Si es necesario, se puede modificar levemente la función de pertenencia, de acuerdo con la descripción de las reglas difusas.
- Obtener el valor de la variable de salida a partir de las variables dependientes y las reglas difusas.
- Los resultados obtenidos permiten tomar decisiones.

En el estado del arte se pueden encontrar trabajos como el de (Chen, 2006) que propone un método basado en PSO, para construir un sistema de reglas difuso. En este método, cada individuo de la población representa a un sistema difuso completo. Este es un enfoque alternativo, al método propuesto en esta investigación que obtiene las reglas de manera individual a medida que va cubriendo los ejemplos. Esto evita tener que definir de antemano, la cantidad máxima de reglas a generar. Como detalle adicional, cabe mencionar que estos autores sostienen que la parametrización utilizada consume bastante tiempo para el entrenamiento.

En el caso de (Jang, 1993), se utilizó una partición adaptativa en el espacio de entrada, para diseñar el sistema difuso basado en ANFIS (adaptive-network-based fuzzy inference system). En este caso, el esquema de partición adaptativa de la red tiene el inconveniente, de que el número de los conjuntos difusos para cada variable de entrada están predeterminados.

En (Wong y Chia-Chong, 2000), aplicaron un AG codificado en binario para determinar un número apropiado de conjuntos difusos para cada variable de entrada, y las funciones de pertenencia. Sin embargo, para obtener posiciones centrales precisas de las funciones de pertenencia, se necesita una codificación larga; lo cual redundaría en un incremento del tiempo de entrenamiento. Además, la partición difusa utilizada en el espacio de entrada tiene inconvenientes, ya que a medida que aumenta la cantidad de dimensiones, el número de reglas difusas se incrementa considerablemente, generando reglas ficticias debido a la falta de datos de entrenamiento, en las regiones difusas. Cada región difusa mapea la premisa de una regla difusa, que está asociada a varias funciones de pertenencia.

Varias técnicas de agrupamiento se han utilizado, para determinar las premisas de un sistema difuso, como el algoritmo fuzzy c-mean (FCM) (Sugeno y Yasukawa, 1993) y el método basado en ART (Simpson, 1992). Este último, propone un método para generar las regiones que determinan la premisa de un sistema difuso. La idea básica, es agrupar los datos de entrada en clusters siendo el número de reglas difusas igual al número de clusters.

El algoritmo FCM (Sugeno y Yasukawa, 1993), se utiliza para determinar las premisas del sistema difuso. Sin embargo, la desventaja del algoritmo FCM es que la

cantidad de clusters deben estar predeterminados. Además, los resultados del agrupamiento del algoritmo **FCM**, también están influenciados por la elección de los centros iniciales y la medida de distancia. En esta investigación el algoritmo **FCM** fue utilizado, para la determinación de las particiones difusas, cuando no se cuenta con el criterio del experto, o no se realiza una distribución equitativa.

En general podemos decir que la **LD** ha permitido que los investigadores obtengan formas de mejorar el desempeño de las metaheurísticas, básicamente para acelerar la convergencia y/u obtener mejor calidad de las soluciones (Arji y col., 2019).

2.4. Representación difusa del conocimiento

El lenguaje natural identifica objetos o situaciones en términos imprecisos: joven, bajo, medio, alto, peligroso, entre otros. El razonamiento basado en estos términos no puede ser exacto, ya que representan impresiones subjetivas que pueden ser consideradas probables pero no exactas.

La teoría de conjuntos difusos representa mejor que la lógica clásica el conocimiento humano, al permitir que los fenómenos y observaciones tengan más de dos estados lógicos. En los sistemas basados en conocimiento, la función de pertenencia debe ser obtenida del experto en ese dominio del conocimiento. Esta función de pertenencia no ha de ser confundida con una función de distribución de probabilidad, ya que no está basada en la repetibilidad de las observaciones, sino en la opinión de un experto.

Para la construcción de los conjuntos difusos, es necesario considerar la participación de los expertos mediante:

- Preguntas directas para obtener conocimiento (métodos directos)
- Preguntas más sencillas que se procesan después (métodos indirectos)

Pueden participar uno o varios expertos en esta área del conocimiento, considerando los siguientes métodos:

- Directos con un experto
- Directos con múltiples expertos
- Indirectos con un experto
- Indirectos con múltiples expertos

En el desarrollo de esta tesis se ha realizado la representación típica del conocimiento en términos difusos a través de las reglas de clasificación. Cada una de las

variables que intervienen como hipótesis en una regla de clasificación debe estar asociada a un **dominio**. Cada uno de los dominios puede encontrarse dividido en tantos conjuntos difusos como los considere el experto, Cada una de estas particiones debe tener asociada una etiqueta lingüística.

La representación del dominio de conocimiento de cada variable se realiza por medio de etiquetas lingüísticas que permiten flexibilizar la representación del conocimiento por medio de reglas de clasificación. Cada predicado simple o hipótesis de la regla se representa, por tanto, por medio de un conjunto difuso. El valor clásico de una variable en la hipótesis formará parte con un cierto grado de pertenencia al conjunto difuso que especifica la premisa de la regla de clasificación.

El grado de verdad de una proposición dependerá del valor de la variable y de la definición del conjunto difuso. Por ejemplo el grado de verdad de que 1500 es un ingreso medio es 0,5, y en ingreso alto a 0,2.

2.5. Métodos para extracción de reglas difusas

Como se dijo anteriormente, las reglas difusas se pueden derivar directamente del conocimiento de un experto. Se han hecho diferentes esfuerzos en el mundo científico para obtener una mejora en el rendimiento de los sistemas basados en reglas difusas. Este problema es conocido como **Aprendizaje de Reglas Difusas (FRL)** y aún es considerado de difícil solución. (Cózar, Ossa y Gámez, 2018)

Las reglas difusas empiezan a utilizarse en (Fernández, Jesus y Herrera, 2009), con el objetivo de controlar el desbalance de clases mejorando el rendimiento del sistema de clasificación basado en reglas difusas. Para esto, se genera una base de reglas lingüísticas inicial sobre la que se aplica un algoritmo genético para seleccionar las mejores reglas. Uno de los problemas que aquí se presentan, es que el método utilizado para la creación de las reglas iniciales genera un conjunto de reglas extenso, lo que dificulta su comprensión.

Existe un gran número de métodos para generar reglas difusas automáticamente a partir de datos numéricos y/o categóricos.

En la literatura se pueden encontrar varios trabajos que utilizan redes neuronales, para la obtención del conocimiento a través de reglas; como el de (Drobics y col., 2001) que propone un enfoque de tres etapas para extraer reglas difusas. En una primera etapa calcula una pequeña representación de los datos de entrada utilizando un mapa auto organizado **SOM**. Posteriormente utiliza el algoritmo **FCM** para identificar grupos de similitud significativos dentro de los nodos. Finalmente, los métodos de aprendizaje inductivo se aplican para generar conjuntos de descripciones difusas de estos grupos, permitiendo la interpretación de los resultados.

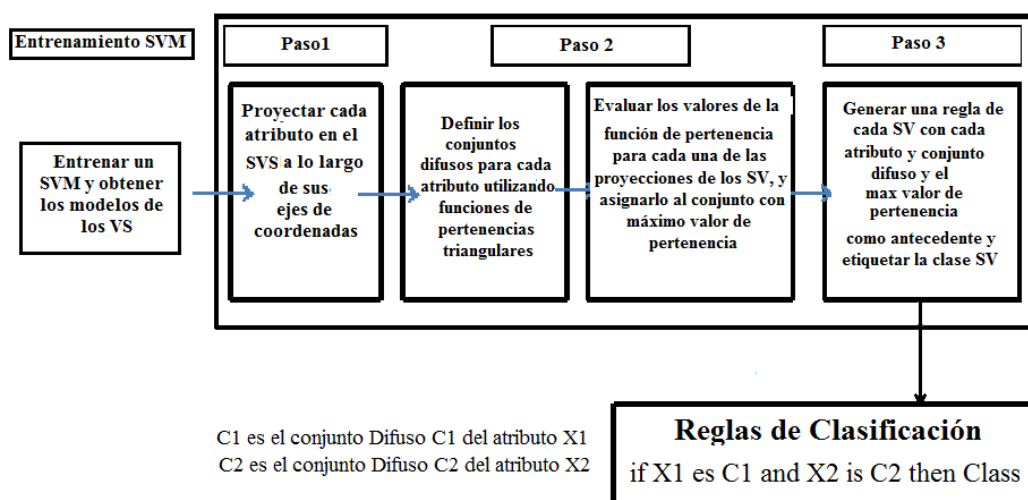


FIGURA 2.14: Representación de un algoritmo para extracción de reglas difusas utilizando SVM

Otros trabajos utilizando **SOM** se han desarrollado para realizar aplicaciones específicas, como es el caso de predicción de tiempos en sistemas para fabricación de semiconductores en el cual logran una buena precisión (Chang y Liao, 2006).

SVM es otra de las técnicas utilizadas para la extracción de reglas difusas, como lo indica (Chaves, Vellasco y Tanscheit, 2005). En dicho trabajo se utilizan sistemas basados en la teoría del aprendizaje estadístico, considerados modelos de caja negra que se encargan de combinar linealmente funciones del kernel, lo que hace que su interpretación sea difícil, representando una desventaja en este tipo de técnicas.

Se han propuesto en la literatura métodos para la extracción de reglas de **SVM** entrenados, sin embargo, las reglas generadas por estos métodos tienen, en sus antecedentes, intervalos o funciones matemáticas. Este formato disminuye la capacidad de interpretación de las reglas generadas y pone en peligro la capacidad de extracción de conocimiento.

Para aumentar la interpretabilidad lingüística de las reglas generadas, se ha propuesto en (Chaves, Vellasco y Tanscheit, 2005), (Xu y col., 2016), (Barkana, Saricicek y Yildirim, 2017), una metodología para extraer reglas difusas de un **SVM** entrenado, donde los antecedentes de la regla están asociados con conjuntos difusos. La idea principal aquí es proyectar cada característica en cada uno de los vectores de soporte (SV) a lo largo de sus ejes de coordenadas. Luego, se deben definir los conjuntos difusos para cada atributo utilizando funciones triangulares y evaluar la función de pertenencia para cada una de las proyecciones de los SV asignándolo al conjunto con el valor máximo. Finalmente se genera la regla para cada SV. En la Figura 2.14 se puede observar la representación gráfica de la obtención de una regla difusa utilizando **SVM**.

En trabajos como (Castro y col., 2007), que también utilizan **SVM** se extraen reglas difusas que presentan valores de precisión de alrededor del 53 % y una confianza del 74.59 %, sin embargo, para estos autores el objetivo es tener un conjunto de reglas que sean fáciles de interpretar, lo que se consigue con la incorporación de la **LD**. Sin embargo, uno de los inconvenientes que se presenta en este método es que se obtiene una gran cantidad de reglas. Una característica de estas reglas es que el antecedente se encuentra formado por todos los atributos.

En la literatura también se encuentran trabajos como el de (Casillas, Cerdón y Herrera, 2000), (Omkar y Karanth, 2008), (Asadi y Shahrabi, 2016), que extraen reglas difusas a partir de técnicas de optimización como **ACO**. Para esto se obtiene un algoritmo que trata el problema del aprendizaje de reglas difusas, como un problema de optimización. Uno de los inconvenientes informados fue la convergencia prematura provocada por el uso de información heurística para orientar la búsqueda global. Existen aplicaciones muy interesantes utilizando este tipo de técnicas tales como (Einipour, 2011) donde se ofrece una solución para el diagnóstico de cáncer de mama. También existen soluciones que combinan **LD** con **ACO** logrando alcanzar una tasa de precisión alta, con un conjunto de reglas fáciles de interpretar, sin embargo, el número de reglas obtenidas sigue siendo alta.

En esta tesis se propone una nueva forma de enfrentar el problema de **FRL** haciendo uso de **PSO**. Para ello, el problema de **FRL** se formula como un problema de optimización, utilizando las características relacionadas con este tipo de algoritmos, como son la creación de la población, la evaluación del individuo, función fitness, y la construcción de la solución.

2.6. Sistemas que combinan Redes Neuronales, Técnicas de optimización y Lógica Difusa

Existen en la literatura varios trabajos que construyen modelos híbridos combinando redes neuronales, técnicas de optimización y lógica difusa. Tal es el caso de los sistemas neuro-difusos que resultan de la combinación de la **LD** y las redes neuronales y tienen por objetivo reunir la capacidad de aprendizaje de las redes neuronales con la habilidad para manejar incertidumbre. Por ejemplo en (Lin y Lee, 2019) y (Jang, Sun y Mizutani, 1997) se utilizó este tipo de modelos en sistemas que expresan el conocimiento a través de reglas difusas.

Los sistemas neuro-difusos cuentan con la capacidad de expresión lingüística y aprendizaje, lo que permite una mejor interpretación, optimización e incorporación de información obteniendo un sistema difuso similar al modelo tradicional. La ventaja que presentan estos sistemas es que en cada capa que posea el modelo podrá evidenciarse en una capa de neuronas, la misma que cuenta con la capacidad de aprendizaje de las redes neuronales. Contando con datos de entrada, seguidos de

la capa de fuzzificación, posteriormente se pasa por una capa de reglas de tipo *SI-ENTONCES*, para después defuzzificar, es decir, evaluar cada una de las reglas para, de este modo, obtener finalmente los datos de salida.

Actualmente los sistemas neuro-difusos han atraído el creciente interés de los investigadores en diversos campos científicos, y de ingeniería (Shihabudheen y Pillai, 2018), debido a sus capacidades efectivas de aprendizaje y razonamiento.

El utilizar la **LD** puede mejorar directamente el razonamiento y la inferencia en aprendizaje de máquinas. El conocimiento cualitativo, aunque impreciso, puede ser modelado para permitir la expresión simbólica del aprendizaje automático, usando **LD**.

Es importante indicar que la representación del conocimiento, y la capacidad de aprendizaje automatizado de un sistema neuro-difuso, lo convierten en un poderoso marco para resolver problemas de machine learning. Entre los sistemas neuro-difusos basados en algoritmos de aprendizaje, se tiene aquellos cuyo entrenamiento se realiza por gradiente, híbrido, técnicas poblaciones o técnicas basadas en **SVM**. Al igual que propagación hacia atrás (back-propagation), y otros algoritmos de aprendizaje supervisado que utilizan la técnica basada en gradiente, sin embargo, hay que tener en cuenta que los sistemas neuro-difusos también pueden caer en el problema de los mínimos locales y la convergencia prematura.

Las técnicas neuro-difusas basadas en **PSO** se utilizan debido a su simplicidad en la implementación, a su menor empleo de memoria, y su capacidad de convergencia a la solución óptima en menor tiempo. Por ejemplo, en (Bagheri, Peyhani y Akbari, 2014) se ha utilizado una técnica neuro-difusa que utiliza **PSO** para realizar predicciones financieras a partir de reglas difusas. En (Shihabudheen y Pillai, 2018) se detalla un análisis de varias técnicas neuro-difusas, concluyendo que aquellos métodos que utilizan **PSO** dan una mejor precisión que las técnicas basadas en gradientes o **SVM**.

En general podemos decir que el desempeño de los sistemas difusos puede mejorarse con la utilización de las redes neuronales, y las técnicas de optimización garantizando la obtención de soluciones viables.

2.7. Conclusiones

Debido a que el método propuesto en esta investigación hace referencia a la modelización de la información disponible utilizando un conjunto de reglas de clasificación difusas, en este capítulo se han definido los conceptos generales asociados a la lógica difusa.

Se comenzó presentando en qué consisten los conjuntos difusos y cuál es su relación con las variables lingüísticas. La necesidad de trabajar con conjuntos difusos

parte del hecho de que existen conceptos que no tienen límites claros. Posteriormente se explicó en qué consiste la función de pertenencia, así como sus diferentes tipos. Esta definición es esencial ya que, para cada una de las variables numéricas que se van a utilizar en el modelo propuesto en esta tesis, deben definirse las funciones de pertenencia que las convertirá en variables lingüísticas. En particular, se han utilizado funciones de pertenencia triangular y se ha comparado su desempeño con funciones de pertenencia definidas por un experto en el tema.

Se explicó en detalle en qué consisten las reglas difusas y la importancia que tienen como herramienta para facilitar la interpretación del conocimiento. Luego se hizo una revisión de los sistemas que combinan redes neuronales, técnicas de optimización y lógica difusa, debido a que esta es la base central del modelo propuesto en esta tesis.

Capítulo 3

Extracción de reglas utilizando PSO

Este capítulo describe las versiones anteriores del método propuesto en esta tesis. En todos los casos se trata de técnicas que utilizan metaheurísticas poblacionales, capaces de generar reglas de clasificación, utilizando **PSO** como herramienta de base. Para cada una de ellas se describirá la representación utilizada, junto con el mecanismo de búsqueda empleado para mejorar el desempeño de las soluciones que conforman la población. En el capítulo 4 serán utilizados los métodos descritos en este capítulo, como algoritmos de prueba para medir la eficiencia y eficacia del método propuesto.

3.1. Introducción

Cuando se desea resolver un problema existen dos tipos de soluciones: exactas y aproximadas. Una solución exacta garantiza el mejor resultado pero por lo general tiene un costo computacional alto por lo que, en muchas situaciones, se opta por una solución aproximada lo suficientemente buena. Como ya se dijo en el capítulo 1, las técnicas de optimización ofrecen este tipo de soluciones siendo las metaheurísticas poblacionales una buena opción a la hora de encontrar soluciones cercanas al óptimo. En esta categoría, la Optimización por cúmulo de partículas o **PSO** es una técnica de optimización poblacional, donde los individuos buscan mejorar su capacidad de resolver el problema a medida que el proceso evolutivo avanza. Durante el proceso evolutivo las partículas compiten entre sí buscando la mejor solución.

En investigaciones anteriores se presentó un método alternativo que utilizando **PSO**, permite obtener un conjunto de reglas de clasificación (Lanzarini y col., 2017; Jimbo y col., 2016). El objetivo de este método se centró en alcanzar una buena cobertura utilizando un conjunto reducido de reglas fáciles de interpretar. Esto último estaba relacionado con la longitud del antecedente de cada regla considerando que a menor número de conjunciones empleadas más fácil resulta interpretar la decisión tomada.

Para aplicar una técnica de optimización como **PSO** es necesario definir la información que almacenará cada individuo, la manera en que será inicializada la población, cómo se efectuará el desplazamiento de las partículas dentro del espacio de búsqueda y cómo se controlará la convergencia prematura. En el capítulo 1 se definió **PSO** de forma general, sin embargo en este capítulo se detallará dicho método para la obtención de reglas de clasificación.

3.2. Representación de las Reglas

En toda estrategia adaptativa, el primer paso es indicar la representación que se va a utilizar. Esta representación va a depender directamente del problema que se desea resolver, lo que va a determinar el tipo de información que estará contenida en cada elemento de la población. Puede decirse que existen dos opciones:

- **Enfoque Pittsburgh:** (Smith, 1980a) Cada individuo de la población puede corresponderse con un conjunto de reglas. Aquí no se conoce con anticipación el número exacto de reglas que conforman el modelo provocando que los algoritmos utilizados consuman mayor cantidad de recursos e incrementen su complejidad. La ventaja de esta representación se basa en que, al ser un solo individuo, es más fácil evaluar la función de aptitud de la solución encontrada.
- Cada individuo de la población puede contener la información de una única regla. En este caso se pueden presentar las siguientes alternativas:
 - **Enfoque Michigan:** En este caso, cada individuo de la población codifica una única regla, sin embargo la solución final puede ser un subconjunto de la población o la población completa, (Smith, 1980b).
 - **Enfoque Aprendizaje Iterativo de Reglas (IRL):** Como su nombre lo dice se utiliza un proceso iterativo, en donde en cada ejecución se extrae al mejor individuo de la población, el mismo que contiene una única regla. Después que la regla ha sido obtenida, se quita del conjunto de ejemplos de entrada los casos correctamente cubiertos, y se repite nuevamente el proceso con los ejemplos restantes hasta llegar a tener el conjunto de reglas completo (Venturini, 1993). Una de las ventajas que presenta este enfoque es que evita generar reglas similares, sin embargo es importante tener en cuenta que el conjunto de reglas, deben ser aplicadas a los datos en el orden que se obtienen. Estas reglas forman lo que se conoce como lista de clasificación.

Tanto en el enfoque Michigan como en el **IRL**, la longitud del individuo es menor que en el enfoque Pittsburgh, debido a que cada individuo representa una única regla, sin embargo se debe considerar que se deberán evaluar varios individuos para lograr obtener la mejor solución.

El método propuesto en esta tesis sigue el enfoque **IRL**. Es importante indicar que el problema de la convergencia prematura a un único individuo se resolvió a través de ejecuciones sucesivas del mismo proceso sobre un conjunto de ejemplos cuya cardinalidad se va reduciendo a lo largo de las iteraciones..

Con respecto a la extracción de reglas de clasificación, para formar el antecedente de la regla se utiliza una representación de longitud fija que operará sobre atributos nominales y numéricos formada por dos partes:

- **PARTE BINARIA**: Permite identificar cuáles 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.

En lo referente al consecuente de la regla este no se codificará, de tal forma que si lo que se desea es cubrir 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 la i -ésima ejecución, el algoritmo descubra reglas sólo para la i -ésima clase, con $i = 1..k$.

3.3. Estructura de la Partícula

Si se desea utilizar **PSO** para obtener reglas de clasificación capaces de operar sobre atributos nominales y numéricos es necesario indicar cuáles son los atributos que formarán parte del antecedente. Si la variable seleccionada fuera numérica, también será preciso indicar cuál será el rango de valores a utilizar.

Para desplazarse en un espacio n -dimensional cada partícula p_i está formada por:

- $pBin_i = (pBin_{i1}, pBin_{i2}, pBin_{i3}, \dots, pBin_{in})$ es un vector binario que almacena la posición actual de la partícula, e indica cuáles son los ítems o condiciones que componen el antecedente de la regla según **PSO**.

- **Vectores Velocidad:**
 $V1_i = (V1_{i1}, V1_{i2}, \dots, V1_{in})$ y $V2_i = (V2_{i1}, V2_{i2}, \dots, V2_{in})$ Se combinan para indicar la dirección en la cual se moverá la partícula. Vector de valores reales.
- **Vector Mejor Posición**
 $pBestBin_i = (pBestBin_{i1}, pBestBin_{i2}, pBestBin_{i3}, \dots, pBestBin_{in})$ almacena la mejor solución encontrada por la partícula hasta el momento. Vector binario.
- $Fitness_i$ almacena el valor de aptitud de la solución actual.
- $FitnesspBest_i$ almacena el valor de aptitud de la mejor solución local encontrada hasta el momento. Vector $pBestBin_i$
- $pReal_i = (pReal_{i1}, pReal_{i2}, pReal_{i3}, \dots, pReal_{in})$ utilizado para los atributos numéricos, contiene los límites actuales de los intervalos los mismos que se encuentran previamente normalizados entre 0 y 1.
- $pBestReal_i = (pBestReal_{i1}, pBestReal_{i2}, pBestReal_{i3}, \dots, pBestReal_{in})$ almacena la mejor solución encontrada por la partícula para los límites de los intervalos.
- $sopBin_i = (sopBin_1, sopBin_2, sopBin_3, \dots, sopBin_n)$ 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 $fitness_i$.
- Velocidad $V3_i = (V3_{i1}, V3_{i2}, \dots, V3_{in})$ indica la dirección de cambio de $pReal_i$.

De la estructura de la partícula detallada anteriormente se puede ver que tiene una parte binaria y otra continua. Las velocidades $V1$ y $V2$ se combinan para determinar la dirección en que la partícula se moverá en el espacio discreto, y $V3$ se utiliza para mover la partícula en el espacio continuo.

Luego, cada vez que la i -ésima partícula se mueve, su posición actual cambia de la siguiente manera.

Parte binaria

$$V1_{ij}(t + 1) = W_{bin}V1_{ij}(t) + \varphi_1.rand_1(2pBestBin_{ij} - 1) + \varphi_2.rand_2(2lBestBin_{ij} - 1) \quad (3.1)$$

Donde:

- W_{bin} es el factor de inercia.
- φ_1 y φ_2 son valores constantes que indican la importancia asignada a las respectivas soluciones.
- $rand_1$ y $rand_2$ son valores en el intervalo $[0, 1]$, que se definen aleatoriamente.

- $pBestBin_{ij}$ y $lBestBin_{ij}$ son los j -ésimos dígitos de los vectores $pBestBin_i$ y $lBestBin_i$. Es importante considerar que cada partícula tiene en cuenta la posición de su vecino más cercano cuyo valor de aptitud sea superior al suyo, por lo que el valor de $lBestBin_i$ corresponde al vector de $pBestBin_j$ de la partícula que se encuentra más cercana a $pBestBin_i$ siempre que se cumpla que $Fitness_j > Fitness_i$ considerando la distancia euclídea.

Cabe señalar que, como se discutió en (Lanzarini y col., 2011) y a diferencia del método binario descrito en (Kennedy y Eberhart, 1997), el movimiento del vector $V1_i$ en las direcciones correspondientes a la mejor solución encontrada por la partícula y la mejor de su vecindario no dependen de la posición actual de la partícula. Luego, cada elemento del vector de velocidad $V1_i$ se calcula con la ecuación 3.2 y se controla con la ecuación 3.3.

$$V1_{ij}(t) = \begin{cases} \delta1_j & \text{if } V1_{ij}(t) > \delta1_j \\ -\delta1_j & \text{if } V1_{ij}(t) \leq \delta1_j \\ V1_{ij}(t) & \text{caso contrario} \end{cases} \quad (3.2)$$

$$\delta1_j = \frac{\text{limite1}_{superiorj} - \text{limite1}_{inferiorj}}{2} \quad (3.3)$$

Luego de esto se actualiza el valor del vector velocidad $V2_i$ como se indica en 3.4:

$$V2_{ij}(t + 1) = V2_{ij}(t) + V1_{ij}(t + 1) \quad (3.4)$$

El vector $V2_i$ se controla de manera similar al vector $V1_i$ cambiando $\text{limite1}_{superiorj}$ y $\text{limite1}_{inferiorj}$ por $\text{limite2}_{superiorj}$ y $\text{limite2}_{inferiorj}$ respectivamente. Esto dará lugar a $\delta2_j$ que se utilizará como en 3.2 para acotar los valores de $v2_i$. Luego se le aplica la función sigmoide 3.5 y se calcula la nueva posición de la partícula según 3.6

$$\text{sig}(X) = \frac{1}{1 + e^{-x}} \quad (3.5)$$

$$\rho Bin_{ij}(t + 1) = \begin{cases} 1 & \text{if } rand_{ij} < \text{sig}(V2_{ij}(t + 1)) \\ 0, & \text{if not} \end{cases} \quad (3.6)$$

Donde $rand_{ij}$ es un número aleatorio con distribución uniforme en $[0,1]$. Es importante indicar que la aplicación de la función sigmoide en la ecuación 3.6, cambia la forma en que se utiliza el vector velocidad para actualizar la posición de la partícula.

Parte continua

Esta parte controla los límites de los atributos numéricos. Para esto es necesario

considerar el valor de $V3_{ij}$ dado por la ecuación 3.7. Sumando luego al vector $V3_{ij}$ el valor de $pReal_{ij}$ como se indica en la 3.8.

$$V3_{ij}(t + 1) = W_{Real}V3_{ij}(t) + \varphi3.rand_3.(pBestReal_{ij} - pReal_{ij}) + \varphi4.rand_4.(lBestReal_{ij} - pReal_{ij}) \quad (3.7)$$

$$pReal_{ij}(t + 1) = pReal_{ij}(t) + V3_{ij}(t + 1) \quad (3.8)$$

Al igual que se indicó en la parte binaria, aquí W_{Real} representa el factor de inercia, $rand_3$ y $rand_4$ son valores aleatorios con distribución uniforme en $[0,1]$ y $\varphi3$ y $\varphi4$ son valores constantes que indican la importancia que se desea darle a las respectivas soluciones halladas previamente, $lBestReal_{ij}$ corresponde al vector $Real_{ij}$ de la misma partícula desde la que se tomó el vector Bin_{ij} para ajustar $V1_i$ con el vector $lBestBin_{ij}$ en la ecuación 3.2. $V3_{ij}$ así como $pReal_{ij}$ están controlados por $limite3_{superiorj}$ y $limite3_{inferiorj}$ por $limite4_{superiorj}$ y $limite4_{inferiorj}$, de manera similar a como se controlan los vectores velocidad $V1$ y $V2$ en la parte binaria.

Aunque el procedimiento que se sigue para actualizar los vectores $V2$ y $pReal_{ij}$ es el mismo (ecuaciones 3.8 y 3.4), los valores de $v2$ se utilizan como argumentos de la función sigmoide (ecuaciones 3.5 y 3.6). Esto permite obtener un valor dentro del intervalo $[0,1]$ que equivale a la probabilidad de que la posición de la partícula tome el valor 1. Por lo tanto, se podrán obtener probabilidades dentro del intervalo $[sig(limite2_{inferiorj}), sig(limite2_{superiorj})]$. La función sigmoide para valores extremos produce valores de probabilidad muy similares, cercanos a 0 o 1, reduciendo la posibilidad de cambio en los valores de las partículas y estabilizando la población.

3.3.1. Función Fitness

La función de aptitud o fitness para cada una de las partículas se calcula de acuerdo a lo que se indica en la ecuación 3.9.

$$FuncionFitness = (Soporte * Confianza * factor1) - (factor2 * \frac{NumAtribsAntecedente}{MaxAtribs}) \quad (3.9)$$

Siendo *Soporte* un valor numérico que mide la representatividad de la regla y *Confianza* otro valor numérico que representa su precisión. El soporte se calcula como el cociente entre la cantidad de ejemplos que la regla predice correctamente dividida por la cantidad total de ejemplos que se analizan. La confianza es el cociente entre la cantidad de veces que la respuesta de la regla fue correcta dividida por

la cantidad de veces que la regla fue aplicada. En ambos casos, se trata de valores numéricos pertenecientes al intervalo $[0,1]$. Si una regla tiene soporte 0 indica que es incorrecta en todos los casos; es decir, que ningún ejemplo las cumple ya sea porque no verifica el antecedente o el consecuente o ambos. Una regla con confianza 1 ha respondido siempre de manera correcta.

El *factor1* es un valor de multa o penalización en el caso de que el soporte no se encuentre dentro de los rangos establecidos en el algoritmo. El segundo término en la función fitness refleja la importancia que se da al número de atributos que forman el antecedente, siendo *factor2* una constante.

El objetivo de este método fue el de 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 facilita la interpretación del modelo ayudando a la toma de decisiones.

3.4. Construcción de Reglas utilizando PSO

El utilizar **PSO** para extraer reglas de clasificación a partir de un conjunto de ejemplos de entrenamiento requiere tener en cuenta algunas consideraciones especiales. La función de aptitud definida en la sección anterior es muy particular ya que pequeños cambios en el antecedente de una regla pueden generar grandes cambios en el valor de aptitud de una partícula. Incluso es posible realizar reiteradas modificaciones sobre una regla sin lograr mejoras en su valor de aptitud. Todo esto hace que la búsqueda de mejores soluciones no sea un proceso trivial. Por lo tanto, se consideró útil utilizar una red neuronal competitiva como herramienta para identificar las zonas más prometedoras del espacio de búsqueda donde resulta conveniente ubicar las partículas antes de comenzar el proceso (Lanzarini y col., 2015a). Se emplearon dos tipos de redes neuronales competitivas **SOM** y **LVQ** para poder analizar si era necesario trabajar con entrenamiento supervisado. Estas dos arquitecturas de redes se encuentran descritas con detalle en el Anexo A.

A partir de este momento empieza un proceso iterativo que puede utilizar cualquiera de las variantes de **PSO** para extraer reglas correspondientes a la clase con la mayor cantidad de ejemplos. En particular se trabajó tanto con cúmulos de tamaño fijo como de tamaño variable. Puede consultarse el Anexo B para una descripción detallada de cada una de estas variantes.

Como se indicó en la sección anterior, sólo el antecedente se encuentra representado en la partícula ya que el consecuente se fija antes de comenzar la evolución del cúmulo.

Luego de que la clase ha sido seleccionada se aplica la técnica de optimización para encontrar el antecedente de la primera regla.

La población de partículas se inicializa utilizando la red neuronal competitiva que fue entrenada. Cada neurona contendrá información de la ubicación del centroide, y de la dispersión de los ejemplos que representa. La información del centroide se utiliza para determinar el vector $V2$ indicado en la sección anterior. Si se trata de un atributo nominal la información se escala entre el $limite2_{inferior_j}$ y $limite2_{superior_j}$, en el caso de que el atributo sea 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 los dos casos se opera con valores entre 0 y 1, para el caso de los atributos numéricos mide el grado de participación del atributo, o el 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. El valor de cada una de esas posiciones, coincidirá con la proporción de ejemplos representados que lo posean. Al escalarlo linealmente entre $limite2_{inferior_j}$ y $limite2_{superior_j}$ tendrán mayor posibilidad de ser elegidos para formar parte del antecedente, los que se encuentren presentes en un mayor número de ejemplos. Para los atributos numéricos, la desviación es la que permite determinar la cercanía del valor del ejemplo con el centroide, teniendo mayor oportunidad de ser seleccionados, aquellos que poseen desviación nula. La velocidad $V1$ se inicializa de forma aleatoria, entre $limite2_{inferior_j}$ y $limite2_{superior_j}$. La medida de similitud utilizada es la distancia euclídea.

Luego que se ha creado la población inicial, se la evoluciona de acuerdo a la versión de PSO que se seleccione ya sea este de población fija o de población variable. Es importante indicar que en el caso de que se trabaje con PSO de población variable se deben identificar las partículas aisladas para luego agregar una variante del centroide que originó cada una de ellas. Esto tiene por objetivo incentivar la búsqueda en aquellas zonas que han sido menos exploradas. Además cada partícula tiene un tiempo de vida que disminuirá con cada iteración; cuando el tiempo de vida sea cero, la partícula será eliminada de la población. El Anexo B contiene una descripción detallada de las distintas versiones de PSO relacionadas con esta tesis.

Una vez que el proceso ha finalizado, se procede a seleccionar la mejor regla de la población. Si esta regla cumple con el soporte y la confianza indicados, se la agrega al conjunto de reglas y se procede a retirar del conjunto de ejemplos de entrenamiento aquellos que son cubiertos correctamente por esta regla. En este último caso, se reducen los ejemplos que quedan sin cubrir dentro de la clase.

Para que el proceso finalice se pueden tener las siguientes condiciones de parada, la que ocurra primero: que se cubran todos los ejemplos, que la cantidad de ejemplos no cubiertos de cada clase se encuentren por debajo del respectivo soporte mínimo establecido o que se haya realizado la máxima cantidad de intentos para obtener una regla.

Como el proceso es iterativo, a medida que avanza se va reduciendo la cardinalidad del conjunto de ejemplos. Luego, para clasificar un ejemplo nuevo, las reglas

deben ser inspeccionadas para su aplicación en el orden en que fueron obtenidas, constituyendo una lista de reglas de clasificación. Esto tiene que ver con que la información utilizada para generar cada regla es diferente a medida que el proceso adaptativo avanza. Finalmente, el ejemplo será clasificado con la clase correspondiente al consecuente de la primera regla cuyo antecedente se verifique para el ejemplo en cuestión.

Este método, utiliza el Pseudocódigo descrito en el algoritmo 3.

Algoritmo 3: Pseudocódigo del método

```
begin  
  Entrenar la red neuronal competitiva utilizando todos los ejemplos  
  de entrenamiento;  
  Calcular el soporte mínimo para cada clase;  
  while no se alcanza la condición de terminación do  
    Elegir la clase con mayor nro.de ejemplos no cubiertos  
    Construir una población reducida de individuos a partir de los centroides  
    Evolucionar la población utilizando la versión de PSO seleccionada  
    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  
Result: la mejor solución encontrada
```

3.5. Reglas de clasificación aplicadas a Riesgo Crediticio

La economía en el mundo actual conlleva a que las personas soliciten créditos, con diferente finalidad como son los créditos productivo, comercial, de consumo, vivienda, inmobiliario, micro-crédito e incluso crédito de inversión pública, entre otros. Esto lleva a que las instituciones financieras analicen una gran cantidad de variables micro-económicas que le permitan realizar el análisis del sujeto de crédito, y así poder dar una respuesta sobre el crédito solicitado, y de acuerdo a su capacidad financiera establecer su forma de pago, tratando de disminuir al máximo el riesgo presente en el otorgamiento del crédito.

En el caso de Riesgo Crediticio las variables que intervienen en el proceso de concesión de crédito, son analizadas por los oficiales de crédito a través de varios métodos entre los que se encuentran el score crediticio, donde se le da un puntaje a cada una de las variables analizadas. Sin embargo este análisis lleva tiempo, recursos y en muchos casos depende del punto de vista del oficial de crédito. Todo esto lleva a que aquellas instituciones financieras que dan el resultado al cliente en el menor tiempo posible, con una menor disminución del riesgo, presenten una ventaja

competitiva frente a otras instituciones financieras, logrando incrementar sus colocaciones. Es por esta razón que resulta de interés contar con técnicas adecuadas que ayuden al oficial de crédito a tomar una decisión.

El método anteriormente detallado, que combina una red neuronal con una técnica de optimización como **PSO**, se presenta como una alternativa para reducir el Riesgo Crediticio al ofrecer un conjunto de reglas de cardinalidad reducida y fácil interpretación con una precisión aceptable. El modelo generado constituye una gran ayuda para el oficial de crédito ya que puede responder con mayor exactitud en un tiempo menor. Esto último se debe a que debe verificar un menor número de reglas. Como resultado de lo antes dicho se logra una disminución en el riesgo crediticio. Las pruebas realizadas sobre dos bases de datos del repositorio **UCI Machine Learning Repository (UCI)** (Dua y Karra Taniskidou, 2017) y dos bases de datos reales de Instituciones Financieras del Ecuador permiten afirmar que este método es una excelente alternativa en esta área.

En el caso de las bases de datos de las Instituciones Financieras, una corresponde a una Cooperativa de ahorro y crédito y la otra a una Institución Financiera encargada de dar crédito de consumo no productivo. En ambos casos se analizaron solicitudes de crédito así como operaciones de crédito concedidas, con los siguientes atributos: estado, fecha de la solicitud, destino del crédito, provincia, monto requerido, monto autorizado, propósito del crédito, efectivo con el que cuenta el cliente, cuentas bancarias, inversiones, otros activos, pasivos y sueldo del solicitante, fecha de la verificación de la información, fecha de autorización, fecha de la aprobación/negación, cuentas bancarias, inversiones, otros activos, pasivos y sueldo del cónyuge del solicitante. En el caso de que el solicitante sea un pequeño negocio los datos solicitados son los ingresos y los gastos del negocio.

Las solicitudes de crédito pueden ser negadas o aceptadas. En caso de ser aceptadas, se realiza una nueva clasificación entre los créditos que fueron cancelados sin ninguna novedad y los que tienen algún retraso en la recuperación de la inversión. A su vez, los créditos vencidos se clasifican de acuerdo con las políticas crediticias entre los que tienen menos de 90 días de retraso, y los que tienen más de 90 días de retraso (inicio de acciones legales). Estos últimos son los que se consideran que pasan a cartera vencida.

En el caso de las bases de datos de las dos Instituciones Financieras, se tomaron datos correspondientes al período que va desde enero 2011 hasta agosto 2014, conteniendo la información de las solicitudes de crédito que fueron aceptadas o negadas. En el caso de los créditos aceptados, se tenía adicionalmente información sobre el reembolso del crédito, por lo que se consideró el tiempo de morosidad del crédito de la siguiente forma: aquellos créditos que no presentaban morosidad o los que la morosidad fuera menor a 90 días, fueron consideradas como solicitudes que debían ser aceptadas, y aquellas que tenían morosidad superior como créditos que no debían ser concedidos (inicio de acciones legales), que son aquellos que se consideran

que pasan a cartera vencida. En este caso todas las variables analizadas para la concesión del crédito, fueron micro-económicas. En el caso de la cooperativa de ahorro y crédito, los datos analizados fueron desde enero 2011 a diciembre 2015.

Para las bases de datos del repositorio UCI (Dua y Karra Taniskidou, 2017), se consideraron las bases de datos Alemana y Australiana.

Las características de todas las tablas analizadas en este estudio se indican en la tabla (3.1). Para cada una de ellas se detallan: la cantidad de tuplas, variables difusas, número de conjuntos difusos, y el número de clases en las que los registros pueden ser clasificados.

TABLA 3.1: Descripción de Bases de datos utilizadas

Bases de datos	# Registros	# Var. Nominales	# Var. Numéricas	# Clases
Australiana	653	6	9	2
Alemana	1000	7	13	2
Banco del Ecuador	36356	18	3	2
Coop. Ahorro Crédito Ecuador	22473	18	3	2

El uso de los datos descriptos anteriormente, compara el rendimiento de varios métodos que combinan dos tipos de PSO, el de población fija y otro de población variable, inicializadas con dos redes neuronales competitivas diferentes: LVQ y SOM. Se comparan estas soluciones con los métodos C4.5 y PART (véase sección 1.6.1).

3.5.1. Resultados Obtenidos

El desempeño de las variantes de los métodos de extracción de reglas de clasificación basados en PSO y Variable Particle Swarm Optimization (varPSO) fueron medidos luego de inicializar la población con cada uno de los tipos de red neuronal competitiva antes mencionados, SOM y LVQ.

Los resultados obtenidos fueron comparados con los arrojados por los algoritmos C4.5 y PART descriptos en 1.6.1 y se resumen en las siguientes tablas (3.2, 3.3, 3.4,3.5).

Para obtener más información se puede revisar (Jimbo y col., 2016) y (Lanzarini y col., 2017).

3.6. Conclusiones

En este capítulo se ha presentado un método de obtención de reglas de clasificación, basado en la combinación de PSO y redes neuronales competitivas SOM y LVQ. El desarrollo de este capítulo cobra importancia, debido a que el desempeño

TABLA 3.2: Resultados con base de datos Australiana del repositorio UCI

Método	Verdadero +	Verdadero -	Falso +	Falso -	Precisión	#Reglas	Long. del antecedente
SOM+PSO	0.4257 ±0.0154	0.4333 ±0.0103	0.1097 ±0.0069	0.0309 ±0.0066	0.8590 ±0.0099	3.0167 ±0.0461	1.3711 ±0.1922
SOM+varPSO	0.4183 ±0.0132	0.4391 ±0.0158	0.1071 ±0.0130	0.0351 ±0.0077	0.8574 ±0.0104	3.0000 ±0.0000	1.5178 ±0.1085
LVQ + PSO	0.4201 ±0.0179	0.4414 ±0.0172	0.1079 ±0.0093	0.0306 ±0.0065	0.8614 ±0.0105	3.0000 ±0.0000	1.2667 ±0.1207
LVQ + varPSO	0.4199 ±0.0179	0.4382 ±0.0172	0.1054 ±0.0075	0.0363 ±0.0073	0.8582 ±0.0092	3.0000 ±0.0000	1.5578 ±0.1336
C4.5	0.3910 ±0.0121	0.4618 ±0.0063	0.0847 ±0.0066	0.0625 ±0.0120	0.8528 ±0.0124	18.2200 ±2.0825	4.8394 ±0.2810
PART	0.3564 ±0.0136	0.3906 ±0.0288	0.1562 ±0.0289	0.0969 ±0.0134	0.7469 ±0.0292	33.3433 ±1.5793	2.4926 ±0.0934

TABLA 3.3: Resultados con base de datos Alemana del repositorio UCI

Método	Verdadero +	Verdadero -	Falso +	Falso -	Precisión	#Reglas	Long. del antecedente
SOM+PSO	0.1026 ±0.0123	0.5993 ±0.0183	0.0984 ±0.0149	0.1994 ±0.0183	0.7019 ±0.0153	8.4400 ±0.6009	2.1619 ±0.1415
SOM+varPSO	0.1046 ±0.0115	0.5954 ±0.0135	0.1034 ±0.0110	0.1965 ±0.0134	0.7000 ±0.0162	8.0233 ±0.6745	2.0464 ±0.1030
LVQ + PSO	0.0999 ±0.0151	0.5997 ±0.0171	0.1017 ±0.0136	0.1986 ±0.0153	0.6996 ±0.0133	8.7767 ±0.7224	2.1802 ±0.1075
LVQ + varPSO	0.1089 ±0.0100	0.5973 ±0.0129	0.1057 ±0.0103	0.1880 ±0.0116	0.7063 ±0.0109	8.8867 ±0.4918	2.0884 ±0.0960
C4.5	0.1219 ±0.0069	0.5894 ±0.0070	0.1106 ±0.0070	0.1781 ±0.0069	0.7113 ±0.0079	86.4600 ±4.0788	5.6267 ±0.1382
PART	0.1404 ±0.0120	0.4385 ±0.0091	0.1687 ±0.0135	0.2258 ±0.0170	0.6967 ±0.0139	70.9133 ±2.1575	3.0138 ±0.0561

TABLA 3.4: Resultados con base de datos de Cooperativa de Ahorro y Crédito del Ecuador

Método	Verdadero +	Verdadero -	Falso +	Falso -	Precisión	#Reglas	Long. del antecedente
SOM+PSO	0.6242 ±0.0069	0.1601 ±0.0062	0.1253 ±0.0057	0.0898 ±0.0059	0.7844 ±0.0059	3.7867 ±0.2980	1.6375 ±0.2151
SOM+varPSO	0.6014 ±0.0052	0.1914 ±0.0059	0.0947 ±0.0057	0.1125 ±0.0047	0.7928 ±0.0030	4.1533 ±0.2801	1.6953 ±0.0867
LVQ + PSO	0.6227 ±0.0048	0.1671 ±0.0055	0.1191 ±0.0051	0.0910 ±0.0039	0.7899 ±0.0031	3.2933 ±0.1837	1.4021 ±0.1066
LVQ + varPSO	0.6029 ±0.0056	0.1902 ±0.0055	0.0956 ±0.0054	0.1114 ±0.0053	0.7930 ±0.0025	4.3733 ±0.2625	1.6553 ±0.0567
C4.5	0.6320 ±0.0014	0.1786 ±0.0013	0.1075 ±0.0013	0.0819 ±0.0013	0.8106 ±0.0011	114.2600 ±6.0543	9.6762 ±0.1144
PART	0.6229 ±0.0065	0.1825 ±0.0064	0.1036 ±0.0064	0.0910 ±0.0065	0.8054 ±0.0023	42.3567 ±2.1661	4.6956 ±0.0880

TABLA 3.5: Resultados con base de datos de Institución Financiera Banco del Ecuador que se encarga de dar crédito de consumo productivo y no productivo

Método	Verdadero +	Verdadero -	Falso +	Falso -	Precisión	#Reglas	Long. del antecedente
SOM+PSO	0.0457 ±0.0058	0.8863 ±0.0047	0.0228 ±0.0045	0.0451 ±0.0056	0.9320 ±0.0050	4.3967 ±0.4895	5.4962 ±0.4221
SOM+varPSO	0.0609 ±0.0029	0.8870 ±0.0059	0.0210 ±0.0051	0.0300 ±0.0030	0.9480 ±0.0063	3.8600 ±0.2415	2.8940 ±0.3532
LVQ + PSO	0.0482 ±0.0052	0.8870 ±0.0056	0.0219 ±0.0055	0.0428 ±0.0049	0.9352 ±0.0054	4.6067 ±0.4193	5.9166 ±0.2771
LVQ + varPSO	0.0565 ±0.0043	0.8882 ±0.0062	0.0198 ±0.0056	0.0346 ±0.0043	0.9447 ±0.0056	3.8533 ±0.2921	3.1013 ±0.3395
C4.5	0.0762 ±0.0003	0.9017 ±0.0003	0.0073 ±0.0003	0.0148 ±0.0003	0.9779 ±0.0003	153.5733 ±5.1687	11.2349 ±0.1565
PART	0.0457 ±0.0058	0.8863 ±0.0047	0.0228 ±0.0045	0.0451 ±0.0056	0.9320 ±0.0050	4.3967 ±0.4895	5.4962 ±0.4221

de cada uno de estos métodos es comparado con el arrojado por el método descrito en el capítulo siguiente. Este último constituye el aporte central de esta tesis.

Su aplicación, se ha realizado con dos bases de datos de créditos reales, la primera de una Cooperativa de Ahorro y Crédito, y la otra de un Banco del mercado Financiero Ecuatoriano, y de dos bases de datos públicas presentes en el repositorio UCI. Las mediciones realizadas permiten afirmar, que el método descrito logra reducir significativamente el número de reglas que se requiere alcanzando un nivel de precisión aceptable.

En el caso de Riesgo Crediticio los resultados obtenidos sugieren que la simplificación de las reglas de decisión genera transparencia en la puntuación de crédito, lo que podría mejorar la reputación de las instituciones financieras, disminuyendo el riesgo al que se enfrentan. Estos modelos logran alcanzar una buena precisión, utilizando un número reducido de reglas, las mismas que son simples en su interpretación (menor número de atributos analizados del sujeto de crédito) disminuyendo el tiempo en el análisis de la solicitud de crédito. Esto ayuda a la toma de decisiones de los oficiales de crédito, presentando una ventaja competitiva para las instituciones financieras, ya que el tiempo en dar la respuesta a una solicitud de crédito se reduce considerablemente.

Capítulo 4

Método propuesto FRvarPSO

Es importante empezar indicando que en el mundo real los problemas son lo suficientemente complejos como para que no exista una única solución que permita resolverlos a todos. Por lo tanto, *"los problemas complejos deben ser divididos en partes, y cada una de estas partes debe resolverse con la técnica que mejor se adapte a sus necesidades"* (Pérez, 2010). La tendencia hacia los sistemas denominados híbridos es clave en la Inteligencia Computacional.

El modelo que se describe en este capítulo constituye el aporte central de esta tesis y se denomina **Fuzzy Rules Variable Particle Swarm Optimization (FRvarPSO)**. Corresponde a la categoría de modelo híbrido - predictivo, ya que combina varias técnicas y es capaz de dar una respuesta ante un nuevo requerimiento ofreciendo como resultado un conjunto de reglas de clasificación difusas.

Las reglas generadas con **FRvarPSO** utilizan el formato tipo Mandani descrito en la sección 2.3.6 para codificar el antecedente mientras que el consecuente continua siendo crisp. Esto último se debe a que el atributo de clase es cualitativo. Por lo tanto, una regla generada con **FRvarPSO** tendrá la siguiente forma:

$$R^{(l)} : \text{IF } x_1 \text{ is } F_1^{(l)} \text{ and } \dots \text{ and } x_n \text{ is } F_n^{(l)} \text{ THEN } y = \text{valor_de_clase} \quad (4.1)$$

donde $F_i^{(l)}$ son conjuntos difusos contenidos en los universos $U_i \subset \mathbb{R}$, $x = (x_1, \dots, x_n)^T \in U_1 \times \dots \times U_n$ son variables lingüísticas e y es el atributo que representa la respuesta esperada.

Nótese que por tratarse de un modelo de clasificación, tal como se explicó en la sección 1.5.2, el consecuente de todas las reglas deberá referirse a un mismo atributo o característica, el cual corresponde a la respuesta esperada (Aggarwal, 2015).

De esta forma, se tendrá un modelo basado en reglas de clasificación difusas en las cuales cada antecedente estará formado por atributos cualitativos y/o condiciones difusas evitando, de esta forma, el uso de umbrales rígidos.

El método **FRvarPSO**, propuesto en esta tesis como herramienta para generar reglas de Clasificación Difusas, combina una red neuronal competitiva (ya sea **LVQ** o

SOM) con una técnica de optimización basada en cúmulo de partículas de tamaño variable **varPSO**. Esta combinación busca resolver uno de los problemas más importantes de la optimización utilizando **PSO**: la inicialización del cúmulo. Las redes neuronales competitivas utilizadas, **LVQ** y **SOM**, son dos alternativas diferentes capaces de identificar el punto de partida adecuado desde donde deberá comenzar a operar la técnica de optimización.

Es importante destacar que la extracción de reglas por medio de una técnica de optimización, a diferencia de los métodos de construcción tradicionales, no se basa en la división sucesiva de la información disponible, sino que selecciona en forma simultánea las condiciones cuya conjunción dará lugar al antecedente de cada regla.

Para medir el desempeño de **FRvarPSO** se utilizaron dos medidas de comparación: la precisión alcanzada y el número de reglas de clasificación generadas.

Versiones anteriores de este método pueden hallarse en (Lanzarini y col., 2017), explicado en detalle en el capítulo 3. En esa oportunidad, el modelo construido fue una lista de reglas de clasificación de cardinalidad baja y precisión aceptable en comparación con los métodos convencionales. El énfasis puesto en la simplicidad de interpretación de la regla estuvo limitado a la construcción de reglas con pocas condiciones en el antecedente.

FRvarPSO es una mejora del método anterior que incorpora el uso de variables lingüísticas con el objetivo de simplificar la interpretación por parte de quien debe tomar decisiones haciendo uso del modelo, manteniendo la baja cardinalidad del conjunto de reglas. Además, las modificaciones incorporadas utilizando lógica difusa han incrementado la precisión en la predicción por lo que considera a esta propuesta una versión superadora con respecto a las anteriores.

FRvarPSO fue probado en doce bases de datos del repositorio **UCI**, y en tres bases de datos reales de Entidades Financieras del Ecuador. Las mismas que corresponden a una Cooperativa de Ahorro y Crédito que se encuentra regulada dentro de la Superintendencia de Economía Popular y Solidaria, y dos Bancos que otorgan diferentes tipos de crédito. Para medir su desempeño se comparó con otros métodos los cuales utilizaron diferentes versiones de **PSO** entre las que se encuentran el **PSO** de población fija y el de población variable **varPSO**. Las versiones convencionales, generalmente utilizadas para optimización de funciones, se encuentran descriptas en detalle en el Anexo B.

4.1. Algoritmo de extracción de reglas propuesto

FRvarPSO se caracteriza especialmente por incorporar la lógica difusa como mecanismo para conducir la manera en que se mejoran las soluciones a través de **varPSO**. Para ello es preciso definir un fuzzificador que determine, para cada valor de cada atributo numérico, los grados de pertenencia de dicho valor a cada conjunto

difuso asociado a sus etiquetas lingüísticas. Es importante ver que con esta transformación se logra una reducción de la granularidad de los dominios de los atributos numéricos.

4.1.1. Variables lingüísticas. Etiquetas y conjuntos difusos

A base de lo antes dicho, el primer paso consiste en definir las particiones correspondientes a las variables lingüísticas de entrada.

Para lograrlo, una solución ideal sería contar con la posibilidad de definir, conjuntamente con el experto, las etiquetas lingüísticas y los conjuntos difusos de cada una de las variables de entrada (Yun, Sik y Bongju, 2013). Sin embargo, esto no siempre es posible. A continuación se detallan un par de alternativas que fueron utilizadas en esta tesis para trabajar con las bases de datos de repositorio ya que no se contaba, para ninguna de ellas, con el asesoramiento del experto antes mencionado. La primera efectúa una partición del rango de cada atributo numérico y la segunda utiliza una técnica de clustering difuso.

Particiones Uniformes

Para esta alternativa se particionó el rango de cada atributo numérico en intervalos de igual longitud y centrandlo, en cada uno de ellos, una función triangular con un solapamiento adecuado. De tal modo, se consideraron particiones difusas uniformes con el mismo número de etiquetas para cada una de las variables numéricas analizadas. Se dividió el rango de cada variable numérica en 9 intervalos iguales. Luego, se formó el primer conjunto difuso con los cuatro primeros intervalos. El segundo conjunto difuso estuvo formado por los valores comprendidos entre la mitad del tercer intervalo y hasta la mitad del séptimo intervalo. Finalmente, el tercer conjunto difuso estuvo formado por los últimos cuatro intervalos. De esta manera, quedan conformados los 3 conjuntos difusos con un solapamiento entre las fronteras, representando lo que ocurre en el razonamiento humano.

Siguiendo la propuesta de construcción anterior, cada una de las variables numéricas puede tener como máximo dos grados de pertenencia a dos de los conjuntos difusos definidos, en el tercer conjunto difuso su grado de pertenencia será cero.

Fuzzy C-Means

FCM es una técnica de agrupamiento definida en (Bezdek, Ehrlich y Full, 1984). Como resultado de su aplicación se obtendrá un modelo basado en centroides y se utilizarán funciones para medir el grado de pertenencia de cada ejemplo en cada grupo. Es decir que a diferencia de los métodos de agrupamiento partitivo, un

mismo ejemplo puede pertenecer a varios grupos llevando asociados sus correspondientes grados de pertenencia dando lugar a particiones difusas.

En un agrupamiento difuso, cada cluster es considerado un conjunto difuso. Uno de los parámetros de entrada del algoritmo FCM es el número C de clusters. En esta investigación se utilizaron 3 clusters. Al momento de utilizarlo, deben tenerse en cuenta las siguientes consideraciones:

- La suma de los grados de pertenencia del i -ésimo ejemplo a los distintos conjuntos difusos debe ser 1.
- El grado de pertenencia del i -ésimo ejemplo en el cluster j , denominado μ_{ij} , debe encontrarse entre 0 y 1.

Durante la adaptación, FCM minimiza la siguiente función objetivo:

$$J = \sum_{i=1}^N \sum_{j=1}^C \mu_{ij} \cdot \|X_i - C_j\|^2 \quad (4.2)$$

donde N es la cantidad de ejemplos, C es el número de clusters a formar, C_j es el centroide del cluster j y μ_{ij} es el grado de pertenencia para el i -ésimo ejemplo X_i en el cluster j . La norma $\|X_i - C_j\|$ se encarga de medir la cercanía del ejemplo X_i al cluster j .

El grado de pertenencia μ_{ij} del ejemplo X_i , al cluster j , se calcula de la siguiente forma:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^C \left[\frac{\|X_i - C_j\|}{\|X_i - C_k\|} \right]^{\frac{2}{m-1}}} \quad (4.3)$$

De tal forma que m es considerado el coeficiente de fuzzificación. Este valor indica cuántos clusters pueden superponerse. Mientras mayor sea el valor de m , mayor será el solapamiento entre los clusters.

El centroide C_j se obtiene:

$$C_j = \frac{\sum_{i=1}^N \mu_{ij}^m X_i}{\sum_{i=1}^N \mu_{ij}^m} \quad (4.4)$$

μ_{ij} se encuentra en el intervalo entre $0 \leq \mu_{ij} \leq 1$, de tal forma que la:

$$\sum_{j=1}^C \mu_{ij} = 1 \quad (4.5)$$

4.1.2. Estructura de una Partícula

Para poder operar con atributos cualitativos y/o condiciones difusas se utiliza información binaria con el fin de indicar cuáles son los atributos que se van a utilizar en la regla.

En el caso de los atributos cualitativos que se representan de forma binaria, se los codifica de tal manera que cada uno de ellos tiene tantos dígitos binarios como valores diferentes pueda tomar la variable. Para el caso de las condiciones difusas se utilizan tres dígitos binarios con valor 1 en caso de que dicha condición se utilice y 0 si no.

Cada partícula p_i contiene la siguiente información:

- $pBin_i = (pBin_{i1}, pBin_{i2}, pBin_{i3}, \dots, pBin_{in})$ es un vector binario que almacena la posición actual de la partícula, e indica cuáles son los ítems o condiciones que componen el antecedente de la regla según **PSO**.
- Vectores Velocidad $V1_i = (V1_{i1}, V1_{i2}, \dots, V1_{in})$ y $V2_i = (V2_{i1}, V2_{i2}, \dots, V2_{in})$ Se combinan para indicar la dirección en la cual se moverá la partícula. Ambos contienen valores reales.
- $pBestBin_i = (pBestBin_{i1}, pBestBin_{i2}, pBestBin_{i3}, \dots, pBestBin_{in})$ almacena la mejor solución encontrada por la partícula hasta el momento. Vector binario.
- $Fitness_i$ almacena el valor de aptitud de la solución actual.
- $FitnesspBest_i$ almacena el valor de aptitud de la mejor solución local encontrada hasta el momento. Vector $pBestBin_i$
- $sopBin_i = (sopBin_{i1}, sopBin_{i2}, \dots, sopBin_{in})$ 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.
- CV_i Criterio de Votación almacena la información de los grados de pertenencia promedio de los ejemplos que cumplen la regla.

Cuando se trabaja con **PSO** las partículas se desplazan dentro del espacio de búsqueda utilizando su propia experiencia y la de su entorno. En el método propuesto en esta tesis se agrega un **Criterio de votación (CV)** para enfatizar el movimiento hacia posiciones recomendadas por los ejemplos que cumplen con la regla representada por la partícula .

La identificación de los atributos que forman el antecedente de la regla se realiza por medio del vector binario $pBin_{ij}$. Su valor es controlado a través de dos vectores velocidad $V1_i$ y $V2_i$.

La ecuación 4.6 indica la forma en que se actualiza el vector $V1_i$ en cada iteración:

$$V1_{ij}(t+1) = W * V1_{ij}(t) + \varphi1 \text{ rand}_1 (2 * pBestBin_{ij} - 1) + \varphi2 \text{ rand}_2 (2 * lBestBin_{ij} - 1) + CV_{ij} \quad (4.6)$$

donde W es el factor de inercia, rand_1 y rand_2 son valores aleatorios con distribución uniforme en $[0,1]$, $\varphi1$ y $\varphi2$ son valores constantes que indican la importancia asignada a las respectivas soluciones; $pBestBin_i$ es la mejor solución hallada por la i -ésima partícula y $lBestBin_i$ es la mejor solución hallada por la partícula más cercana, usando distancia euclídea, que tenga un valor de aptitud superior. En otras palabras, $lBestBin_i$ será a igual a $pBestBin_k$ si $Fitness_k > Fitness_i$. CV_i es el criterio de votación, que se se actualiza de acuerdo a la ecuación 4.14.

Como se observa en la ecuación 4.6 y a diferencia del método binario descrito en (Kennedy y Eberhart, 1997), el movimiento del vector $V1_i$ no depende de la posición actual de la partícula sino que se realiza en las direcciones indicadas por su mejor solución encontrada y por la mejor de su vecindario. Luego, cada elemento del vector $V1_i$ es controlado según la ecuación definida en 4.7 que utiliza el vector umbral $\delta1$ calculado según la ecuación 4.8.

$$V1_{ij}(t) = \begin{cases} \delta1_j & \text{if } V1_{ij}(t) > \delta1_j \\ -\delta1_j & \text{if } V1_{ij}(t) \leq \delta1_j \\ V1_{ij}(t) & \text{caso contrario} \end{cases} \quad (4.7)$$

donde

$$\delta1_j = \frac{\text{limite1}_{superiorj} - \text{limite1}_{inferiorj}}{2} \quad (4.8)$$

Luego de esto se actualiza el valor del vector velocidad $V2_i$ como se indica en la ecuación 4.9.

$$V2_{ij}(t+1) = V2_{ij}(t) + V1_{ij}(t+1) \quad (4.9)$$

El vector $V2_i$ se controla como en 4.7 utilizando $\delta2$ en lugar de $\delta1$. El valor de $\delta2$ se calcula como en 4.8 cambiando $\text{limite1}_{inferior}$ y $\text{limite1}_{superior}$ por $\text{limite2}_{inferior}$ y $\text{limite2}_{superior}$ respectivamente.

Luego se le aplica la función sigmoide 4.10 y se calcula la nueva posición de la partícula según 4.11.

$$\text{sig}(x) = \frac{1}{1 + e^{-x}} \quad (4.10)$$

$$\rho Bin_{ij}(t+1) = \begin{cases} 1 & \text{if } rand_{ij} < \text{sig}(V2_{ij}(t+1)) \\ 0, & \text{if not} \end{cases} \quad (4.11)$$

donde $rand_{ij}$ es un número aleatorio con distribución uniforme en $[0, 1]$, y $\text{sig}(V2_{ij})$ es la probabilidad de que el elemento ρBin_{ij} perteneciente al vector que contiene la posición actual se convierta en 1.

Es importante indicar que en PSO binario los valores extremos al ser mapeados por la función sigmoide produce valores de probabilidad similares, cercanos entre 0 y 1, reduciendo la probabilidad de cambio en los valores de la partícula. Si la velocidad de una partícula se encuentra determinada por un vector nulo, cada uno de los dígitos binarios que determinan su posición tiene la probabilidad de 0,5 de cambiar a 1, esto es lo más aleatorio que puede ocurrir.

Es importante tener en cuenta que aunque el movimiento de la partícula sigue siendo real, es la función sigmoide la que se encarga de binarizar el resultado identificando la nueva posición. En este trabajo se limitó el vector velocidad $V1$ al rango entre $[-0,5, 0,5]$. Los valores de $\varphi1$, $\varphi2$ para la parte binaria fueron fijados en el valor de 0,25. En el caso del factor de inercia W se estableció entre 1.25 y 0.25 de forma ascendente para W .

Se utiliza W_{Bin} para actualizar el vector velocidad $V1$. W_{Bin} se actualizan dinámicamente según la ecuación 4.12:

$$W = W_{inicial} - \frac{(W_{inicial} - W_{final})}{Total\ Iteraciones} * Iteracion\ actual \quad (4.12)$$

El método FRvarPSO propuesto en esta tesis utiliza varPSO. Para obtener un mayor detalle de este método se recomienda consultar el Anexo B.

En la tabla 4.1 se ejemplifica el antecedente de una regla difusa para la base de datos del repositorio UCI denominada *credit_a*. Esta base de datos tiene 15 atributos de los cuales 9 son cualitativos y 6 son numéricos. hay que considerar que cada atributo numérico tiene tres posiciones donde cada una de ellas representa a un conjunto difuso. Mientras que los cualitativos ocupan tantas posiciones como valores distintos tengan.

En la regla que se presenta como ejemplo tres atributos presentan el valor 1 en la parte binaria. Estos atributos conforman el antecedente de la regla. El vector binario es el que indica si se deben utilizar o no los atributos, para cada uno de los atributos

cualitativos se tiene máximo un único dígito binario en 1.

Para la regla $Si (A2 = medio) Y (A9 = f) Y (A14 = alta)$

TABLA 4.1: Antecedente de una regla difusa

Individuo	...	A2b	A2m	A2a	...	A91	A92	A14b	A14m	A14a
Parte binaria	...	0	1	0	...	1	0	0	0	1

$A2m$ condición difusa perteneciente al conjunto difuso medio con grado de pertenencia promedio de 0.9928

$A9$ variable nominal que toma los valores de "f", "t"

$A14a$ condición difusa perteneciente al conjunto difuso alto con grado de pertenencia promedio de 0.7700

En este ejemplo se puede observar en la parte binaria que tres atributos y/o condiciones difusas tienen el valor 1, siendo estos los que conforman el antecedente de la regla, como se puede observar este vector es el que indica que atributos son los que deben o no utilizarse.

Representación del Consecuente

Para la representación del consecuente una de las alternativas que se presentan en (Holden y Freitas, 2008) 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 difusas que deban cubrir un conjunto k de clases diferentes se deberá ejecutar el algoritmo al menos k veces de manera que en la i -ésima ejecución, el algoritmo descubra reglas difusas sólo para la i -ésima clase, con $i = 1..k$. Por lo tanto no se incluirá el consecuente de la regla en la representación del individuo.

Función de Aptitud

El valor de aptitud o fitness de cada partícula se calcula de acuerdo a la ecuación 4.13:

$$FuncionFitness = (Soporte * Confianza * factor1) - \left(factor2 * \frac{NumAtribsAntecedente}{MaxAtribs} \right) \quad (4.13)$$

Siendo *Soporte* un valor numérico que mide la representatividad de la regla difusa y *Confianza* otro valor numérico que representa su precisión. El soporte se calcula como el cociente entre la cantidad de casos que la regla difusa predice correctamente

dividida por la cantidad total de casos analizados. La confianza es el cociente entre la cantidad de veces que la respuesta de la regla difusa fue correcta dividida por la cantidad de veces que la regla difusa fue aplicada. En ambos casos, se trata de valores numéricos pertenecientes al intervalo $[0,1]$. Si una regla difusa tiene soporte 0 indica que es incorrecta en todos los casos; es decir, que ningún caso la cumple ya sea porque no verifica el antecedente o el consecuente o ambos. Una regla difusa con confianza 1 ha respondido siempre de manera correcta.

El *factor1* es un valor de multa o penalización en el caso de que el soporte no se encuentre dentro de los rangos establecidos en el algoritmo. El segundo término en la función fitness refleja la importancia que se da al número de atributos que forman el antecedente, siendo *factor2* una constante.

Una vez que se ha obtenido la primera regla difusa, se obtiene el Grado de pertenencia Gp_i del valor de la condición en el conjunto difuso el mismo que está dado por una *t - norma*, que utiliza el operador mínimo, entre los grados de pertenencia de las condiciones difusas implicados en el antecedente de la regla difusa.

Criterio de Votación

Posteriormente se calcula el **CV** como el promedio de los grados de pertenencia Gp de los ejemplos que cumplen la regla tal como se indica en la ecuación 4.14:

$$CV_j = \frac{\sum_{i=1}^n Gp_{ij}}{n} \quad (4.14)$$

Este CV_j es utilizado en el movimiento de la próxima partícula, con lo que la velocidad de *j*-ésima partícula en la parte binaria se actualiza de acuerdo a la ecuación 4.6.

El **CV** utilizado en la selección de las condiciones difusas se toma en consideración para el movimiento del individuo, tomando en cuenta que los individuos van a tener como prioridad el movimiento en la dirección donde se encuentra el valor más alto, es decir aquel valor que tuvo la mayor "*votación*".

Se debe indicar que para ayudar en la simplificación del antecedente cada vez que se evalúa la partícula también se analizan las distintas reglas difusas que se forman al suprimir una a una las condiciones que participan en el antecedente. De todas las reglas difusas 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. 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 $sopBin_i$ 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 *FRvarPSO* de explorar cerca de la solución actualmente propuesta por la partícula.

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

Cuando se ha obtenido una regla difusa aquel conjunto de ejemplos que ha sido cubierto por la regla difusa generada se retira de la base de datos de entrada, el proceso se sigue realizando de forma iterativa hasta que se llegue al máximo número de iteraciones, o hasta que se logren cubrir todos los ejemplos o hasta que la cantidad de ejemplos de cada una de las clases resultantes sean considerados mínimos, lo que pase primero. Si se va a clasificar un ejemplo nuevo, las reglas difusas 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 difusa cuyo antecedente se verifique para el ejemplo que se encuentra analizando.

Es importante indicar que se obtienen aquellas reglas con atributos cualitativos y/o condiciones difusas en el antecedente que tienen un soporte alto. Debido a que se analizan primero las clases que tienen el mayor número de ejemplos, el soporte mínimo para cada una de las clases disminuye conforme se van realizando las respectivas iteraciones, este proceso continúa hasta que se logre cubrir todos los ejemplos, o hasta que se haya llegado al número máximo de iteraciones, o hasta que la cantidad de ejemplos no cubiertos de cada una de las clases sea menor al soporte que se estableció como mínimo.

Para la clasificación de un ejemplo nuevo las reglas difusas deberán aplicarse en el orden que se obtuvieron, es decir aquella primera regla difusa que verifique el antecedente del ejemplo indicará la clase que se tiene en el consecuente, es necesario considerar que las primeras reglas difusas generadas tendrán un mayor soporte.

El Algoritmo 4 muestra el pseudocódigo del método *FRvarPSO* propuesto.

4.2. Resultados obtenidos

Se comparó el desempeño del método propuesto *FRvarPSO* con los métodos C4.5 y *PART* detallados en 1.6.1 utilizando doce bases de datos del repositorio *UCI* (Dua y Karra Taniskidou, 2017), una base de datos de una cooperativa de ahorro y crédito del Sistema Financiero del Ecuador y dos bases de datos de dos Bancos del Ecuador, uno encargado de dar crédito de consumo productivo y no productivo y el otro encargado de micro-crédito. En la tabla 4.2 se indica para cada una de las bases de datos, la cantidad de tuplas, variables difusas, número de conjuntos difusos, y el número de clases en las que los registros pueden ser clasificados.

Algoritmo 4: Algoritmo FRvarPSO

```
begin
  for cada valor nítido (crisp) de la variable do
    Elegir el numero de clusters C ;
    Elegir la norma (utilizar la distancia euclidiana) ;
    Asignar la tolerancia  $\epsilon \leq 0,01$ ;
    Generar la matriz de los grados de pertenencia  $\mu_{ij}$  de forma aleatoria
    while (no se alcance la condición de terminación) do
      Calcular los nuevos centros utilizando la ecuación 4.4
      Calcular las distancias del i-esimo ejemplo al centro ;
      Actualizar la matriz de los grados de pertenencia  $\mu_{ij}$  utilizando la
      ecuación 4.3
    Entrenar la red neuronal competitiva utilizando los ejemplos de
    entrenamiento;
    Determinar el soporte mínimo para cada una de las clases;
    while no se alcance el criterio de terminación do
      Elegir la clase que tenga el mayor numero de ejemplos que no han
      sido cubiertos;
      Construir una población considerando los centroides de la red
      neuronal;
      Evaluar el valor de aptitud (Fitness) de cada partícula utilizando la
      ecuación 4.13 ;
      while la población de partículas no se estabilice do
        for cada partícula do
          Calcular el criterio de votación (grado de pertenencia
          promedio de los ejemplos que cumplen con la regla indicada
          por la partícula) utilizando la ecuación 4.14:
          Calcular la velocidad utilizando la ecuación 4.6
          Obtener la nueva posición de la partícula sumándole la
          velocidad anterior según ecuación 4.9 y acotar según
          corresponda;
        En caso de usar elitismo recuperar la mejor solución de la iteración
        anterior;
      Obtener la mejor regla de la población;
      if la regla cumple con el soporte y la confianza then
        Incorporar la regla difusa al conjunto de reglas difusas;;
        Obtener el valor de salida de la regla difusa que esta dado por el
        grado de pertenencia de los ejemplos que cumplen la regla
        utilizando la respectiva t-norma;
        Retirar del conjunto de entrada los ejemplos cubiertos correctamente;;
        Recalcular el soporte mínimo de la clase que se ha considerado;
  Result: Conjunto de Reglas Difusas formado por las mejores soluciones
  encontradas
```

TABLA 4.2: Descripción de Bases de datos del repositorio UCI y de tres instituciones financieras del Ecuador

Bases de datos	# Registros	# Var. Difusas	# Var. Nominales	# Var. Numéricas	# Clases
Adult Data	30128	6	8	0	2
Balance scale	625	3	0	0	3
Breast w	683	9	0	0	2
Credit a	653	6	9	0	2
Credito coop	22432	8	1	3	2
Credit g	1000	13	1	0	2
Credit banco1	95744	8	1	3	2
Credit banco2	36346	8	1	3	2
Diabetes	768	8	0	0	2
Drugte	400	3	3	0	5
DrugY	200	3	3	0	5
Heart c	296	5	6	0	2
Heart Statlog	270	6	0	7	2
Vinos	178	13	0	0	3
Zoo	101	1	16	0	7

4.2.1. Particionando el rango de cada atributo numérico en intervalos de igual longitud

En este caso se consideraron particiones difusas uniformes, con el mismo número de etiquetas para cada una de las variables analizadas, para esto a cada una de las variables se le determinó los límites superior e inferior, luego se distribuyó uniformemente en nueve intervalos, el primer conjunto difuso formado por los cuatro primeros intervalos, el segundo conjunto difuso desde la mitad del tercer intervalo hasta la mitad del séptimo intervalo, el tercer conjunto difuso está formado desde el sexto al noveno intervalo. De esta forma cada uno de los conjuntos difusos tiene una distribución uniforme, y existe un solapamiento entre las fronteras, representando lo que ocurre en el razonamiento humano. Cada una de las variables numéricas puede tener como máximo dos grados de pertenencia a dos de los conjuntos difusos definidos, en el tercer conjunto difuso su valor será cero. Los conjuntos difusos tomaron los nombres de *Bajo*, *Medio* y *Alto*.

Utilizando una función de pertenencia triangular, se fuzzificaron las variables, luego de lo cual se utilizó la red neuronal SOM y LVQ para inicializar el cúmulo de partículas.

Se realizaron 30 ejecuciones independientes de cada uno de los métodos, para SOM+PSO se utilizó una grilla de 6x5 con 4 vecinas por neurona. Mientras que en el caso de LVQ+PSO la red utilizada está formada por 30 neuronas. Para las versiones que utilizan población variable se empezó con una población menor agregando o quitando partículas de acuerdo a lo que se requiera.

En la tabla 4.3, se encuentran los resultados de la precisión obtenidos al aplicar los métodos C4.5, , **SOM+pso**, **SOM+varPSO**, **LVQ+PSO**, **LVQ+varPSO**. Los mismos que se comparan con **FRvarPSO**. En cada uno de los casos se ha considerado la precisión del conjunto de reglas difusas. Se indica también su respectiva desviación estándar.

TABLA 4.3: Resultados de la Precisión obtenidos al aplicar los métodos a un conjunto de bases del repositorio UCI y de tres instituciones financieras del Ecuador

Data Set	C4.5	PART	SOM + PSO	SOM + var-PSO	LVQ + PSO	LVQ + var-PSO	FRvarPSO
Adult Data	0,8517 ±0,0019	0,8436 ±0,0025	0,8001 ±0,0012	0,7953 ±0,0034	0,8156 ±0,0053	0,8046 ±0,0041	0,8118 ±0,0047
Balance scale	0,7516 ±0,0120	0,8105 ±0,0019	0,7317 ±0,0090	0,7222 ±0,0045	0,7421 ±0,0011	0,7325 ±0,0011	0,7619 ±0,0015
Breast w	0,9556 ±0,0076	0,9525 ±0,0081	0,9454 ±0,0101	0,9526 ±0,0067	0,9457 ±0,0088	0,9512 ±0,0072	0,9565 ±0,0063
Credit a	0,8524 ±0,0036	0,8494 ±0,0027	0,8507 ±0,0186	0,8658 ±0,0080	0,8630 ±0,0077	0,8635 ±0,0088	0,8689 ±0,0122
Credito coop	0,8105 ±0,0011	0,8054 ±0,0023	0,7718 ±0,0019	0,7701 ±0,0023	0,7871 ±0,0031	0,7894 ±0,0023	0,7988 ±0,0129
Credit g	0,7040 ±0,0112	0,7110 ±0,0101	0,6997 ±0,0019	0,6935 ±0,0078	0,6958 ±0,0126	0,6914 ±0,0151	0,7592 ±0,0058
Credit banco1	0,9778 ±0,0003	0,9761 ±0,0007	0,9688 ±0,0033	0,9700 ±0,0026	0,9737 ±0,0037	0,9778 ±0,0018	0,9880 ±0,0026
Credit banco2	0,8492 ±0,0010	0,8413 ±0,0095	0,8133 ±0,0024	0,8264 ±0,0041	0,8356 ±0,0028	0,8415 ±0,0035	0,8501 ±0,0033
Diabetes	0,7414 ±0,0101	0,7434 ±0,0325	0,7314 ±0,0146	0,7207 ±0,0104	0,7362 ±0,0185	0,7213 ±0,0165	0,7442 ±0,0148
Drugte	0,9300 ±0,0301	0,9075 ±0,0112	0,6300 ±0,0010	0,6150 ±0,0106	0,7950 ±0,0035	0,8409 ±0,0336	0,8513 ±0,0195
DrugY	0,9275 ±0,0035	0,9055 ±0,0069	0,8178 ±0,0389	0,7537 ±0,0136	0,7666 ±0,0141	0,8275 ±0,0106	0,8405 ±0,0247
Heart c	0,7601 ±0,0442	0,7939 ±0,0125	0,7277 ±0,0019	0,7617 ±0,0071	0,7576 ±0,0118	0,7614 ±0,0259	0,7866 ±0,0248
Heart Statlog	0,7666 ±0,0274	0,7333 ±0,0276	0,7497 ±0,0028	0,7772 ±0,0145	0,7611 ±0,0131	0,7957 ±0,0219	0,7962 ±0,0236
Vinos	0,8763 ±0,0102	0,8820 ±0,0126	0,8749 ±0,0029	0,8833 ±0,0118	0,8778 ±0,0079	0,8833 ±0,0157	0,8944 ±0,0109
Zoo	0,9207 ±0,0144	0,9227 ±0,0134	0,9300 ±0,0015	0,9500 ±0,0018	0,9432 ±0,0051	0,9526 ±0,0045	0,9601 ±0,0023

Para probar la hipótesis de que los modelos difusos son mejores que aquellos que no utilizan variables difusas, se realizó un test de diferencia de medias. Se verificó que los modelos difusos presentan en la mayor parte de los casos una precisión superior a la del método **PART**, pero ligeramente mayor a la alcanzada por el método C4.5. Sin embargo, si se observa la cantidad de reglas generadas para alcanzar dicha precisión, los métodos difusos utilizan una cantidad promedio de reglas menor que

TABLA 4.4: Resultados de la Precisión obtenidos al aplicar los métodos difusos a un conjunto de bases del repositorio UCI y de tres instituciones financieras del Ecuador

Data Set	# SOM + PSO Difuso	SOM + varPSO Difuso	LVQ + PSO Difuso	FRvarPSO
Adult Data	0,8461 ±0,0046	0,7977 ±0,0660	0,8239 ±0,0900	0,8118 ±0,0047
Balance scale	0,7326 ±0,0201	0,7246 ±0,0216	0,7460 ±0,0192	0,7619 ±0,0015
Breast w	0,9344 ±0,0068	0,9401 ±0,0140	0,9544 ±0,0114	0,9565 ±0,0063
Credit a	0,8550 ±0,0131	0,8657 ±0,0098	0,8578 ±0,0109	0,8689 ±0,0122
Credito coop	0,7857 ±0,0026	0,7891 ±0,0021	0,7922 ±0,0029	0,7988 ±0,0129
Credit g	0,7536 ±0,0106	0,7697 ±0,0081	0,7578 ±0,0091	0,7592 ±0,0058
Credit banco1	0,9819 ±0,0037	0,9840 ±0,0032	0,9869 ±0,0024	0,9880 ±0,0026
Credit banco2	0,8316 ±0,0022	0,8385 ±0,0033	0,8455 ±0,0029	0,8501 ±0,0033
Diabetes	0,7325 ±0,0136	0,7269 ±0,0158	0,7369 ±0,0164	0,7442 ±0,0148
Drugte	0,8480 ±0,0217	0,8435 ±0,0195	0,8501 ±0,0139	0,8513 ±0,0195
DrugY	0,7818 ±0,0342	0,7846 ±0,0275	0,7899 ±0,0267	0,8405 ±0,0247
Heart c	0,7698 ±0,0198	0,7663 ±0,0266	0,7983 ±0,0024	0,7866 ±0,0248
Heart Statlog	0,7664 ±0,0318	0,7886 ±0,0113	0,7666 ±0,0335	0,7962 ±0,0236
Vinos	0,8722 ±0,0309	0,8916 ±0,0235	0,8811 ±0,0071	0,8944 ±0,0109
Zoo	0,9326 ±0,0275	0,9567 ±0,0025	0,9588 ±0,0102	0,9601 ±0,0023

TABLA 4.5: Resultados del número de reglas obtenidos al aplicar los métodos a un conjunto de bases del repositorio UCI y de tres instituciones financieras del Ecuador

Data Set	C4.5	PART	SOM + PSO	SOM + var- PSO	LVQ + PSO	LVQ + var- PSO	FRvarPSO
Adult Data	563,5000 ±3,2510	836,4232 ±7,3026	3,9420 ±0,3536	3,2496 ±0,0707	4,0987 ±0,1414	4,0422 ±0,3536	3,0106 ±0,2267
Balance scale	52,0015 ±1,2478	47,0051 ±1,3102	7,8943 ±0,4243	8,7994 ±0,1414	8,3486 ±0,2121	9,2498 ±0,0707	8,1523 ±0,5794
Breast w	11,0129 ±0,2150	11,1290 ±0,1201	2,7151 ±0,1701	2,6430 ±0,2014	2,7381 ±0,1075	2,6763 ±0,1476	2,3256 ±0,1567
Credit a	17,1100 ±0,1923	37,0012 ±0,1847	3,0099 ±0,0316	3,0099 ±0,0118	3,0000 ±0,0019	3,0000 ±0,0027	3,0000 ±0,0005
Credito coop	114,0000 ±6,0000	42,0000 ±2,0000	6,2966 ±0,2990	5,6899 ±0,2801	5,4772 ±0,1837	5,3535 ±0,2650	5,2990 ±0,1907
Credit g	98,0009 ±0,9868	78,0015 ±0,3214	7,7909 ±0,0407	8,2722 ±0,0999	8,6670 ±0,0622	8,8927 ±0,0903	8,2120 ±0,0564
Credit banco1	154,0000 ±5,0000	81,0000 ±2,0000	7,4939 ±0,2928	7,0936 ±0,2642	7,6993 ±0,2307	7,5894 ±0,1880	6,3972 ±0,1915
Credit banco2	126,0000 ±4,0000	59,0000 ±3,0001	8,4994 ±0,1907	8,2951 ±0,2121	6,4926 ±0,3145	6,2801 ±0,3536	5,9901 ±0,3112
Diabetes	26,0017 ±0,5129	9,0163 ±0,3223	4,3517 ±0,4218	4,4488 ±0,0327	4,2687 ±0,3327	4,3419 ±0,2759	4,1371 ±0,1647
Drugte	17,1001 ±0,0425	14,0204 ±0,0171	7,3498 ±0,0707	7,2481 ±0,2121	8,6081 ±0,0221	7,7459 ±0,3536	7,2002 ±0,3466
DrugY	12,0016 ±0,0039	10,0162 ±0,0401	8,5498 ±0,0707	7,8771 ±0,0348	8,4994 ±0,0914	8,2377 ±0,0636	7,7301 ±0,0168
Heart c	19,0032 ±0,2711	22,0191 ±0,2251	4,7833 ±0,0565	5,0079 ±0,0119	5,2962 ±0,0282	4,7958 ±0,0182	3,5908 ±0,0129
Heart Statlog	18,0093 ±0,2720	24,0152 ±0,2670	4,8836 ±0,0445	4,6957 ±0,0092	5,0906 ±0,0078	4,0496 ±0,0407	3,3468 ±0,0316
Vinos	7,1002 ±0,0105	5,0019 ±0,0726	4,3358 ±0,0750	4,3954 ±0,0282	4,6989 ±0,0141	4,1496 ±0,0707	4,1144 ±0,0481
Zoo	9,0101 ±0,0190	8,0018 ±0,0094	6,9992 ±0,0141	7,1000 ±0,0011	7,0281 ±0,0145	6,9832 ±0,0219	6,5422 ±0,0156

TABLA 4.6: Resultados del número de reglas obtenidos al aplicar los métodos Difusos a un conjunto de bases del repositorio UCI y de tres instituciones financieras del Ecuador

Data Set	# SOM + PSO Difuso	SOM + varPSO Difuso	LVQ + PSO Difuso	FRvarPSO
Adult Data	2,7928 ±0,2828	2,7919 ±0,2828	2,7386 ±0,3536	3,0106 ±0,2267
Balance scale	8,6188 ±0,4596	8,2277 ±0,4451	8,2038 ±0,3348	8,1523 ±0,5794
Breast w	2,6480 ±0,1080	2,5720 ±0,2150	2,4977 ±0,1247	2,3256 ±0,1567
Credit a	3,0083 ±0,0009	3,0000 ±0,0015	3,0000 ±0,0010	3,0000 ±0,0005
Credito coop	5,9874 ±0,2429	5,6920 ±0,2245	5,4958 ±0,2128	5,2990 ±0,1907
Credit g	7,7612 ±0,0540	8,3848 ±0,0141	8,3595 ±0,0608	8,2120 ±0,0564
Credit banco1	6,9930 ±0,2121	6,3888 ±0,2828	6,7588 ±0,2642	6,3972 ±0,1915
Credit banco2	6,4415 ±0,1814	6,3990 ±0,2099	6,6904 ±0,3021	5,9901 ±0,3112
Diabetes	4,2658 ±0,2003	4,2154 ±0,2098	4,2513 ±0,2914	4,1371 ±0,1647
Drugte	8,1167 ±0,2440	7,9860 ±0,2644	7,7944 ±0,3127	7,2002 ±0,3466
DrugY	8,4304 ±0,0319	8,3984 ±0,0487	8,2753 ±0,0497	7,7301 ±0,0168
Heart c	3,7380 ±0,0512	3,8452 ±0,0201	3,6829 ±0,0371	3,5908 ±0,0129
Heart Statlog	3,8806 ±0,0400	3,5804 ±0,0039	3,7929 ±0,0280	3,3468 ±0,0316
Vinos	5,3102 ±0,0810	5,1066 ±0,0221	4,4981 ±0,0106	4,1144 ±0,0481
Zoo	6,9970 ±0,0197	6,9951 ±0,0015	6,8745 ±0,0106	6,5422 ±0,0156

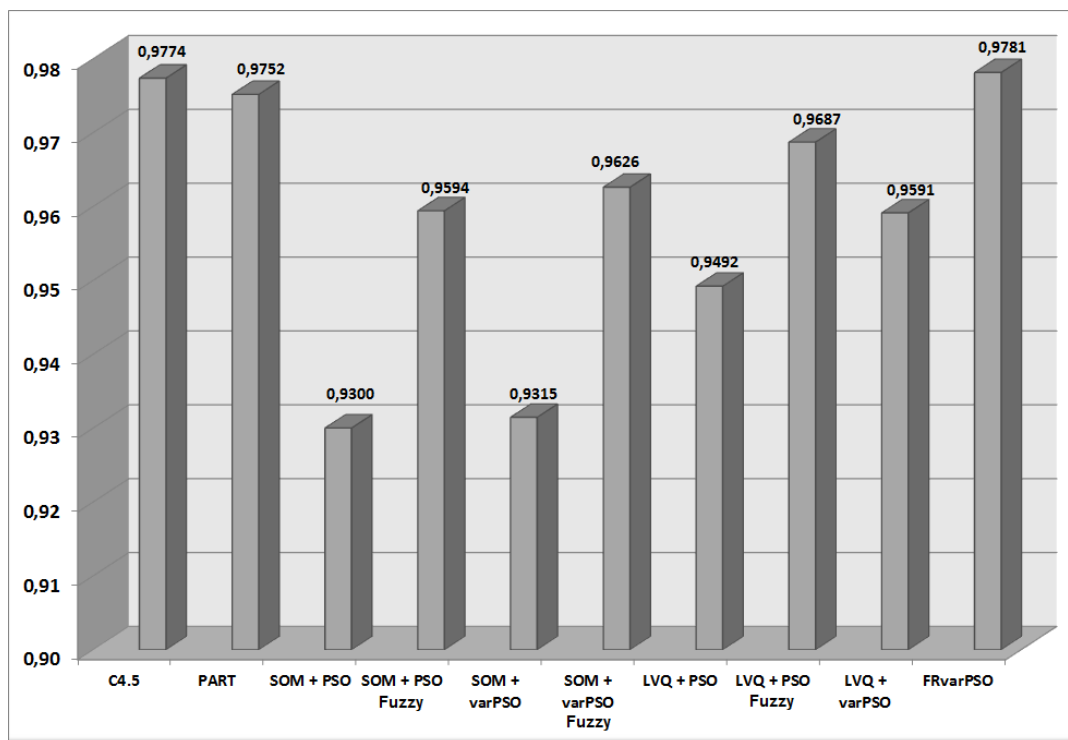


FIGURA 4.1: Comparación de la Precisión de los métodos

las reglas de C4.5 y **PART**. Esto demuestra el éxito del *énfasis puesto en la sencillez del modelo*.

En la figura 4.1 se puede observar la relación que existe de la Precisión de cada uno de los métodos, con la mejor solución encontrada. Para encontrar esta relación se obtuvo el promedio del valor de la precisión obtenido por el método dividido para la mejor precisión, es decir la mayor precisión de la base de datos analizada.

En la figura 4.2 se puede observar la relación que existe de la cardinalidad (número de reglas) de cada uno de los métodos, con la mejor solución encontrada. Para encontrar esta relación se obtuvo el promedio del valor del número de reglas obtenido por el método dividido para la solución que tiene el menor número de reglas de la base de datos analizada.

En la figura 4.3 se puede observar la relación entre la precisión y la cardinalidad de los métodos analizados.

4.2.2. Utilizando Fuzzy-C Means

En la tabla 4.7, se encuentran los resultados de la precisión obtenidos al aplicar cuatro métodos utilizando **FCM** para la obtención de los conjuntos difusos. Se indica también su respectiva desviación estándar.

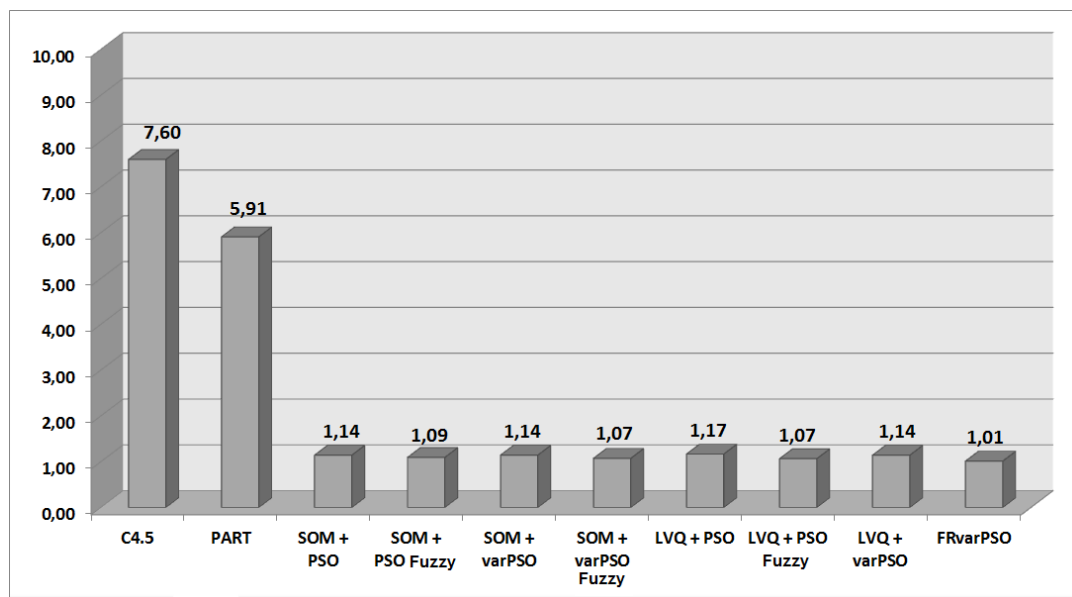


FIGURA 4.2: Comparación de la cardinalidad de los métodos

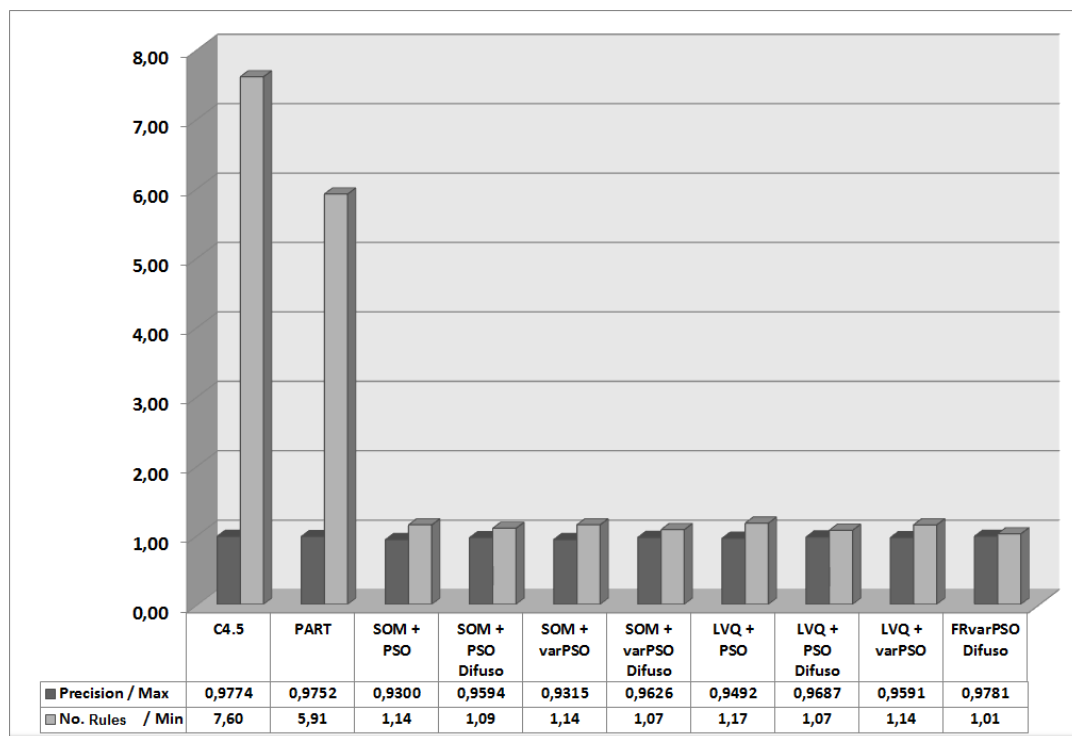


FIGURA 4.3: Comparación de la precisión y Cardinalidad de los métodos

TABLA 4.7: Resultados de la Precisión obtenidos al aplicar los métodos a un conjunto de bases del repositorio UCI y tres instituciones financieras del Ecuador utilizando **FCM**

Data Set	SOM + PSO	SOM + varPSO	LVQ + PSO	FRvarPSO
Adult Data	0,8501 ±0,0020	0,8005 ±0,0799	0,8309 ±0,0271	0,8326 ±0,0066
Balance scale	0,7395 ±0,0125	0,7485 ±0,0152	0,7422 ±0,0243	0,7761 ±0,0020
Breast w	0,9500 ±0,0093	0,9539 ±0,0111	0,9473 ±0,0171	0,9682 ±0,0033
Credit a	0,8538 ±0,0097	0,8838 ±0,0052	0,8608 ±0,0104	0,8738 ±0,0041
Credito coop	0,7820 ±0,0012	0,7835 ±0,0049	0,7890 ±0,0035	0,7901 ±0,0092
Credit g	0,7570 ±0,0190	0,7530 ±0,0214	0,7653 ±0,0017	0,7677 ±0,0326
Credit banco1	0,9559 ±0,0015	0,9802 ±0,0081	0,9811 ±0,0032	0,9872 ±0,0005
Credit banco2	0,8295 ±0,0045	0,8215 ±0,0011	0,8410 ±0,0033	0,8488 ±0,0092
Diabetes	0,7324 ±0,0157	0,7303 ±0,0149	0,7402 ±0,0182	0,7494 ±0,0169
Drugte	0,8465 ±0,0204	0,8471 ±0,0193	0,8575 ±0,0172	0,8595 ±0,0204
Heart c	0,7790 ±0,0206	0,7693 ±0,0295	0,7753 ±0,0195	0,7933 ±0,0329
Heart Statlog	0,7703 ±0,0496	0,7925 ±0,0125	0,7907 ±0,0265	0,8014 ±0,0254
Vinos	0,8666 ±0,0460	0,8933 ±0,0341	0,9005 ±0,0065	0,8999 ±0,0406
Zoo	0,9335 ±0,0210	0,9590 ±0,0077	0,9600 ±0,0099	0,9699 ±0,0045

En la tabla 4.8, se encuentran los resultados del número de reglas obtenidos al aplicar cuatro métodos, utilizando FCM para la obtención de los conjuntos difusos. Se indica también su respectiva desviación estándar.

TABLA 4.8: Resultados del Número de Reglas obtenidos al aplicar los métodos a un conjunto de bases del repositorio UCI y tres instituciones financieras del Ecuador utilizando FCM

Data Set	SOM + PSO	SOM + varPSO	LVQ + PSO	FRvarPSO
Adult Data	2,6543 ±0,2321	2,3933 ±0,027	2,6933 ±0,0090	2,2366 ±0,2580
Balance scale	8,6500 ±0,1895	7,3300 ±0,2141	8,0700 ±0,1659	7,1200 ±0,2017
Breast w	2,6200 ±0,1093	2,3500 ±0,1670	2,6601 ±0,1117	2,4311 ±0,1400
Credit a	2,9009 ±0,0013	2,9800 ±0,0010	2,9810 ±0,0046	2,8055 ±0,0022
Credito coop	6,0100 ±0,1821	5,8918 ±0,1698	5,7600 ±0,1453	5,4081 ±0,1275
Credit g	7,5401 ±0,0115	8,0040 ±0,0172	8,16400 ±0,0172	7,9602 ±0,0206
Credit banco1	7,0113 ±0,3130	6,6220 ±0,2811	6,8421 ±0,3103	6,5901 ±0,1723
Credit banco2	6,4911 ±0,1712	6,5611 ±0,2099	6,8415 ±0,2911	6,3102 ±0,2872
Diabetes	4,7658 ±0,1250	4,1732 ±0,1484	4,2400 ±0,1549	4,1020 ±0,1523
Drugte	8,1050 ±0,1525	7,6000 ±0,1925	7,5300 ±0,1525	7,0501 ±0,1250
Heart c	3,6600 ±0,0417	3,7702 ±0,0115	3,4001 ±0,0221	3,2300 ±0,0132
Heart Statlog	3,9500 ±0,0122	3,600 ±0,0021	3,4289 ±0,0121	3,1295 ±0,0124
Vinos	4,975 ±0,0793	5,0099 ±0,0141	4,2200 ±0,0105	4,0009 ±0,0327
Zoo	6,9700 ±0,0138	6,9000 ±0,0037	6,7002 ±0,0098	6,3107 ±0,0107

4.2.3. Interpretación de las Reglas Difusas

La interpretación de una regla de clasificación difusa involucra evaluar el antecedente mediante la aplicación de un operador difuso, y el segundo paso es la aplicación del resultado del antecedente al consecuente. Para esto se utilizan los operadores de composición de conjuntos difusos y la aplicación de un sistema de inferencia llamado razonamiento difuso o razonamiento aproximado.

Para cada una de las reglas obtenidas, se aplica el operador “and”, que significa intersección en el antecedente de la regla, por lo que es necesario utilizar una t-norma, representando la intersección de los conjuntos difusos. La intersección de dos conjuntos difusos A y B es un conjunto difuso $A \cap B$, cuya función de pertenencia $\mu_{A \cap B}$ se calcula como el mínimo de las respectivas funciones de los conjuntos A y B como se indicó en la ecuación 2.8. Este operador mínimo calcula el mínimo grado de pertenencia del antecedente de la regla, y se calcula de la siguiente manera:

- De los ejemplos que cumplen la regla, obtener el promedio de las funciones de pertenencia de las variables difusas presentes en el antecedente de la regla.
- Obtener el mínimo de las funciones de pertenencia de las variables presentes en el antecedente de la regla.

Es decir para cada regla se observa el grado de cumplimiento de las variables medidas, a las respectivas etiquetas lingüísticas, realizando luego la conjunción de los grados de cumplimiento.

Por ejemplo, si se tiene una regla difusa cuyo antecedente está formado por dos variables, donde cada una de las variables de los ejemplos que cumplen la regla tiene como grado de pertenencia promedio 0.7 y 0.4 respectivamente, al emplear el operador mínimo se obtendrá el valor 0.4. Este valor puede considerarse como el grado del cumplimiento de un ejemplo genérico promedio al antecedente de la regla. En otras palabras, como la regla está formada por condiciones difusas unidas por conjunciones, si se toma el menor de todos los grados de pertenencia del ejemplo a las condiciones que conforman el antecedente de la regla, se puede considerar que esta métrica, de alguna forma, es proporcional al riesgo.

Es decir si la regla difusa es:

Si (Capacidad de Pago = Bajo) y (Morosidad = Media) Entonces (Crédito = N) puede indicarse que *la aplicación de la regla difusa a un ejemplo, representaría un nivel de riesgo para la institución financiera de 0,4*. Este valor es una ponderación clara del sustento ofrecido por los ejemplos o situaciones de crédito históricas que fueron utilizados para su construcción, por lo que se lo considera una métrica directamente proporcional al riesgo de su aplicación.

Por ejemplo en el caso de una institución financiera, si esta tiene liquidez y su objetivo es la colocación, el oficial de crédito al tener un nivel de riesgo en la concesión de 0,4 podrá tomar la decisión y conceder el crédito, incrementando ya sea la tasa de interés o las garantías, entre otros.

4.3. Conclusiones

En este capítulo se ha detallado el método **FRvarPSO** que es el objetivo central de esta tesis. El mismo que es capaz de extraer reglas de clasificación difusas.

El método **FRvarPSO** inicia con la determinación de los conjuntos difusos, y las funciones de pertenencia de cada variable difusa asociada a cada uno de ellos. Para esto se utilizaron dos métodos. En un primer caso se realizó una distribución equitativa para la determinación de los conjuntos difusos, considerando particiones difusas uniformes con el mismo número de etiquetas para cada una de las variables. En el segundo caso se utilizó el algoritmo **FCM** con 3 clusters, donde cada uno de ellos representa un conjunto difuso. Posteriormente se continúa con el agrupamiento de los ejemplos de entrenamiento a través de una red neuronal competitiva **LVQ**. Una vez entrenada la red, se procede a obtener una a una las reglas de clasificación ejecutando reiteradamente la técnica de optimización **varPSO**, la misma que incorpora un **CV** empleado en el movimiento de la partícula.

El método propuesto se ha analizado en doce bases de datos del repositorio **UCI** y en tres bases de datos reales pertenecientes a instituciones financieras del Ecuador. Se obtuvieron resultados satisfactorios. En todos los casos las reglas difusas obtenidas se caracterizan por tener una buena precisión, baja cardinalidad. Gracias a la incorporación de la **LD** se consigue una mejor interpretabilidad del modelo, logrando una mayor precisión sin importar la forma en que se obtuvieron los conjuntos difusos. Por lo indicado anteriormente se cumple el objetivo de esta investigación.

Capítulo 5

Caso de estudio: Riesgo Crediticio

Las técnicas detalladas en los capítulos 3 y 4 pueden emplearse en la solución de problemas reales. Luego es importante considerar lo siguiente:

- Para resolver eficientemente un problema, este debe entenderse bien y en su totalidad.
- Todo problema debe descomponerse en partes.
- Cada una de estas partes puede resolverse utilizando varios métodos. Después de un análisis del problema a resolver se deberá seleccionar el mejor método.
- Una mejor solución se presenta cuando se combinan diversos métodos. Aquí radica una de las ventajas del método **FRvarPSO** propuesto en esta tesis que combina una red neuronal competitiva con una técnica de optimización de población variable cuya estrategia de búsqueda utiliza **LD**. Se consideran varias técnicas para la obtención de los conjuntos difusos entre las que se incluye el conocimiento del experto crediticio.

Con el objetivo de mostrar un ejemplo real del método **FRvarPSO**, en este capítulo se utilizará el método propuesto para obtener las reglas de clasificación difusas correspondientes a crédito.

A continuación se hará una revisión de en qué consiste el Riesgo de Crédito, cuáles fueron las variables analizadas y cuáles las transformaciones que se efectuaron en los datos reales. Luego se detallarán los resultados obtenidos. Para más detalle se puede consultar (Jimbo y col., 2016), (Jimbo, Lanzarini y Fernandez, 2018a), (Jimbo, Lanzarini y Fernandez, 2018b).

5.1. Riesgo Crediticio

La gestión de riesgos es una parte integral de la planificación de las actividades operativas de las más grandes empresas e instituciones financieras a nivel mundial. Esto debido a su importancia y a la necesidad de asegurar la sostenibilidad de sus

actividades. De hecho, el nuevo énfasis en este tema es el resultado de crisis y experiencias recientes, y refleja una mejor comprensión de la importancia de anticipar lo inesperado en lugar de contentarse con reaccionar después del hecho.

Varios tipos de riesgo pueden afectar la supervivencia de una institución financiera. Entre estos riesgos se puede citar el *riesgo de mercado*, *riesgo de crédito*, *riesgo operacional*, *riesgo reputacional*, *riesgo de tipo de interés* entre otros. El riesgo de crédito [un elemento central en la gestión bancaria](#).

El **Riesgo de Crédito** puede ser definido como la posibilidad de pérdida debido al incumplimiento del prestatario o la contra parte en operaciones directas, indirectas o de derivados que conlleva el no pago, el pago parcial o la falta de oportunidad en el pago de las obligaciones pactadas. El incumplimiento en cambio se caracteriza por no efectuar el pago pactado dentro del período determinado, o efectuarlo con posterioridad a la fecha que estaba programado, o en distintas condiciones a las pactadas en el contrato.

Cada institución financiera tiene su propio perfil de riesgo según las características de los mercados en los que opera, y de los productos que ofrece; por lo tanto al no existir un modelo único de administración del riesgo de crédito, cada entidad debe desarrollar su propio esquema.

Lo importante es, que las instituciones financieras cuenten con un proceso formalmente establecido de administración del riesgo de crédito, que asegure la calidad de sus portafolios y además permita identificar, medir, controlar / mitigar y monitorear las exposiciones de riesgo de la contra parte y las pérdidas esperadas, a fin de mantener una adecuada cobertura de provisiones o de patrimonio técnico.

De acuerdo a lo anterior la evaluación del riesgo de crédito, es la evaluación práctica para que las instituciones financieras decidan la capacidad de los prestatarios, a fin de que puedan cumplir sus obligaciones financieras.

La administración del riesgo crediticio durante las últimas décadas ha sido una de las áreas con mayor crecimiento. Existen varios modelos para medir el riesgo de crédito (Saavedra y Saavedra, 2010). Dentro de las técnicas de calificación más utilizadas para la administración del riesgo crediticio se tiene al credit scoring (otorgamiento de crédito) y el behavioral scoring (comportamiento crediticio). Así como varias herramientas para la estimación del riesgo financiero con relación a los préstamos o financiamientos.

En el mercado crediticio ecuatoriano (el caso de estudio considerado en esta tesis), el crédito de consumo tiene la siguiente clasificación: créditos a microempresas y a personas naturales (cuyo destino es consumo y vivienda).

Todos los solicitantes de crédito, así como los clientes consolidados de instituciones financieras cuentan con una calificación en la solicitud del crédito. Adicionalmente se almacena información del comportamiento en la vida del crédito.

Existen dos objetivos importantes en las técnicas para la calificación crediticia. Uno de ellos es la necesidad de identificar el riesgo de consumo, mientras que el otro es minimizar el porcentaje de clientes incumplidos. Con lo cual las instituciones bancarias o crediticias optimizan sus carteras para mejorar su negocio.

Dada la particularidad de su objetivo, el sector de las microfinanzas, constituido por microempresarios que solicitan créditos y en ocasiones no presentan ninguna garantía, es un sector sensible al riesgo sobre todo porque hay otros factores que influyen en la tasa de reembolso del préstamo y como se sabe la información de la que se dispone no es simétrica. Esto lleva a que el tipo de actividades que se financian induzca a las instituciones financieras a evaluar subjetivamente el riesgo que representa el cliente. Lo anterior perjudica la aplicación de la teoría clásica, ya que lo que se requiere es disponer de bases de datos confiables y de calidad que aseguren el poder representar la realidad de los microempresarios.

En el Ecuador, a través de la Superintendencia de bancos y seguros (Seguros, 2017), se establece que las instituciones financieras serán las responsables de determinar las metodologías y procesos para identificar, medir, controlar y monitorear el riesgo de crédito. En esta resolución también se indica que las instituciones controladas, deberán contar con un sistema para monitorear los niveles del riesgo de crédito en forma permanente a través de las diferentes metodologías adoptadas por cada entidad. Para cada modalidad de crédito (comercial, consumo, vivienda y microcrédito). Dentro de las cuales se determinarán los principios y criterios generales para la evaluación del riesgo de crédito.

Las metodologías implantadas deben considerar la combinación de criterios cuantitativos y cualitativos, de acuerdo con la experiencia y las políticas estratégicas de la entidad. Esta metodología debe garantizar su idoneidad, al igual que la relevancia de las variables utilizadas.

Por lo anterior, se requiere desarrollar nuevos modelos con base en la información histórica de los clientes, así como incorporar información interna y externa que permita generar modelos de decisión en el otorgamiento y considerar el comportamiento con los clientes consolidados.

La misma resolución manifiesta que las metodologías y técnicas analíticas basadas en el comportamiento histórico de las operaciones de crédito, van a permitir determinar la pérdida esperada sobre la base de la probabilidad de incumplimiento, el nivel de exposición y la severidad de la pérdida. Todo esto lleva a la disminución del riesgo financiero.

Es importante considerar que las instituciones financieras tratan de incrementar el volumen de operaciones concedidas, sin ver afectado su indicador de riesgo. Otra de las consideraciones que se debe tener en cuenta es que para atraer más clientes,



FIGURA 5.1: Proceso de credit scoring

el proceso de concesión debe ser rápido. Como resultado, las instituciones financieras deben mejorar la precisión del crédito aprobado para evitar el riesgo de incumplimiento, y disminuir el tiempo en la concesión del mismo lo que les lleva a una ventaja competitiva.

Como se dijo anteriormente, uno de los métodos utilizados para gestionar el riesgo es el **credit scoring**. Los modelos de "credit scoring" son métodos matemáticos, estadísticos o econométricos empleados para medir el riesgo y/o la probabilidad del incumplimiento del sujeto de crédito. Generalmente, la evaluación del riesgo de crédito se asocia con varias características del prestatario, como la reputación, el apalancamiento, la volatilidad de los ingresos (la misma que va a depender en parte de la economía del país), el destino del crédito y las garantías, entre otras.

Las técnicas tradicionales de credit scoring permiten realizar una evaluación de los sujetos de crédito basados en: características del propio sujeto, capacidad de pago, capital disponible como respaldo, cualquier activo (por ejemplo vivienda) como garantía del préstamo, condiciones de la economía en general.

El credit scoring comenzó a ganar popularidad en los años noventa. Por su intermedio, la valoración que se realiza del sujeto de crédito se hace a base de varios factores utilizando medidas cuantitativas de desempeño, y características de préstamos anteriores para predecir el futuro de créditos que se lleguen a conceder con características similares. La puntuación de crédito no aprueba ni rechaza una solicitud de préstamo, pero puede predecir la probabilidad de ocurrencia de bajo rendimiento. La técnica de credit scoring se ilustra en la Figura 5.1

En el Ecuador, el sector de las microfinanzas experimentó una crisis de sobreendeudamiento en la década de 1990. La no realización de una apropiada regulación ni supervisión al sistema financiero fue una de las causas para que se adoptara el dólar como medio de pago legal y obligatorio a partir del 9 de enero del 2000. Esto dio lugar a que los dueños de los bancos soliciten créditos vinculados dando como garantía empresas que sólo existían en el papel. Lo anterior fue una práctica común en muchos bancos privados. Las instituciones encargadas de ejercer el control no se dieron por enteradas o en todo caso no hicieron nada por evitarlo. Esto motivó que muchos bancos cerraran, provocando desconfianza en el sistema, masivos retiros de dinero y por consiguiente la quiebra de muchos bancos, como en un efecto dominó.

Posteriormente a la dolarización los clientes a quienes se les había concedido préstamos pasaron de tener sus deudas de sucres a dólares, y aquellos sujetos de

crédito que habían tenido un comportamiento excelente de repente cayeron en morosidad, incrementando aún más el riesgo financiero. Es aquí donde radica la importancia de considerar en el análisis para la concesión de crédito las variables macroeconómicas.

En la literatura se puede encontrar que existen dos enfoques claramente definidos en la evaluación del riesgo crediticio, aquellos que se basan en la ponderación de variables del sujeto de crédito (cuantitativas y cualitativas), y aquellos que están basados en conceptos de tipo fundamental proyectando variables económicas y financieras en el tiempo, y su impacto en los estados financieros de la empresa, hasta el punto de determinar las condiciones en las que estará en quiebra contablemente (Elizondo y Altman, 2004).

Existen varios modelos basados en ponderaciones de factores de riesgo, que se basan en técnicas de inteligencia artificial como los sistemas expertos, redes neuronales (Zhao y col., 2015), (Bekhet y Eletter, 2014), máquinas de vectores de soporte (Li y col., 2017), algoritmos genéticos entre otros. En los estudios como el de (Lessmann y col., 2015) se demuestra que la combinación de modelos presenta mejores resultados, que utilizar una sola técnica para la clasificación de riesgo crediticio. Así se tiene el trabajo de (Oreski y Oreski, 2014), que predice la precisión en la clasificación de crédito utilizando la combinación de un algoritmo genético con una red neuronal.

Cualquier técnica que se utilice debe decidir qué características se usarán para su análisis. Dentro de estas se encuentran las variables microeconómicas y las macroeconómicas.

Variables microeconómicas: son aquellos elementos, características o atributos que describen el comportamiento de una persona como por ejemplo el estado civil, su instrucción, ingresos, salario, egresos, activos, pasivos, entre otras.

Variables macroeconómicas: son aquellas que hacen referencia a la economía como un todo, dentro de estas se puede citar:

- **Inflación:** es el crecimiento continuo y generalizado de los precios de los bienes, servicios y factores productivos de una economía a lo largo del tiempo.
- **Producto bruto interno:** es el valor monetario de los bienes y servicios finales producidos por una economía en un período determinado.
- **Tasa de desempleo:** es una medida de la extensión del desempleo y se calcula como un porcentaje dividiendo el número de personas desempleadas por todas las personas que se encuentran en la fuerza laboral.
- **Tasa de cambio o Tipo de cambio:** mide la cantidad de dinero que se debe pagar por una unidad de moneda extranjera. Es decir qué cantidad de moneda

X se puede adquirir con una moneda Y , entendiéndose como la relación cambiaria que existe entre las monedas de dos países; generalmente se toma como referencia el dólar o el euro.

- **Tasa de interés:** hace referencia al porcentaje al que se invierte un capital en un determinado periodo de tiempo. Puede considerarse como el precio que tiene el dinero que se abona o se percibe para pedirlo o cederlo en préstamo en un momento en particular.

Existen otra serie de factores que influyen en la concesión del crédito como la asimetría de la información, el tipo de actividades que se van a financiar y la forma del préstamo que puede ser individual o grupal.

5.1.1. Técnicas Informáticas utilizadas en Riesgo Crediticio

Las investigaciones relacionadas con el uso de técnicas informáticas aplicables al análisis de riesgo crediticio, se han desarrollado considerando dos problemas. El primero por naturaleza hace referencia a los datos desbalanceados, ya que la clase minoritaria siempre es la de los créditos que no debían concederse o de los clientes que son malos pagadores, que en el mejor de los casos representa el 20 %. El segundo problema es la determinación o no de la concesión del crédito en la que se deben analizar las características del cliente. Este es el tema en el cual se centra este trabajo.

Para la evaluación del crédito se pueden citar varias técnicas informáticas entre estas se destacan los métodos estadísticos que utilizan técnicas como discriminante lineal (Abdou y Pointon, 2011), (Mahmoudi y Duman, 2015) y regresión logística (Daqi y col., 2016), (Dong, Lai y Yen, 2010), (Nie y col., 2011). Estos métodos tienen suposiciones ideales, ya que se consideran que las variables tienen distribución normal multivariada, lo que limita el rendimiento de los modelos.

En el área de la minería de datos, se presenta un conjunto de modelos que se han aplicado en el área de credit scoring, con el objetivo de caracterizar a los sujetos de crédito y de ayudar a decidir si el crédito debe ser otorgado o no. Pueden encontrarse en la literatura numerosos casos exitosos donde estos modelos constituyen una alternativa superando a la valoración subjetiva que habitualmente se aplica a la hora de decidir el otorgamiento del crédito (Thomas, 2000).

Existen también varios modelos que vinculan la morosidad a las características del sujeto de crédito (Reinke, 1998), (Zeller, 1998). Sin embargo estos modelos presentan varios problemas, entre los que pueden mencionarse que no son robustos porque están basados en muestras pequeñas y usan variables que no siempre se poseen.

Algunos de estos aspectos han sido resueltos utilizando técnicas de Machine Learning. Tal es el caso de (Yingjie y col., 2015) que utilizó una red neuronal competitiva SOM para agrupar clientes según sus características más representativas. En dicho trabajo se dividió a los clientes en varios grupos y se analizaron las características de los clientes en cada uno de ellos. Se obtuvieron conclusiones interesantes entre las que se destacó el hecho de que el grupo de mayor riesgo estaba formado por clientes con el ingreso promedio más alto. Una de las razones es que los clientes de este grupo obtienen fácilmente créditos debido a sus altos ingresos. Luego el problema que se presenta es que no se realiza un buen análisis de las variables microeconómicas, ni se consideran las variables macroeconómicas.

En los últimos años los investigadores han presentado otros métodos para mejorar el rendimiento de los modelos de clasificación crediticia. (Shen y col., 2018) propone utilizar SVM semi-supervisada para mejorar el rendimiento de los modelos de crédito, el rendimiento del modelo propuesto es superior a las medidas de referencia. (Zhao y col., 2015) también propone una mejora a un modelo que utiliza una red neuronal Perceptron multicapa entrenada con el algoritmo back propagation para mejorar el credit scoring. Aquí el problema se presenta cuando las clases se encuentran desbalanceadas.

En (Lessmann y col., 2015; Wang y Ma, 2012) propone una nueva técnica basada en SVM utilizando dos estrategias diferentes: la primera selecciona los atributos de arranque y la segunda los toma de forma aleatoria. Si bien el rendimiento de este modelo fue superior a otros modelos de clasificación, sólo utilizan variables microeconómicas para su análisis. (Oreski y Oreski, 2014) utilizan un algoritmo híbrido basado en AG y redes neuronales para predecir la precisión de la clasificación en la concesión de crédito. (Zięba, Tomczak y Tomczak, 2016) utiliza Extreme Gradient Boosting para el aprendizaje de un conjunto de árboles de decisión utilizados para la predicción de pérdida financiera.

El análisis de riesgos es una de las áreas en las que desde sus inicios se reconoció la utilidad de la LD. Los modelos de inferencia difusos en el estudio de crédito fueron propuestos por (Medina Hurtado, 2008). Estos modelos presentan viabilidad para minimizar el riesgo de la institución financiera en los procesos de otorgamiento de créditos.

En trabajos como el de (González, 2012) se pudo medir el riesgo crediticio mediante indicadores aplicados al cliente como son la capacidad de pago, la capacidad de endeudamiento y la calificación crediticia. Estos indicadores alimentan el sistema difuso como herramienta de evaluación, entregando respuestas consistente en el monto y el plazo de concesión. Aquí también pueden encontrarse trabajos como el propuesto por (Córdova, Molina y Navarrete, 2017), donde se pueden observar los resultados de ratios financieros de cooperativas, validando el nivel de pertenencia que tienen los indicadores analizados al compararlo con la meta estandarizada del modelo camel y su calificación de riesgo.

Otro trabajo es el de (Teles y col., 2019) que utiliza la LD y árboles de decisión. Aquí se trata de predecir el valor que se recuperará del crédito. El problema radica en que los datos utilizados para el análisis son de 1980, lo que limita las variables utilizadas.

Entre los modelos de credit scoring que emplean reglas se encuentran trabajos como el de (Bennouna y Tkiouat, 2018), que utiliza un enfoque de lógica difusa permitiendo un tratamiento individual y dinámico de los clientes mediante la integración de la opinión de los expertos.

Otro de los trabajos que propone un modelo de lógica difusa basado en reglas para la evaluación de créditos es el de (Mammadli, 2016), el mismo que se encarga de evaluar préstamos minoristas. Se analizan cinco variables microeconómicas, y a cada una de ellas se le asignan tres términos lingüísticos como "bajo", "medio", "alto". Sin embargo la metodología utilizada determina un conjunto de 729 reglas, las mismas que resultan excesivas el momento del análisis por el oficial de crédito.

En trabajos como el de (Soui y col., 2019) generan un modelo basado en reglas de clasificación, donde consideran a la evaluación del riesgo de crédito como un problema de optimización basado en búsqueda. Aquí el objetivo es minimizar la complejidad de la solución generada, maximizar la precisión y también maximizar el valor que representa la importancia de las reglas. Este trabajo incorpora LD mejorando la comprensión de las reglas. En él se comparan varias técnicas de optimización obteniendo mejor resultado con aquella que utiliza una variante de PSO. En dicho trabajo, las únicas bases de datos que se utilizan son la alemana y la australiana del repositorio UCI.

El objetivo de esta investigación es utilizar un enfoque de LD para evaluar el comportamiento del cliente, con el fin de reducir el riesgo en los préstamos que se van a conceder, con lo que se puede garantizar la viabilidad y la sostenibilidad de las instituciones microfinancieras.

En (Jimbo, Lanzarini y Fernandez, 2020) se hace un análisis de las distintas variantes de cúmulo de partículas en la obtención de las reglas de clasificación aplicables a riesgo crediticio, dentro de estas se hace un análisis del método FRvarPSO, propuesto en esta tesis, que se caracteriza por la obtención de un conjunto reducido de reglas de clasificación difusas, de cardinalidad baja y precisión mayor que el método propuesto en (Jimbo, Lanzarini y Fernandez, 2018a) donde no se utilizó lógica difusa. Además, al igual que en (Lanzarini y col., 2015b; Jimbo, Lanzarini y Fernandez, 2018b), se incorporó a la información del sujeto de crédito la situación económica del país, y se realizó un análisis de la morosidad. La diferencia con el trabajo anterior es que en esta investigación para la obtención de las reglas difusas, no se necesitó del conocimiento de un experto, lo que lo convierte en una gran ventaja.

5.2. **FRvarPSO** aplicado a Riesgo Crediticio

Para las variables difusas los rangos utilizados se normalizaron linealmente en el intervalo $[0, 1]$. Para la determinación de la función de pertenencia de cada uno de los conjuntos difusos se utilizaron tres métodos: (ver su detalle en la sección 4.1.1.

- Particionando el rango de cada atributo numérico.
- Utilizando **FCM**.
- Considerando el criterio de un experto crediticio

Finalmente la variable de salida de las reglas de clasificación indicará si el crédito es o no otorgado. La técnica utilizada **FRvarPSO** da como resultado una lista de reglas de clasificación difusas. Estas reglas deben ser evaluadas por el oficial de crédito en el orden que fueron obtenidas.

FRvarPSO puede ser utilizado para **Predicción y análisis financiero**. En esta investigación se realizaron pruebas en el área de *Concesión de créditos*, donde se evaluaron los riesgos que corre la entidad financiera al conceder un crédito a un cliente. Uno de los puntos principales a tomar en cuenta es el hecho de que si luego de aplicar la regla difusa el resultado es que el crédito debe ser negado, y la entidad financiera tiene liquidez, en este caso puede ser primordial para la organización la colocación.

Es aquí donde el oficial de crédito debe tomar en cuenta el otorgar el crédito analizando el riesgo que representa la concesión (dado en la evaluación de la regla difusa), y disminuir dicho riesgo incrementando por ejemplo la información colateral (activos), la tasa de interés, los garantes entre otros.

5.2.1. **Datos de crédito de Instituciones Financieras**

En el Capítulo 1 en la sección 1.1 se describió con detalle el proceso de extracción del conocimiento. Una de las etapas es la recopilación y preparación de la información con la cual se va a trabajar. Los datos utilizados para realizar las pruebas del modelo **FRvarPSO** son datos de tres instituciones del sistema financiero Ecuatoriano. Estos datos presentan información de los clientes, así como el historial de los créditos otorgados. Se tomaron en cuenta en el análisis variables microeconómicas y macroeconómicas.

En el caso de las variables microeconómicas se puede identificar el comportamiento del cliente a través de tres variables:

- **Descriptivas:** Definen las características generales del cliente
- **Comportamiento de créditos anteriores:** Indican historial crediticio del cliente

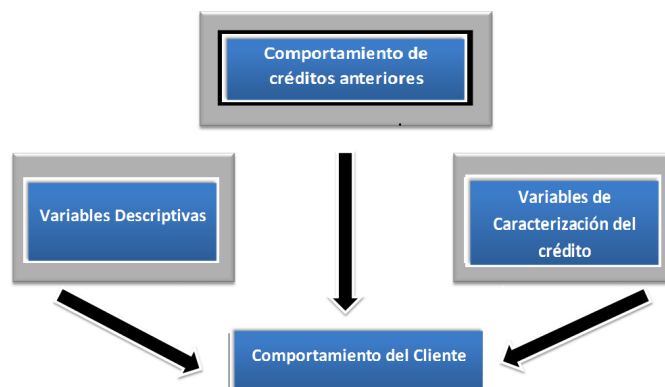


FIGURA 5.2: Proceso de credit scoring

- **Caracterización del préstamo:** Se encargan de las particularidades del préstamo que se va a conceder

En la la Figura 5.2 se indican los variables de entrada para el modelo **FRvarPSO**.

Se analizaron datos de las siguientes entidades financieras:

- **Cooperativa de Ahorro y Crédito** del Ecuador que se encuentra en el segmento 2 dentro de la Superintendencia de Economía Popular y Solidaria, activos mayor a 20'000.000,00 hasta 80'000.000,00
- **Banco del Ecuador encargado de Microcrédito** especializado en colocación masiva de crédito
- **Banco del Ecuador encargado de crédito de consumo y crédito productivo** o empresarial

La Cooperativa de ahorro y crédito es una entidad controlada por la Superintendencia de Economía popular y solidaria, y los Bancos por la Superintendencia de Bancos. La información que se analizó de las tres instituciones financieras corresponden a operaciones crediticias concedidas desde enero de 2012 hasta diciembre de 2018.

Se analizaron tres estructuras de datos:

R1. Sujetos de crédito

Esta estructura tiene periodicidad mensual y comprende información de los titulares, garantes y codeudores de las operaciones de crédito y contingentes, así como de los tarjetahabientes que deben ser reportados en las diferentes estructuras de datos del mes correspondiente.

Su estructura está dada en la tabla 5.1:

TABLA 5.1: R1. Sujetos de crédito

Nro.	Variable	Tipo de Dato	Descripción
1	Tipo de identificación	Caracter (1)	Se refiere al tipo de documento de identificación del sujeto, puede ser "C", "R" ó "E"
2	Identificación del sujeto	Caracter (13)	Corresponde al número de identificación del sujeto. Para personas naturales ecuatorianas será el número de cédula, para personas jurídicas el número de RUC y para personas extranjeras el código de extranjero asignado por la Superintendencia de Bancos.
3	Clase del sujeto	Caracter (1)	Código que indica si el sujeto de riesgo es persona N (natural) o J (jurídica).
4	Actividad económica	Caracter (3)	Código del sector económico o actividad económica a la que se dedica el sujeto de riesgo.
5	Patrimonio del sujeto	Numérico (15,2)	Valor del patrimonio que tiene el sujeto de riesgo. Se debe actualizar al menos en forma semestral. Si el deudor es una entidad controlada por la Superintendencia de Bancos, esta actualización debe ser al menos trimestral. En caso de que el deudor no posea bienes, se tomará como referencia el valor de sus ingresos
6	Fecha de actualización del patrimonio	Fecha (dd/mm/aaaa)	Fecha a la que se encuentra actualizado el valor del patrimonio del sujeto
7	Causal de vinculación	Caracter (4)	Determina la causa de vinculación del sujeto de riesgo con la entidad reportante. En el caso de que el sujeto sea vinculado por propiedad y gestión, primará la causal por propiedad
8	Provincia	Caracter (2)	Código de la provincia en el Ecuador donde reside el sujeto.
Sigue ...			

TABLA 5.1: R1. Sujetos de crédito (continuación)

Nro.	Variable	Tipo de Dato	Descripción
9	Cantón	Caracter (2)	Código del cantón el Ecuador donde reside el sujeto.
10	Parroquia	Caracter (2)	Código de la parroquia en el Ecuador donde reside el sujeto.
11	Género	Caracter (1)	Código del género al que pertenece el sujeto, masculino o femenino.
12	Estado civil	Caracter (1)	Código del estado civil al que pertenece el sujeto.
13	Nivel de estudios	Caracter (1)	Es el código que representa al nivel de educación que tiene el sujeto.
14	Profesión	Caracter (1)	Es el código que representa al título profesional que tiene el sujeto.
15	Tipo de vivienda	Caracter (1)	Código que indica el tipo de vivienda en la que habita el sujeto.
16	Valor de la vivienda	Numérico (15,2)	Es el monto o valor de mercado de la vivienda en la que habita el sujeto.
17	Tiempo residencia vivienda actual	Numérico (2)	Es el tiempo expresado en años en los que el sujeto vive o habita en la última vivienda.
18	Relación de dependencia laboral	Caracter (1)	Es el código que indica el tipo de relación de dependencia laboral actual que tiene el sujeto
19	Tiempo último trabajo	Numérico (3)	Corresponde al número de meses que el sujeto tiene laborando en su actual trabajo.
20	Fecha inicio trabajo anterior	Fecha (dd/mm/aaaa)	Es la fecha en la cual el sujeto ingresó a laborar en su trabajo anterior.
21	Fecha salida trabajo anterior	Fecha (dd/mm/aaaa)	Es la fecha en la cual el sujeto dejó de laborar en su trabajo anterior.
22	Cargas familiares	Numérico (2)	Es el número de cargas familiares que tiene el sujeto.
23	Origen de ingresos	Caracter (1)	Es el código que indica de donde provienen principalmente los ingresos del sujeto.
Sigue ...			

TABLA 5.1: R1. Sujetos de crédito (continuación)

Nro.	Variable	Tipo de Dato	Descripción
24	Tiempo cliente Institución Financiera	Numérico (3)	Es el tiempo efectivo expresado en meses durante el cual el cliente ha tenido una relación transaccional con la institución.
Fin de R1. Sujetos de crédito			

R2. Operaciones concedidas

En esta estructura de periodicidad mensual se encuentran las operaciones de crédito y contingentes que se hayan concedido, renovado, refinanciado o reestructurado en el mes que corresponde a la fecha en la que se encuentran los datos. Su estructura está dada en la tabla 5.2.

TABLA 5.2: R2. Operaciones concedidas

Nro.	Variable	Tipo de Dato	Descripción
1	Tipo de identificación	Caracter (1)	Se refiere al tipo de documento de identificación del sujeto, puede ser "C", "R" ó "E"
2	Identificación del sujeto	Caracter (13)	Corresponde al número de identificación del sujeto. Para personas naturales ecuatorianas será el número de cédula, para personas jurídicas el número de RUC y para personas extranjeras el código de extranjero asignado por la Superintendencia de Bancos.
3	Número de operación	Caracter (32)	Código único con el que la entidad financiera identifica a cada operación que efectúa.
4	Valor de la operación	Numérico (15,2)	Valor por el que se suscribió la operación concedida, que se registra en la cuenta contable respectiva. Dependiendo de la cuenta contable, puede incluir capital más intereses, aunque normalmente solo debe ser capital.
Sigue ...			

TABLA 5.2: R2. Operaciones concedidas (continuación)

Nro.	Variable	Tipo de Dato	Descripción
5	Tasa de interés nominal	Numérico (4,2)	Es la tasa de interés pactada en la concesión de la operación.
6	Tea	Numérico (4,2)	Es el valor de la tasa efectiva anual expresada en porcentaje, debiendo considerar todos los factores de afectación que determinen el valor del préstamo para el deudor.
7	Moneda	Caracter (3)	Código de la moneda en la que se realizó la operación.
8	Fecha de concesión	Fecha (dd/mm/aaaa)	Fecha en que fue instrumentada la operación.
9	Fecha de vencimiento	Fecha (dd/mm/aaaa)	Fecha en que vencerá la operación
10	Línea de crédito	Caracter (1)	Código que indica si la operación se concedió con fondos propios, con redescuento en banca de segundo piso o con recursos internacionales..
11	Periodicidad de pago	Caracter (2)	Es el código que indica la periodicidad con la que el sujeto deberá realizar los pagos de los dividendos del crédito.
12	Frecuencia de revisión	Numérico (4)	Indica la periodicidad o frecuencia con la cual se modifican las tasas de interés. Debe estar expresado en días.
13	Oficina de concesión	Numérico (4)	Código de la oficina (matriz, sucursal o agencia) de la institución financiera en que se instrumentó la operación.
14	Garantes o garantías	Caracter (2)	Indica si la operación cuenta o no, con garantes y/o garantías reales.
15	Indicador de operación exenta	Caracter (1)	Código que indica si la operación concedida está exenta o no del cálculo de límites de crédito.
Sigue . . .			

TABLA 5.2: R2. Operaciones concedidas (continuación)

Nro.	Variable	Tipo de Dato	Descripción
16	Tipo de crédito	Caracter (2)	Código que identifica al tipo de crédito otorgado por la entidad.
17	Total ingresos sujeto	Numérico (15,2)	Para las personas naturales, registrar el monto total de los ingresos mensuales familiares que tiene el sujeto de crédito al momento en que se le concede la operación, y que han sido verificados y aceptadas por la entidad como reales.
18	Total egresos sujeto	Numérico (15,2)	Para las personas naturales, corresponde al monto total de los egresos mensuales familiares que tiene el sujeto de crédito al momento en que se le concede la operación, y que han sido verificados y aceptadas por la entidad como reales, sin considerar las cuotas provenientes de deudas en el sistema financiero y en sector real. Para el caso de microempresarios, corresponde a los gastos efectuados para el sustento familiar.
19	Producción actual que se mantiene	Numérico (15,2)	Es la cantidad de producción para consumo interno (volumen de unidades producidas) expresado en unidades monetarias (dólares) que declara el solicitante. Siempre que el campo tipo de crédito corresponda a los códigos de crédito productivo, comercial y microcrédito.
Sigue ...			

TABLA 5.2: R2. Operaciones concedidas (continuación)

Nro.	Variable	Tipo de Dato	Descripción
20	Destino de crédito para bienes inmuebles	Caracter (2)	Este código será asignado para describir el uso de los valores producto del crédito recibido, cuando el tipo de crédito corresponda a crédito inmobiliario o vivienda de interés público, caso contrario será nulo.
21	Valor comercial del inmueble	Numérico (15.2)	Corresponde al valor comercial, definido por el perito en el avalúo total del inmueble, en el proceso de adquisición o construcción de la vivienda o inmueble. Se registrará valor cuando el campo tipo de crédito corresponda a crédito inmobiliario y vivienda de interés público, en cualquier otro caso registrará cero.
22	Número de personas que aportan al total de ingresos	Numérico (2)	Número de personas que aportan al total de ingresos.- Se indicará el número de personas que aportan al total de ingresos reportado en el campo "Total ingresos sujeto", cuyo monto fue analizado para la concesión del crédito. Cuando el ingreso corresponda únicamente al deudor se reportará el número "1".
Fin de R2. Operaciones concedidas			

R3. Saldos de Operaciones

Esta estructura tiene periodicidad mensual y comprende el detalle de los saldos de operaciones de crédito y contingentes que aún se encuentra activas, indica también las operaciones que se encuentran con morosidad, detallando el número de días. Su estructura se encuentra detallada en la tabla 5.3.

TABLA 5.3: R3. Saldo de operaciones

Nro.	Variable	Tipo de Dato	Descripción
1	Tipo de identificación	Caracter (1)	Se refiere al tipo de documento de identificación del sujeto, puede ser "C", "R" ó "E"
2	Identificación del sujeto	Caracter (13)	Corresponde al número de identificación del sujeto. Para personas naturales ecuatorianas será el número de cédula, para personas jurídicas el número de RUC y para personas extranjeras el código de extranjero asignado por la Superintendencia de Bancos.
3	Número de operación	Caracter (32)	Código único con el que la entidad financiera identifica a cada operación que efectúa.
4	Días de morosidad	Numérico (5)	Indica el número de días que la operación se encuentra en mora. (número de días impagos).
5	Metodología de calificación	Caracter (1)	Es el código que indica el tipo de metodología que la institución financiera está aplicando para calificar a la operación.
6	Valor por vencer de 1 a 30 días	Numérico (15,2)	Saldo de la operación que está por vencer de 1 a 30 días.
7	Valor por vencer de 31 a 90 días	Numérico (15,2)	Saldo de la operación que está por vencer de 31 a 90 días.
8	Valor por vencer de 91 a 180 días	Numérico (15,2)	Saldo de la operación que está por vencer de 91 a 180 días.
9	Valor por vencer de 181 a 360 días	Numérico (15,2)	Saldo de la operación que está por vencer de 181 a 360 días.
10	Valor por vencer de más de 360 días	Numérico (15,2)	Saldo de la operación que está por vencer a más de 360 días.
11	Valor que no devenga intereses de 1 a 30 días	Numérico (15,2)	Es el saldo de la operación que no devenga intereses de 1 a 30 días.
Sigue ...			

TABLA 5.3: R3. Saldo de operaciones (continuación)

Nro.	Variable	Tipo de Dato	Descripción
12	Valor que no devenga intereses de 31 a 90 días	Numérico (15,2)	Es el saldo de la operación que no devenga intereses de 31 a 90 días
13	Valor que no devenga intereses de 181 a 360 días	Numérico (15,2)	Es el saldo de la operación que no devenga intereses de 181 a 360 días.
14	Valor que no devenga intereses de más de 360 días	Numérico (15,2)	Es el saldo de la operación que no devenga intereses de más de 360 días.
15	Valor vencido de 1 a 30 días	Numérico (15,2)	Es el saldo de la operación que se encuentra vencido de 1 a 30 días.
16	Valor vencido de 31 a 90 días	Numérico (15,2)	Es el saldo de la operación que se encuentra vencido de 31 a 30 días.
17	Valor vencido de 91 a 180 días	Numérico (15,2)	Es el saldo de la operación que se encuentra vencido de 1 a 30 días.
18	Valor vencido de 181 a 360 días	Numérico (15,2)	Es el saldo de la operación que se encuentra vencido de 1 a 30 días.
19	Valor vencido de más de 360 días	Numérico (15,2)	Es el saldo de la operación que se encuentra vencido de más de 360 días.
Fin de R3. Saldo de operaciones			

Como se puede observar en las tablas detalladas anteriormente, se dispone de un gran número de variables. Fue necesario realizar un análisis de aquellas variables que son más representativas en riesgo crediticio, por lo que se realizaron las siguientes tareas:

- Eliminar todas aquellas variables que tenían pocos datos.
- Eliminar variables que tenían información duplicada.
- Uso del coeficiente de correlación entre los distintos pares de atributos para eliminar las variables poco representativas.
- Transformación de variables.
- Adicionalmente se unificaron varios campos como por ejemplo sumar varios campos para tener los ingresos, egresos y deudas del sujeto de crédito.

Las variables microeconómicas que se consideraron luego de la transformación se indican en la tabla 5.4:

TABLA 5.4: Variables microeconómicas de análisis

Nro.	Variable	Tipo de Dato	Descripción
1	Año	Numérico (4)	Año en el que se concedió el crédito
2	Mes	Numérico (2)	Mes en el que se concedió el crédito
3	Provincia	Numérico (4,2)	Provincia en la que se concedió el crédito, esta relacionada con la agencia
4	Destino del crédito	Caracter (2)	Indica el destino del crédito otorgado por la entidad
5	Días de morosidad	Numérico (5)	Indica el número de días que la operación se encuentra en mora. (número de días impagos).
6	Valor de la operación	Numérico (15,2)	Monto del crédito concedido
7	Disponible	Numérico (15,2)	Valor en efectivo disponible por el sujeto de crédito
8	Ingresos totales	Numérico (15,2)	Valor de los ingresos totales que percibe el sujeto de crédito de forma mensual
9	Total de bienes	Numérico (15,2)	Valor total de los bienes de los que dispone el sujeto de crédito
10	Egresos totales	Numérico (15,2)	Valor de los egresos totales que tiene el sujeto de crédito
11	Total de deudas	Numérico (15,2)	Valor de las deudas totales que tiene el sujeto de crédito

Se consideraron adicionalmente los siguientes indicadores:

- **Capacidad de pago:** [Ingresos totales – Egresos totales]
- **Capacidad de endeudamiento:** [Pasivo Corriente / Activo Corriente]

El considerar estos dos indicadores va a permitir medir y evaluar la capacidad de solvencia del sujeto de crédito, lo que lleva a la reducción del riesgo en la institución financiera.

Un análisis especial fue el comportamiento del crédito una vez que este había sido otorgado. Entre los datos analizados se revisó la morosidad del cliente, y si este se encontraba con cartera vencida superior a sesenta días, o incluso aquellos créditos cuya posibilidad de recuperación fuera mínima o prácticamente nula, lo que se conoce como cartera castigada. Si ocurrían estas condiciones implicaba que el comportamiento del sujeto crediticio no era el adecuado por lo cual el crédito no se debió conceder. La clase resultante debió ser negada.

Si la morosidad del cliente se encontraba entre cero y sesenta días se consideró que el crédito presentaba riesgo. La ventaja de este análisis de los datos es que se revisó el comportamiento del sujeto de crédito en un horizonte de tiempo, incluyendo el comportamiento pasado del cliente. Para el análisis de la morosidad se tomó en cuenta una nueva variable que es la "calificación crediticia del sujeto de crédito", la misma que toma en cuenta el comportamiento del pago del cliente, de acuerdo a la tabla 5.5.

TABLA 5.5: Morosidad

Categoría	Tiempo	Riesgo	Etiquetas lingüísticas
A	0 – 30 días de mora	Normal	Bajo
B	31 – 60 días de mora	Aceptable, superior al normal	Bajo
C	61 – 90 días de mora	Apreciable	Medio
D	91 – 180 días de mora	Significativo	Alto
E	180 días de mora	Incobrabilidad	Alto

Adicionalmente se consideró la situación financiera del país con las variables macroeconómicas. Las que se utilizaron en el análisis se indican en la tabla 5.6.

TABLA 5.6: Variables macroeconómicas de análisis

Nro.	Variable	Tipo de Dato	Descripción
1	Indice de precios al consumidor IPC	Numérico (4,2)	Indica el aumento general de los precios del mercado
2	Remuneración básica unificada (RBU)	Numérico (6,2)	Valor del salario básico
3	Precio de acción de la favorita	Numérico (6,2)	Es una de las empresas que siempre se cotiza en la bolsa de valores del mercado ecuatoriano, siendo unas de las principales referentes de la economía del país

5.2.2. Fuzzificación de las variables

En el capítulo 2 se explicó con detalle en que consiste la fuzzificación de las variables.

En el análisis del crédito se utilizaron las siguientes variables difusas

- Valor de la operación
- Disponible
- Ingresos totales

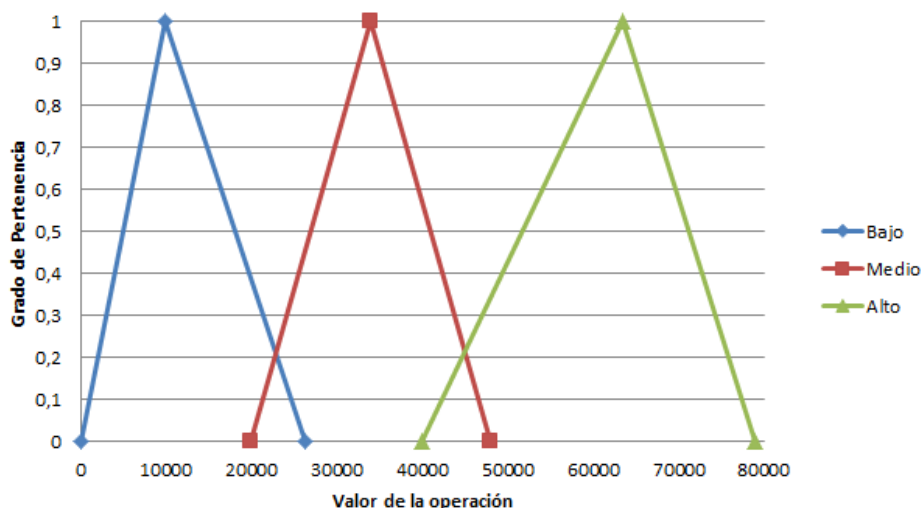


FIGURA 5.3: Función de pertenencia triangular de la variable "Valor de la operación"

- Total de bienes
- Egresos totales
- Total de deudas

El primer paso fue definir las variables lingüísticas que se iban a utilizar junto con las funciones de pertenencia de sus respectivos conjuntos difusos. Se utilizaron los tres métodos definidos en detalle en la sección 4.1. Para la clase resultante se consideraron los valores "O" y "N" para identificar a los créditos otorgados y negados, respectivamente. En las Figuras 5.3, 5.4, y 5.5 se indica la función de pertenencia definida por el experto crediticio para varias variables de entrada. Los datos son de la Cooperativa de Ahorro y Crédito del Ecuador que se encuentra en el segmento 2 dentro de la Superintendencia de Economía Popular y Solidaria, activos mayor a 20'000.000,00 hasta 80'000.000,00

Independientemente del método utilizado, todas las variables difusas utilizan la siguiente tres etiquetas lingüísticas *BAJO*, *MEDIO* y *ALTO*.

5.3. Resultados obtenidos

Para poder determinar si el método propuesto es efectivo, primero se evaluó el rendimiento de varios métodos que combinan dos tipos de PSO (detallado en el Capítulo 3) uno de población fija y otro de población variable. Inicializados con dos redes neuronales competitivas diferentes LVQ y SOM. El resultado de las reglas de clasificación del sistema difuso además de indicar si el crédito se debe conceder o no, en el caso que de que no se conceda se puede considerar el nivel de riesgo que existe para la institución financiera al otorgar el crédito.

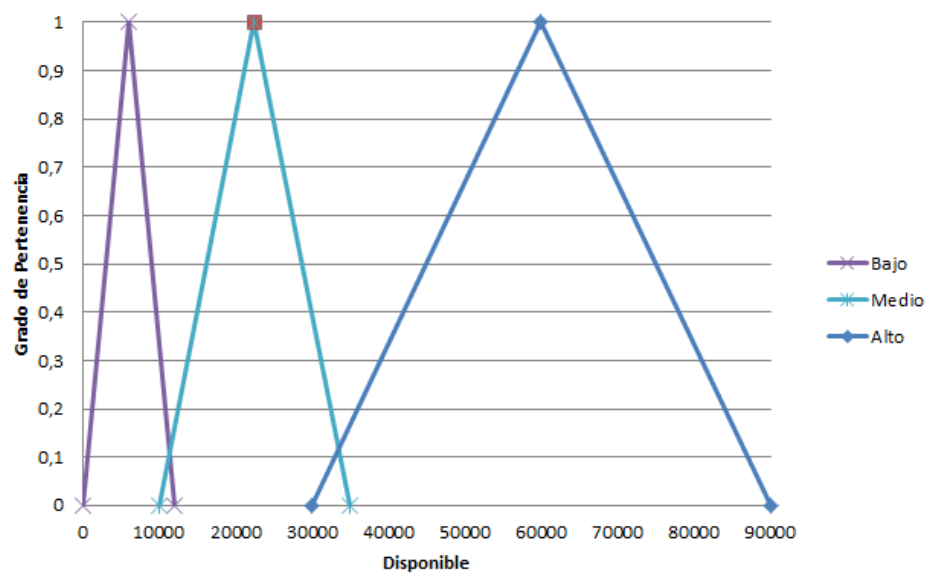


FIGURA 5.4: Función de pertenencia triangular de la variable "Disponible"

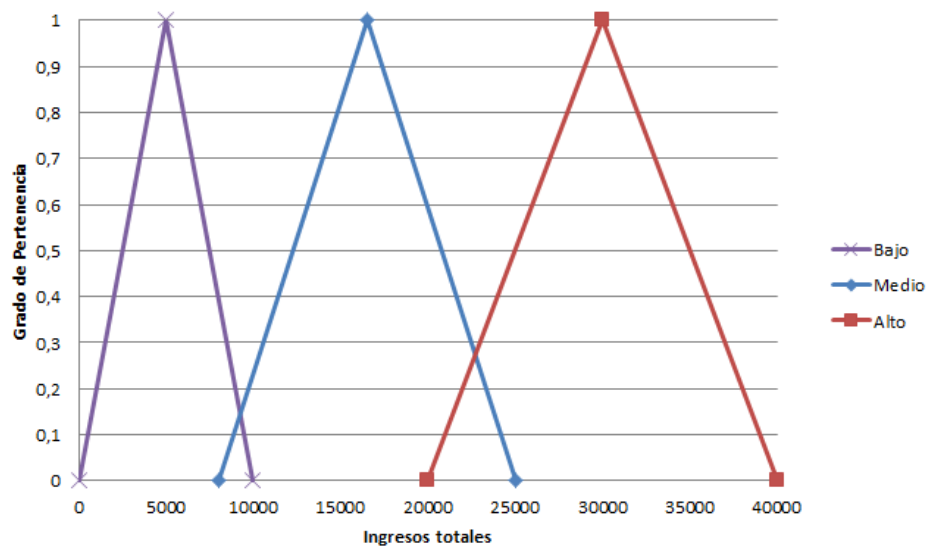


FIGURA 5.5: Función de pertenencia triangular de la variable "Ingresos totales"

Se realizaron 30 ejecuciones de forma independiente para cada uno de los métodos. La característica más importante de los resultados obtenidos con **FRvarPSO** es la mejora en la precisión, con respecto a otras investigaciones como por ejemplo las de (Córdova, Molina y Navarrete, 2017) donde también se utiliza lógica difusa para la obtención de reglas. Otra de las variantes importante es que la lista de reglas de clasificación difusas se obtiene a partir de variables micro y macroeconómicas.

5.3.1. Utilizando el criterio del experto crediticio

A continuación se presentan seis tablas, cada una de ellas tiene una matriz de confusión donde se indican el número de clases pronosticadas. Esta matriz contiene la información de las clasificaciones correctas, es decir, aquellos créditos que fueron clasificados correctamente en créditos otorgados o negados. Los créditos que fueron negados cuando en realidad debían ser otorgados, representa una pérdida para la entidad financiera, este error toma el nombre de error *Tipo I* o también denominado *Falso Positivo*.

Los créditos que fueron otorgados cuando de acuerdo al análisis del sujeto de crédito no debían haberse concedido, que son los créditos que cumplen con el consecuente de la regla de clasificación pero no con el antecedente. Estos créditos se indican como el error de *Tipo II* o denominados *Falsos Negativos*. Aquí es donde el oficial de crédito debe prestar más atención debido a que estos son los créditos que incurren en morosidad y son los que representan mayor riesgo para la institución financiera. De cada uno de estos valores se indica su desviación estándar.

Posteriormente se indica la precisión que es calculada como el cociente entre las predicciones correctas y el número total de predicciones (correctas e incorrectas), también se puede observar el número promedio de reglas obtenidas, y el promedio de variables utilizadas para formar el antecedente de la regla.

En la base de datos de la Cooperativa de ahorro y crédito, el método **LVQ+varPSO** tiene una precisión del 78,94 %, en el Banco encargado de microcrédito, el método **LVQ+varPSO** tiene una precisión del 97,78 %, finalmente la institución financiera encargada de crédito de consumo y crédito productivo o empresarial en el método **LVQ+varPSO** tiene una precisión del 98,80 %.

Utilizando los modelos difusos en la base de datos de la Cooperativa de ahorro y crédito, el método **FRvarPSO** tiene una precisión del 79,88 %, con un 1 % menos de reglas que el método no difuso. En el Banco encargado de microcrédito, especializado en colocación masiva de crédito, el método **FRvarPSO** tiene una precisión del 98,80 %, con un 13,4 % menos de reglas que el método no difuso. Finalmente la institución financiera encargada de crédito de consumo y crédito productivo o empresarial, en el método **FRvarPSO** tiene una precisión del 85,01 %, con un 4,62 % menos de reglas.

Como se puede observar en la Tabla 5.7 en las tres instituciones financieras la precisión de la clasificación que se obtiene utilizando **FRvarPSO** es mejor comparada con los otros métodos que no utilizan lógica difusa. De esta forma se cumple el objetivo que es que el oficial de crédito pueda responder rápido al cliente con la mayor exactitud, utilizando variables microeconómicas y macroeconómicas.

Con estos datos se ratifica que el método que utiliza lógica difusa es una excelente alternativa para ser utilizado en las instituciones financieras. De esta forma no solo se puede obtener el perfil de un buen cliente lo que lleva a la reducción del riesgo crediticio, con la disminución de la morosidad. Si no, que a futuro va a poder permitir determinar "*buenos clientes*", "*clientes en riesgo*", "*malos clientes*".

TABLA 5.7: Cooperativa de Ahorro y Crédito del Ecuador

Método	Tipo de predicción	Negados	Aceptados	Precisión	# reglas	Long. del antecedente
SOM + PSO	Negados	0.6023 ±0.0029	0.1358 ±0.0017	0.7718	6.2966	4.3952
	Aceptados	0,0923 ±0.0019	0.1695 ±0.0022	±0.0019	±0.2990	±0.1439
SOM + varPSO	Negados	0.6070 ±0.0042	0.1439 ±0.0047	0.7701	5.6899	4.8471
	Aceptados	0.0859 ±0.0037	0.1631 ±0.0049	±0.0023	±0.2801	±0.17714
LVQ + PSO	Negados	0.6113 ±0.0038	0.1147 ±0,0041	0.7871	5.4772	5.2956
	Aceptados	0.0982 ±0.0029	0.1758 ±0.0045	±0.0031	±0.1837	±0.1834
LVQ + varPSO	Negados	0.6136 ±0.0046	0.0996 ±0,0044	0.7894	5.3535	4.9001
	Aceptados	0.1109 ±0.0043	0.1758 ±0.0045	±0.0023	±0.2650	±0.3441

En las figuras 5.6, 5.7 y 5.8 se puede observar el análisis comparativo de la precisión promedio de cada método.

5.3.2. Resultados comparativos con conjuntos equitativos, FCM y criterio del experto crediticio

5.4. Conclusiones

En este capítulo se ha visto la utilidad del método **FRvarPSO** propuesto en esta tesis. Este método es un claro ejemplo de como la técnica propuesta se utilizó en casos reales. Su utilidad se puede ver en la descripción del perfil de los clientes y predecir su comportamiento para poder otorgar o negar el crédito. Se consideró en su análisis la situación del cliente y adicionalmente variables macroeconómicas que

TABLA 5.8: Cooperativa de Ahorro y Crédito del Ecuador

Método	Tipo de predicción	Negados	Aceptados	Precisión	# reglas	Long. del antecedente
SOM + PSO Difuso	Negados	0.6341 ±0.0048	0.1167 ±0.0045	0.7857	5,9874	7.7330
	Aceptados	0,0976 ±0.0036	0.1516 ±0.0045	±0.0026	±0.2429	±0.3362
SOM + varPSO Difuso	Negados	0.6235 ±0.0034	0.0870 ±0.0030	0.7891	5.6920	7.9917
	Aceptados	0.1238 ±0.0033	0.1656 ±0.0032	±0.0021	±0.2245	±0.2953
LVQ + PSO Difuso	Negados	0.6571 ±0.0031	0.1122 ±0.0029	0.7922	5.4958	7.6240
	Aceptados	0.0955 ±0.0036	0.1351 ±0.0027	±0.0029	±0.2128	±0.2888
LVQ + varPSO Difuso	Negados	0.6499 ±0.0021	0.1145 ±0,0017	0.7988	5.2990	8,3969
	Aceptados	0.0867 ±0.0016	0.1489 ±0.0028	±0.0029	±0.1907	±0.3913

TABLA 5.9: Resultado de Reglas Banco del Ecuador encargado de Microcrédito

Método	Tipo de predicción	Negados	Aceptados	Precisión	# reglas	Long. del antecedente
SOM + PSO	Negados	0.7873 ±0.0019	0.0167 ±0.0019	0.9688	7.4939	5,8892
	Aceptados	0,0044 ±0.0025	0.1815 ±0.0020	±0.0033	±0.2928	±0.3901
SOM + varPSO	Negados	0.8005 ±0.0028	0.0114 ±0.0031	0.9700	7.6936	5.9966
	Aceptados	0.0085 ±0.0029	0.1695 ±0.0027	±0.0026	±0.2642	±0.3005
LVQ + PSO	Negados	0.8020 ±0.0035	0.0105 ±0,0031	0.9737	7.6993	5.6234
	Aceptados	0.0058 ±0.0031	0.1717 ±0.0032	±0.0037	±0.2307	±0.3866
LVQ + varPSO	Negados	0.8032 ±0.0024	0.0076 ±0,0021	0.9778	7.3894	5.1209
	Aceptados	0.0046 ±0.0018	0.1746 ±0.0015	±0.0018	±0.1880	±0.3032

TABLA 5.10: Resultado de Reglas Difusas Banco del Ecuador encar-
gado de Microcrédito

Método	Tipo de predicción	Negados	Aceptados	Precisión	# reglas	Long. del antecedente
SOM + PSO Difuso	Negados	0,8028 ±0.0036	0.0086 ±0.0049	0.9819	6.9930	5.7707
	Aceptados	0,0065 ±0.0032	0.1791 ±0.0030	±0.0037	±0.2121	±0.3480
SOM + varPSO Difuso	Negados	0.8038 ±0.0034	0.0071 ±0.0032	0.9840	6.1888	5.6533
	Aceptados	0.0089 ±0.0029	0.1802 ±0.0035	±0.0032	±0.2828	±0.2547
LVQ + PSO Difuso	Negados	0.8004 ±0.0016	0.0076 ±0,0011	0.9869	6.7588	5.9593
	Aceptados	0.0034 ±0.0019	0.1865 ±0.0023	±0.0024	±0.2642	±0.2997
LVQ + varPSO Difuso	Negados	0.9037 ±0.0023	0.0089 ±0,0024	0.9880	6.3972	5.1549
	Aceptados	0.0031 ±0.0027	0.0843 ±0.0029	±0.0026	±0.1915	±0.2048

TABLA 5.11: Resultado de Reglas Banco del Ecuador encargado de
crédito de consumo y crédito productivo o empresarial

Método	Tipo de predicción	Negados	Aceptados	Precisión	# reglas	Long. del antecedente
SOM + PSO	Negados	0,6634 ±0.0019	0.1091 ±0.0019	0.8133	8.4994	6.7538
	Aceptados	0,0775 ±0.0015	0.1499 ±0.0017	±0.0024	±0.1907	±0.2547
SOM + varPSO	Negados	0.6675 ±0.0038	0.1057 ±0.0035	0.8264	8.2951	6.5513
	Aceptados	0.0679 ±0.0031	0.1589 ±0.0034	±0.0041	±0.2121	±0.3928
LVQ + PSO	Negados	0.6523 ±0.0015	0.1221 ±0,0021	0.8256	6.4926	6.2304
	Aceptados	0.0522 ±0.0015	0.1733 ±0.0032	±0.0028	±0.3145	±0.3854
LVQ + varPSO	Negados	0.6670 ±0.0034	0.1118 ±0,0022	0. 8315	6.2801	5.9569
	Aceptados	0.0566 ±0.0027	0.1645 ±0.0028	±0.0035	±0.3536	±0.2601

TABLA 5.12: Resultado de reglas Difusas Banco del Ecuador encargado de crédito de consumo y crédito productivo o empresarial

Método	Tipo de predicción	Negados	Aceptados	Precisión	# reglas	Long. del antecedente
SOM + PSO Difuso	Negados	0,6723 ±0.0019	0.1023 ±0.0012	0.8316	6.4415	6.5538
	Aceptados	0,0721 ±0.0015	0.1593 ±0.0027	±0.0022	±0.1814	±0.3469
SOM + varPSO Difuso	Negados	0.6670 ±0.0038	0.0999 ±0.0022	0.8385	6.3990	6.2832
	Aceptados	0.0016 ±0.0025	0.1715 ±0.0034	±0.0033	±0.2099	±0.3496
LVQ + PSO Difuso	Negados	0.6744 ±0.0025	0.0744 ±0.0019	0.8455	6.6904	6.1818
	Aceptados	0.0945 ±0.0021	0.1711 ±0.0032	±0.0029	±0.3021	±0.3176
LVQ + varPSO Difuso	Negados	0.6627 ±0.0034	0.0903 ±0,0028	0.8501	5.9901	5.9623
	Aceptados	0.0669 ±0.0030	0.1874 ±0.0022	±0.0033	±0.3112	±0.2854

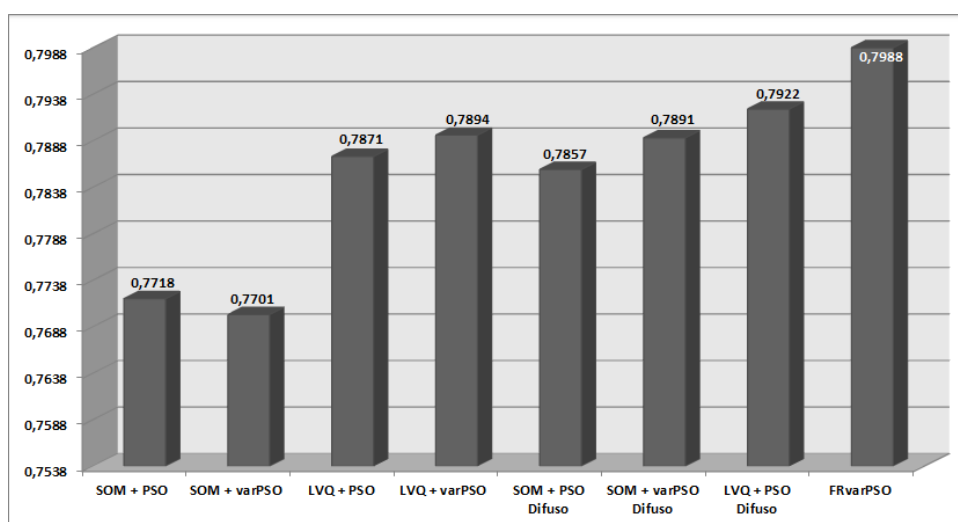


FIGURA 5.6: Análisis comparativo de la precisión promedio de cada método en la Cooperativa de Ahorro y Crédito

TABLA 5.13: Comparación de la precisión entre los modelos híbridos y aquellos que utilizan Lógica Difusa

	Método	Precisión
Cooperativa de Ahorro y Crédito	SOM + PSO	0.7718
	SOM + varPSO	0.7701
	LVQ + PSO	0.7871
	LVQ + varPSO	0.7894
	SOM + PSO Difuso	0.7857
	SOM + varPSO Difuso	0.7891
	LVQ + PSO Difuso	0.7922
	FRvarPSO	0.7988
Banco del Ecuador encargado de Microcrédito, especializado en colocación masiva de crédito	SOM + PSO	0.9688
	SOM + varPSO	0.9700
	LVQ + PSO	0.9737
	LVQ + varPSO	0.9778
	SOM + PSO Difuso	0.9819
	SOM + varPSO Difuso	0.9840
	LVQ + PSO Difuso	0.9869
	FRvarPSO	0.9880
Banco del Ecuador encargado de crédito de consumo y crédito productivo o empresarial	SOM + PSO	0.8133
	SOM + varPSO	0.8264
	LVQ + PSO	0.8256
	LVQ + varPSO	0.8315
	SOM + PSO Difuso	0.8316
	SOM + varPSO Difuso	0.8385
	LVQ + PSO Difuso	0.8455
	FRvarPSO	0.8501

TABLA 5.14: Resultado de reglas Difusas Cooperativa de Ahorro y Crédito del Ecuador

Método	Tipo de predicción	Negados	Aceptados	Precisión	# reglas
FRvarPSO Conjuntos Equitativos	Negados	0.6120 ±0.0034	0.0865 ±0.0030	0.7815	5.7321
	Aceptados	0.1319 ±0.0027	0.1695 ±0.0022	±0.0041	±0.1357
FRvarPSO FCM	Negados	0.6296 ±0.0035	0.0857 ±0.0022	0.7901	5.4081
	Aceptados	0.1118 ±0.0039	0.1728 ±0.0027	±0.0092	±0.1275
FRvarPSO Experto	Negados	0.6499 ±0.0021	0.1145 ±0.0017	0.7988	5.2990
	Aceptados	0.0867 ±0.0016	0.1489 ±0.0028	±0.0029	±0.1907

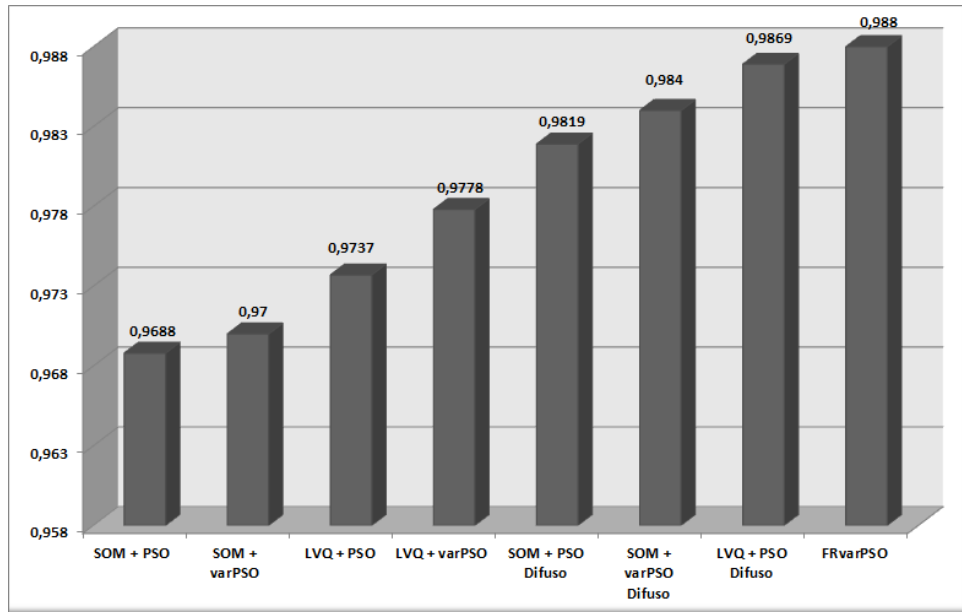


FIGURA 5.7: Análisis comparativo de la precisión promedio de cada método en el Banco del Ecuador encargado de Microcrédito, especializado en colocación masiva de crédito

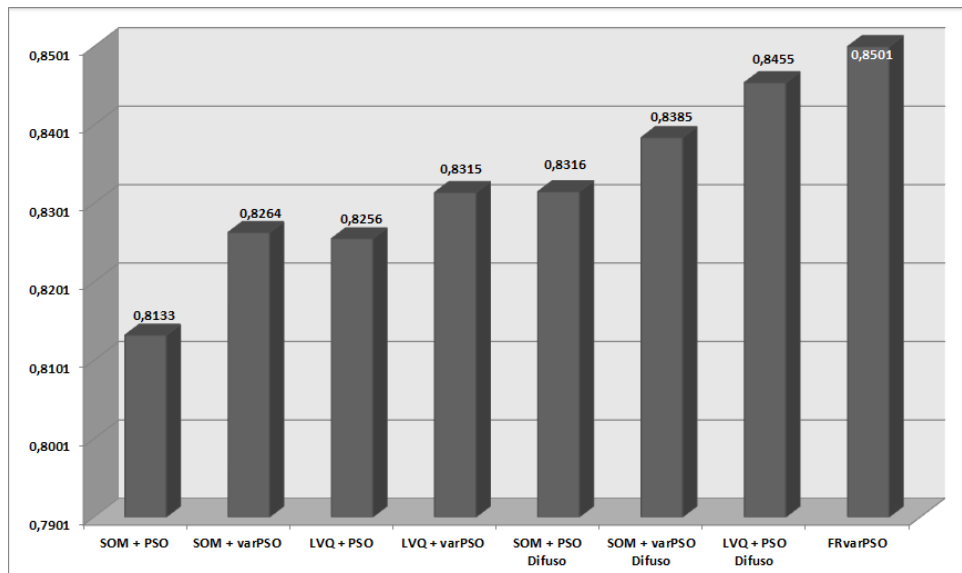


FIGURA 5.8: Análisis comparativo de la precisión promedio de cada método en Banco del Ecuador encargado de crédito de consumo y crédito productivo o empresarial

TABLA 5.15: Resultado de reglas Difusas Banco del Ecuador encargado de Microcrédito

Método	Tipo de predicción	Negados	Aceptados	Precisión	# reglas
FRvarPSO Conjuntos Equitativos	Negados	0.8623 ±0.0029	0,0382 ±0.0017	0.9551	6.9086
	Aceptados	0.0072 ±0.0019	0.0922 ±0.0022	±0.0057	±0.2011
FRvarPSO FCM	Negados	0.8835 ±0.0038	0.0067 ±0.0041	0.9872	6.5901
	Aceptados	0.0056 ±0.0029	0.1041 ±0.0045	±0.0005	±0.1723
FRvarPSO Experto	Negados	0.9037 ±0.0023	0.0089 ±0,0024	0.9880	6.3972
	Aceptados	0.0031 ±0.0027	0.0843 ±0.0029	±0.0026	±0.1915

TABLA 5.16: Resultado de reglas Difusas Banco encargado de crédito de consumo y crédito productivo o empresarial

Método	Tipo de predicción	Negados	Aceptados	Precisión	# reglas
FRvarPSO Conjuntos Equitativos	Negados	0.6728 ±0.0022	0.0981 ±0.0017	0.8423	6.8716
	Aceptados	0.0595 ±0.0019	0.1695 ±0.0020	±0.0027	±0.3018
FRvarPSO FCM	Negados	0.6818 ±0.0029	0.0807 ±0.0041	0.8488	6.3102
	Aceptados	0.0704 ±0.0032	0.1670 ±0.0037	±0.0092	±0.2872
FRvarPSO Experto	Negados	0.6627 ±0.0034	0.0830 ±0.0028	0.8501	5.9901
	Aceptados	0.0669 ±0.0030	0.1874 ±0.0022	±0.0033	±0.3112

denotan la situación del país, la misma que refleja el entorno en el que se entrega el crédito. Tanto las variables microeconómicas como las macroeconómicas son las que influyen directamente en la morosidad del cliente.

El modelo difuso **FRvarPSO** completa el enfoque en el análisis crediticio, disminuyendo el riesgo, considerando que es aplicable a todos los clientes (nuevos o con créditos renovados). El análisis realizado en este trabajo resulta de sumo interés para el oficial de crédito, ya que además de influir en la determinación del scoring de crédito, es posible analizar el riesgo que representa para la institución financiera la concesión del crédito. En el caso de que el análisis indique que el crédito debe ser negado, este dependiendo del peso que tiene la regla puede ser concedido incrementando garantías o aumentando la tasa de interés.

Capítulo 6

Conclusiones Finales y Líneas de Trabajo Futuras

6.1. Conclusiones

Dentro de las etapas de **KDD** la minería de datos es considerada como el proceso de extraer conocimiento desde grandes bases de datos, para lo cual se utilizan un conjunto de técnicas. Algunos autores definen de forma general tres etapas entre las que se puede considerar: como primera obtener la vista de datos minable, la misma que debe contener ya la limpieza y transformación de los datos; la segunda se encarga de la generación del modelo y la tercera y última etapa consiste en analizar e interpretar los resultados obtenidos.

El conocimiento que se genera utilizando técnicas de minería de datos resulta de gran interés para las organizaciones, ya que les va a permitir la toma de decisiones tácticas y estratégicas, lo que se convierte en una ventaja competitiva. Existe un conjunto de técnicas presentes en la minería de datos, cada una de ellas tiene características diferentes. Para poder seleccionar el tipo de técnica es fundamental tener conocimiento del problema que se va a resolver. En el estado del arte se pueden encontrar referencias que indican que se obtienen mejores resultados cuando se utilizan modelos híbridos, aquí se enmarca el modelo desarrollado.

Uno de los objetivos de esta tesis es la revisión de los diferentes métodos que permiten la obtención de reglas de clasificación, presentando un nuevo método que encuentra una mejor solución comparado con otras técnicas. El método propuesto denominado **FRvarPSO** (Fuzzy Rules Variable Particle Swarm Optimization), permite la representación del conocimiento en términos difusos a través de reglas de clasificación difusas extraídas de un conjunto de datos de entrada.

Las reglas de clasificación difusas obtenidas a través de **FRvarPSO** se caracterizan por tener cardinalidad baja y precisión adecuada, y especialmente por ser fáciles de interpretar gracias a la incorporación de la lógica difusa. Para conseguir esto se consideraron dos aspectos importantes; el primero tiene que ver con la capacidad del método de operar con variables lingüísticas y el segundo se relaciona con la

inserción de información basada en grados de pertenencia tanto en la evaluación de la función fitness, así como en la manera en que se realiza la búsqueda por medio de la técnica de optimización.

El método inicia con la determinación de los conjuntos difusos, y las funciones de pertenencia de cada variable difusa asociada a cada uno de ellos, es aquí donde radica la eficiencia y eficacia de este método. Para esto se utilizaron dos variantes. La primera variante se encarga de particionar el rango de cada atributo numérico en intervalos de igual longitud centrandlo, en cada uno de ellos, una función triangular con un solapamiento adecuado. En este caso se consideraron particiones difusas uniformes con el mismo número de etiquetas para cada una de las variables y se definieron los conjuntos difusos de tal manera que el valor de una variable difusa sólo puede pertenecer a dos conjuntos de forma simultánea. En la segunda variante se mejora el proceso de obtención de los conjuntos difusos utilizando el algoritmo **FCM** con tres clusters, donde cada uno de ellos representaba un conjunto difuso.

Posteriormente se utiliza una técnica de optimización de población variable **var-PSO** la misma que emplea una red neuronal competitiva **LVQ**, con la finalidad de obtener las zonas más prometedoras para realizar de búsqueda. Esta información suministrada por dicha red es utilizada para inicializar el cúmulo. El método utiliza el grado de pertenencia de las variables difusas, incidiendo directamente a través de la técnica de optimización **varPSO**, en el vector velocidad que controla el movimiento de las partículas. Ninguna de las técnicas por separado logra obtener los resultados que se indican en esta investigación. Los resultados obtenidos permiten afirmar que el modelo híbrido propuesto **FRvarPSO** tiene un mejor desempeño que sus versiones anteriores a raíz de la incorporación de la lógica difusa.

Una de las ventajas del método propuesto es que además de probarlo con bases de datos del repositorio **UCI**, se realizaron pruebas con base de datos reales pertenecientes a tres instituciones financieras. Aquí, para la determinación de los conjuntos difusos se utilizó el criterio de un experto en el área de crédito. Adicionalmente, se incorporó en el modelo no solo las variables del sujeto de crédito, si no también aquellas variables que hacen referencia a la economía del país en su conjunto, que son las variables macroeconómicas. Esto da una situación más real para la toma de decisiones, por parte del oficial de crédito en las instituciones financieras.

En este caso **FRvarPSO** no solo va a influir en la determinación del scoring de crédito, permitiendo analizar el riesgo que representa para la institución financiera la concesión del crédito, si no que en el caso de que el análisis indique que el crédito debe ser negado, este dependiendo del peso que tiene la regla de clasificación difusa puede ser concedido incrementado garantías o aumentando la tasa de interés. Toda estas características hacen que **FRvarPSO** sea una buena opción para análisis de riesgo crediticio.

FRvarPSO no solo obtiene un modelo más simple ya que utiliza menor cantidad

de reglas que otros métodos, si no que presenta una buena precisión, y especialmente gracias a la incorporación de la lógica difusa mejora la interpretabilidad de la regla. Luego, este método aporta al área informática y se ha demostrado que también realiza aportes en el área de la economía, a través del análisis del riesgo crediticio, incorporando para ello variables macro económicas.

Todo lo anterior conlleva a que los objetivos propuestos en esta investigación se cumplieron.

6.2. Líneas de trabajo futuras

Es importante que en investigaciones futuras se consideren:

Una optimización de la función de pertenencia. Dentro de esta una alternativa puede ser la utilización de algoritmos genéticos, o de la misma optimización por cúmulo de partículas, con la finalidad de identificar automáticamente los parámetros de dicha función, sin perder de lado el objetivo que es tener un conjunto de reglas reducidas, pero conservando la interpretabilidad con las variables lingüísticas.

Un segundo aspecto que se debe considerar es la obtención de una nueva forma de representación de los atributos nominales, con el objetivo de reducir el tiempo computacional, y disminuir la longitud de la representación del antecedente dentro de cada partícula.

Otra variante que se debe tener en cuenta es incorporar técnicas para el manejo del desbalance de las clases, ya que en el área de riesgo crediticio la clase que corresponde a los créditos otorgados es la que tiene la mayor cantidad de ejemplos.

Es necesario incorporar al modelo la defuzificación de la variable de salida. Este valor crisp que se obtiene de la aplicación de la reglas de clasificación, va a indicar el porcentaje de riesgo que existe. En este caso para las reglas de clasificación aplicables a riesgo crediticio, se va a tener no solamente como salida la concesión o no del crédito, si no también el riesgo que implica el cliente, logrando disminuir considerablemente las consecuencias negativas para la institución financiera. Este mismo criterio puede ser utilizado en otras áreas de riesgo como el operacional, liquidez entre otros en el área financiera. Incluso el método propuesto puede utilizarse en otras áreas como la medicina, para saber el riesgo que un paciente tiene en contraer una determinada enfermedad.

Apéndice A

Redes Neuronales para inicialización de PSO

Según (Abu-Mostafa, 1986) los problemas que necesitan resolverse con extensos algoritmos o utilizan un gran número de datos, aprovechan mejor la estructura de una red neuronal. Este es el caso de esta investigación, el método propuesto **FRvarPSO** para la inicialización de **PSO** se toma como punto de partida las redes neuronales, considerando dos tipos de red **SOM** y **LVQ**. Sin embargo hay que considerar que el resultado del entrenamiento varía en función del tamaño de la red, y los valores de los pesos iniciales.

Las redes neuronales emulan el hardware del cerebro con la finalidad de reproducir algunas de sus capacidades asociadas a la inteligencia, como pueden ser el reconocimiento de patrones, percepción entre otros. La unión entre dos neuronas se denomina sinapsis, aquí se puede hablar de las neuronas que envían las señales que son neuronas presinápticas y aquellas que reciben las señales que son las postsinápticas, la información que fluye a través de la sinapsis es unidireccional.

Al tratar las redes neuronales de simular los sistemas nerviosos, podemos hablar de los sistemas paralelos, distribuidos y adaptativos.

Los elementos que constituyen una neurona i son:

- Entradas $x_j(t)$ estas variables pueden ser binarias o continuas
- Pesos sinápticos de la neurona i , p_{ij} es la intensidad de la interacción entre la neurona presináptica j y la neurona postsináptica i .
- Regla de propagación $\sigma(P_{ij}, X_j(t))$ que proporciona el valor del potencial postsináptico $h_i(t) = \sigma(P_{ij}, X_j(t))$ de la neurona i en función de sus pesos y entradas
- Función de activación $f_i(a_i(t-1), h_i(t))$ proporciona el estado de activación actual $a_i(t) = f_i(a_i(t-1), h_i(t))$ de la neurona i , en función de su estado anterior $a_i(t-1)$ de su potencial sináptico actual. En muchos modelos se considera

que el estado actual de la neurona no depende de su estado anterior, sino únicamente del actual. Esta función es determinista y en la mayor parte de los modelos es monótona creciente y continua

- Función de salida $F_i(a_i(t))$ proporciona la salida actual $y_i(t) = F_i(a_i(t))$ de la neurona i en función de su estado de activación.

Considerando todo lo anterior la operación de la neurona i puede expresarse como (A.1):

$$y_i(t) = F_i(f_i[a_i(t-1), \sigma_i(p_{ij}, x_j(t))]) \quad (\text{A.1})$$

Una regla de propagación que permite obtener a partir de las entradas y los pesos, el valor del potencial postsináptico h_i de la neurona es la distancia Euclídea, utilizada en los mapas de Kohonen o **LVQ**, que se indica en la ecuación (A.2).

$$h_i^2(t) = \sum_j (X_j - P_{ij})^2 \quad (\text{A.2})$$

donde h_i^2 Representa la distancia al cuadrado que existe entre el vector de entradas y el de pesos. Cuando ambos vectores son muy similares la distancia es muy pequeña y cuando son muy diferentes la distancia es mayor.

Cuando se construye un sistema neuronal se empieza de un cierto modelo de neurona, y de una determinada arquitectura de red. Se establecen los pesos sinápticos iniciales como nulos o aleatorios, posteriormente para que la red resulte operativa es necesario realizar el respectivo entrenamiento, lo que toma el nombre de **aprendizaje**.

En las redes neuronales se define el **aprendizaje** como el proceso por el que se produce el ajuste de los parámetros libres de la red, a partir de un proceso de estimulación por el entorno que rodea la red.

Este entrenamiento puede efectuarse en dos niveles:

- Modelado de la sinapsis: modifica los pesos sinápticos siguiendo una cierta regla de aprendizaje, la misma que se construye a partir de la optimización de una función de error o coste. Si se denomina $P_{ij}(t)$ al peso que conecta la neurona presináptica j con la postsináptica i en la iteración t , el algoritmo de aprendizaje proporcionará $\Delta P_{ij}(t)$ dando la modificación que se debe incorporar al peso, hasta que la red alcanza el rendimiento que se requiere con lo que se tiene, la ecuación (A.3):

$$\Delta P_{ij}(t+1) = P_{ij}(t) + \Delta P_{ij}(t) \quad (\text{A.3})$$

- Modificación de la propia arquitectura de red: en este nivel de aprendizaje se tiene la creación o destrucción de las neuronas

Para que la red resulte operativa es necesario entrenarla, lo que constituye el modo aprendizaje. Este puede ser de varios tipos el **supervisado**: al que se le proporciona información sobre las funciones (distribución de clases, etiquetas de patrones de entrada o salidas asociadas a cada patrón), es decir a la red se le presenta un conjunto de patrones conjuntamente con la salida deseada, iterativamente se van ajustando los pesos hasta obtener la salida que se requiere. En el **No Supervisado o autoorganizado**: no se proporciona ninguna información, es decir se presenta a la red multitud de patrones sin adjuntar la respuesta que se desea. Aquí la red por medio de la regla de aprendizaje estima $\rho(x)$, a partir de lo cual pueden reconocerse regularidades en el conjunto de entradas, extraer rasgos o agrupar patrones según su similitud (clustering). El tercer tipo es el aprendizaje **híbrido**, en este tipo de aprendizaje en la red coexisten los dos tipos de aprendizaje el supervisado y el no supervisado, los que tienen lugar en distintas capas de neuronas. Por último tenemos el **Aprendizaje reforzado**, se sitúa a medio camino entre el supervisado y el autorganizado. En el primero se indica lo bien o mal que esta actuando sin dar más información, como en el caso del no supervisado no se indica explícitamente la salida deseada.

Las redes neuronales se clasifican dependiendo del modelo de neurona concreta que se utilice, de la arquitectura o topología de conexión y del algoritmo de aprendizaje.

Una red neuronal es un grafo dirigido, en el cuál los nodos son las neuronas y las conexiones son las sinapsis, que posee las siguientes propiedades:

- A cada nodo i se asocia una variable de estado X_i
- A cada conexión (i, j) de los nodos i, j se asocia un peso $P_{i,j} \in R$
- A cada nodo i se asocia un umbral θ_i
- Para cada nodo i se define una función $f_i(x_j, p_{ij}, \theta_i)$, que depende de los pesos de sus conexiones, del umbral y de los estados de los nodos j a él conectados. Esta es la función que proporciona el nuevo estado del nodo.

Uno de los aspectos fundamentales de los Sistemas Neuronales Artificiales **Sistemas Neuronales Artificiales (ANS)**, en las redes neuronales supervisadas es la capacidad de generalizar a partir de los ejemplos, lo que constituye el problema de **memorización frente a generalización**. Hay que tener en cuenta que la generalización es la capacidad de la red para dar una respuesta correcta frente a patrones que no se consideraron en el entrenamiento; si la red neuronal ha sido correctamente entrenada entonces generalizará, por lo que responderá correctamente ante patrones nunca antes vistos.

En el proceso de entrenamiento es importante considerar el error de aprendizaje. Cuando la red es lo suficientemente grande este error puede reducirse solo llevando

a cabo un mayor número de iteraciones. Existe otro error, el error de generalización que se puede medir utilizando un conjunto de patrones diferentes a los utilizados en el entrenamiento. Una red neuronal puede ser entrenada utilizando un conjunto de patrones, y se puede comprobar su eficiencia real o **error de generalización** mediante un conjunto de test.

Hay que tener en cuenta que en un momento determinado el sistema se ajusta demasiado a las particularidades de los patrones empleados en el entrenamiento, aprendiendo incluso el ruido presente, por lo que crece el error que comete ante patrones diferentes a los utilizados en el entrenamiento (error de generalización). Es aquí cuando la red está memorizando los patrones del conjunto de aprendizaje, incluidos los errores que se pueden presentar, esto toma el nombre de **sobreaprendizaje o sobreajuste**.

La capacidad de generalización de la red la determina de forma general: *la arquitectura de la red, el número de ejemplos de entrenamiento y la complejidad del problema* (Kubat, 1999). En síntesis para evitar el sobre entrenamiento de la red se puede utilizar la *validación cruzada y limitar el tamaño de la arquitectura de la red*. Se recomienda utilizar el 80 % para el entrenamiento y el 20 % para test. También es necesario limitar el tamaño de la red, se debe elegir una arquitectura de tamaño mínimo que permita realizar el aprendizaje con los ejemplos disponibles. Se puede considerar la relación $\rho = 10.p$ donde ρ es el número de ejemplos y p es el número de pesos de la red. De esta forma se puede establecer el tamaño aproximado de la red considerada ideal para el conjunto de ejemplos disponibles, o dada una arquitectura de red tener indicios de si se puede incurrir en el sobre entrenamiento.

Para que un problema sea resuelto utilizando Redes Neuronales Artificiales para obtención de reglas debe cumplir:

- No se disponga de un conjunto de reglas sistemáticas que describan completamente el problema.
- Disponer de un conjunto grande de ejemplos.
- Conjunto de datos que sean imprecisos o que incluyan ruido.
- Problemas de elevada dimensionalidad.
- Si las situaciones son cambiantes.

A.1. SOM

Las redes **SOM** son modelos de redes neuronales artificiales no supervisadas unidireccionales (Kohonen, 1990). En la red neuronal no supervisada (auto organizada)

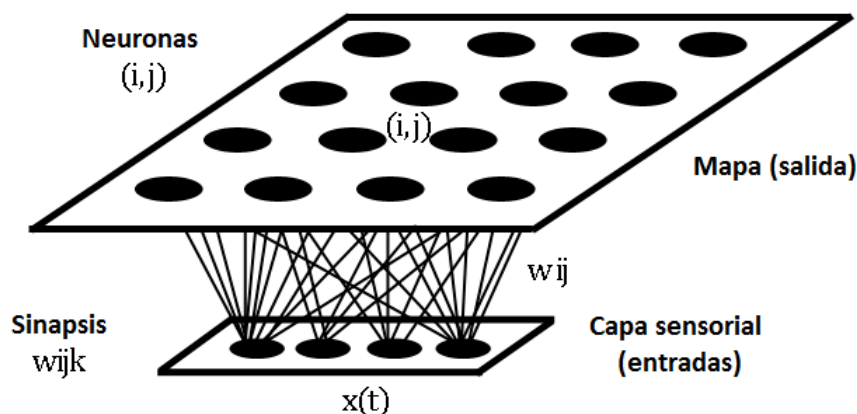


FIGURA A.1: Arquitectura del SOM

no se dispone de ninguna salida que indique si la red neuronal está operando correcta o incorrectamente. Es por esto que durante el aprendizaje la red autorganizada debe descubrir por si misma rasgos comunes, regularidades, correlaciones o categorías en los datos de entrada e incorporarlos a sus pesos (estructura interna de conexiones). Este tipo de redes se ha utilizado con eficacia en: clasificación de patrones, cuantificación vectorial, reducción de dimensiones, extracción de rasgos, monitorización de procesos, análisis exploratorio entre otros.

En este tipo de red las neuronas se organizan en una arquitectura unidireccional de dos capas, como se indica en la Figura A.1.

La primera capa es la capa de entrada, tiene m neuronas una por cada una de las variables de entrada, distribuyen la información procedente del espacio de entrada a las neuronas de la segunda capa. Las entradas son muestras estadísticas $X(t) \in R^m$ del espacio sensorial. El procesamiento se realiza en la segunda capa, consistente en una estructura rectangular de $n_x \times n_y$ neuronas que operan en paralelo. A cada una de las m neuronas de entrada con el índice $k (1 \leq k \leq m)$, y las $n_x \times n_y$ neuronas del mapa con un par de índices $i \equiv (i, j) (1 \leq i \leq n_x, 1 \leq j \leq n_y)$ determinando su localización espacial. Cada neurona de entrada k está conectada a todas las neuronas (i, j) del mapa mediante un peso sináptico $P_{i,j,k}$.

Aquí se introduce la función de vecindad, que no es más que el entorno alrededor de la neurona ganadora actual, de tal forma que durante el aprendizaje se actualizan los pesos de la neurona ganadora así como los de sus vecinas logrando que las neuronas próximas sintonicen con patrones similares. Esto permite que el ritmo efectivo de convergencia mejore y el sistema sea más robusto frente a variaciones en los valores iniciales de los pesos (Ritter, Martinetz. y Schulten, 1993).

En el algoritmo 5 se describe el algoritmo de aprendizaje de una red SOM.

El algoritmo SOM es más utilizado en visualización de datos que en clustering.

Algoritmo 5: Algoritmo de Aprendizaje para SOM

```

begin
    Inicializar con un valor de partida predeterminando los pesos sinápticos
     $P_{ij}$  partiendo desde  $t = 0$ 
    Determinar el número de iteraciones
    for cada una de las iteraciones do
        Presentar el patrón  $X(t)$ 
        for Cada neurona  $i \equiv (i, j)$  do
            Calcular la similitud entre su vector de pesos sinápticos  $P_{ij}$  y el
            actual vector de entrada  $X$ , utilizar la distancia euclidea
             $d^2(P_{ij}, X) = \sum_{k=1}^n (P_{ij} - X_k)^2$ 
        Determinar la neurona ganadora  $g \equiv (g_1, g_2)$  considerando aquella
        que tenga la menor distancia
        Actualizar los pesos sinápticos de la neurona ganadora así como de
        sus vecinas
         $P_{ijk}(t + 1) = P_{ijk}(t) + \text{alfa}(t) * h(|i - g|, t) * (X_k(t) - P_{ijk}(t))$ 
        donde Alfa es la velocidad de aprendizaje y toma al inicio el valor de
        0,25 y la función  $h$  es considerada la función de vecindad que
        depende de la distancia entre la neurona  $i$  y la ganadora  $g$ .
    
```

A.2. LVQ

LVQ (Kohonen, 1990) es un modelo de red neuronal que utiliza aprendizaje supervisado unidireccional compuesto por una capa simple de neuronas de Kohonen sin relaciones de vecindad. Durante el proceso adaptativo premia a aquellas neuronas que clasifican correctamente un determinando ejemplo actualizando los pesos con la regla convencional, y castiga a las que realizan una clasificación errónea, restando un determinado valor a sus pesos. La red neuronal **LVQ** es utilizada para clasificación.

LVQ basa su funcionamiento en una red neuronal competitiva formada por tres capas. La primera capa es la que recibe la información de entrada, la segunda capa es donde se realiza la competencia y finalmente la última capa es la encargada de realizar la clasificación y a su vez permite entregar los valores de salida. Cada neurona de la segunda capa o capa competitiva lleva asociado un vector de pesos numérico de igual dimensión que los ejemplos de entrada y una etiqueta que indica la clase a la cual va a representar. Estos vectores son los que al finalizar el proceso adaptativo contienen la información de los centroides o prototipos de la clasificación.

Si se considera que cada una de las neuronas de entrada representa una clase y se tiene un vector prototipo de una clase $P_j (j = 1, 2, 3, \dots, n)$, si se presenta a la red un patrón X compuesto por m entradas, cada una de las neuronas calculará la distancia que separa su vector de referencia con el de entrada a través de la distancia euclidea,

la misma que se indica en la ecuación (A.4).

$$d^2(P_j, X) = \sum_{k=1}^m (P_{jk} - X_k)^2 \quad (\text{A.4})$$

De acuerdo con este criterio, dos vectores tendrán mayor similitud mientras su distancia sea menor. Es decir que una correlación máxima se corresponde con una distancia euclídea mínima.

Entonces se puede indicar que la neurona c de menor distancia, cuyo vector de referencia es P_c se va a encargar de determinar la clase C en la que el patrón x queda clasificado.

El aprendizaje se basa en actualizar los pesos con la regla de Kohonen por medio del cual si la neurona clasifica correctamente el patrón la neurona recibe un premio. Por el contrario, si la clasificación es errónea sufre una penalización modificando los pesos en sentido contrario.

Si se dispone de p patrones de aprendizaje $X^u = (u = 1, \dots, p)$ donde cada patrón pertenece a una clase determinada C_{xu} , si se tiene un total de n clases así como neuronas, cada neurona de pesos P_j es etiquetada con una clase C_{pj} . Si se considera que P_c es el vector de referencia que se encuentra más próximo al patrón de entrada X^u y C_{pc} es la clase definida por el vector de referencia se tiene la siguiente actualización para la neurona C .

- Si $C_{pc} = C_{xu}$ la clasificación es correcta, en este caso existe un premio, el vector de referencia se actualiza, por la ecuación (A.5):

$$P_c(t+1) = P_c(t) + f_t \alpha(t) [X^u - P_c(t)] \quad (\text{A.5})$$

- Si $C_{pc} \neq C_{xu}$ la clasificación no es correcta, en este caso existe una penalización, por lo que el vector de referencia tiene la actualización indicada en la ecuación (A.6):

$$P_c(t+1) = P_c(t) - f_t \alpha(t) [X^u - P_c(t)] \quad (\text{A.6})$$

Este proceso se repite de forma iterativa con los ejemplos de entrenamiento, hasta que se haya llegado a un número máximo de iteraciones. Para evitar sobre entrenamiento se debe realizar una parada temprana del proceso de entrenamiento; para esto debe considerarse un número de neuronas entre 50 y 200 veces el número de neuronas de la red. Para establecer los vectores de referencia iniciales P_j pueden emplearse vectores x^u pertenecientes a la clase que representa la neurona, o también puede utilizarse el algoritmo k-medias.

Hay que considerar que en este modelo cada una de las neuronas representa una clase.

Apéndice B

Versiones de PSO

A continuación se describen las variantes de **PSO** que fueron utilizadas tanto para comparación como para el desarrollo del método propuesto en esta investigación, **FRvarPSO**.

B.1. PSO Básico

La técnica de optimización **PSO** fue definida en (Eberhart y Kennedy, 1995). Trabaja sobre un conjunto de soluciones de tamaño fijo denominado población. Dicha población está formada por un conjunto de partículas donde cada una de ellas representa una solución al problema. Sus movimientos se encuentran acotados en un espacio de búsqueda predefinido, por lo que las partículas no pueden desplazarse fuera de este espacio.

El algoritmo parte distribuyendo las partículas, de forma aleatoria, en el espacio de búsqueda. Luego de que la población se ha inicializado, los individuos comienzan a moverse por el espacio de búsqueda por medio de un proceso iterativo. Se identifica una partícula como la mejor de todas $gBest$ considerando el valor de la función *Fitness*. Si el nuevo fitness del individuo es el mejor encontrado hasta el momento, se actualizan los valores de mejor posición $pBest_i$ y fitness $fitness_{pBest_i}$. A continuación todas las partículas del cúmulo son atraídas por esta partícula, moviéndose también la dirección de su mejor solución descubierta hasta el momento. Todas las partículas tienen la oportunidad de conocer una mejor ubicación en el camino, transformándose en la mejor partícula $gBest$. Cuando esto ocurre las otras partículas del cúmulo cambiarán de dirección, dirigiéndose hacia la nueva ubicación.

El vector velocidad es modificado tomando en cuenta su experiencia y el entorno, con la siguiente ecuación (B.1):

$$V_i(t+1) = W.V_i(t) + (\varphi_1.rand_1.(pBest_i - X_i(t)) + \varphi_2.rand_2.(g_i - X_i(t))) \quad (B.1)$$

Donde W es el factor de inercia, φ_1 y φ_2 son las constantes de aceleración y $rand_1$

y $rand_2$ son valores aleatorios dentro de $[0,1]$ y 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 $globalBest$). La posición de la partícula se actualiza con la ecuación (B.2):

$$X_i(t + 1) = X_i(t) + V_i(t + 1) \quad (B.2)$$

El algoritmo 6 contiene el pseudocódigo del procedimiento de búsqueda del PSO básico. Es importante indicar que PSO se caracteriza por realizar muy pocas suposiciones con respecto al problema que se está resolviendo y que presenta un alto desempeño en espacios altamente dimensionales. Se trata de un algoritmo sencillo con una rápida convergencia a buenas soluciones.

Algoritmo 6: Algoritmo PSO Básico

```

begin
     $Pop = CrearPoblacion(N);$ 
    while no se alcanza 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
    
```

A partir de la idea original se han desarrollado diferentes variaciones, donde cambian diversos parámetros del algoritmo. También se han realizado cambios en la velocidad de las partículas con la finalidad de lograr diversidad, y evitar el estancamiento en el proceso de búsqueda.

B.2. PSO Binario

En la actualidad gran parte de los problemas son de naturaleza combinatoria, donde una de las formas de resolverlos es a través de la discretización de valores continuos. Cualquier problema de optimización discreto o continuo, puede ser expresado en notación binaria (Kennedy y Eberhart, 1997), por lo que se consideró de utilidad disponer de una versión binaria discreta de PSO, para problemas de optimización discretos que se 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 partícula 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.

PSO binario está formado por:

- Vector $X_i = (X_{i1}, X_{i2}, \dots, X_{in})$ este vector binario almacena la posición actual de la partícula en el espacio de búsqueda
- Vector $pBest_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{in})$ vector binario que almacena la mejor solución encontrada por la partícula hasta el momento.
- Vector Velocidad $V_i = (V_{i1}, V_{i2}, \dots, V_{in})$ vector de valores reales que almacena la dirección sobre la cual se moverá la partícula.
- $FitnessX_i$ almacena el valor de aptitud de la solución actual. Vector X_i
- $FitnesspBest_i$ almacena el valor de aptitud de la mejor solución local encontrada hasta el momento. Vector $pBest_i$

El vector velocidad está formado por las probabilidades de cambio que tiene cada bit que determina la posición de la partícula de tomar el valor 1. El vector velocidad se actualiza según (B.3)

$$V_{id}(t + 1) \leftarrow W.V_{id} + (\varphi_1.rand_1.(p_{id} - X_{id}(t)) + \varphi_2.rand_2.(g_{id} - X_{id}(t))) \quad (B.3)$$

Como puede observarse, las ecuaciones (B.1) y (B.3) coinciden pero ahora p_{id} y X_{id} son enteros entre $[0,1]$, y V_{id} , por ser una probabilidad, debe ser acotada entre $[0,1]$, para lo cual se utiliza la función sigmoide que se indica en (B.4):

$$sig(V_{id}(t)) = \frac{1}{1 + e^{-V_{id}(t)}} \quad (B.4)$$

Luego de lo cual, el vector posición de la partícula se actualiza de acuerdo a la ecuación (B.6):

$$X_{id}(t + 1) = \begin{cases} 1 & \text{if } rand() < sig(V_{id}(t + 1)) \\ 0, & \text{if not} \end{cases} \quad (B.5)$$

Donde $rand()$ es un número aleatorio con distribución uniforme en $[0,1]$, distinto para cada dimensión, el pseudo código del PSO binario se puede observar en el algoritmo 7.

Algoritmo 7: Algoritmo PSO Binario

```
begin
   $Pop = CrearPoblacion(N);$ 
  while no se alcanza 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
         $pBest_i \leftarrow (X_i);$ 
         $fitness(pBest_i) \leftarrow fitness(X_i);$ 
      for  $i = 1$  to  $size(Pop)$  do
        Elegir  $g_i$  de acuerdo al criterio de vecindario usado;
         $V_i \leftarrow W.V_i + (\varphi_1.rand_1.(pBest_i - X_i) + \varphi_2.rand_2.(g_i - X_i));$ 
        for  $d = 1$  to  $n$  do
          if  $rand() < sig(V_{id})$  then
             $X_{id} = 1$ 
          else
             $X_{id} = 0$ 
```

Result: la mejor solución encontrada

B.3. PSO de Población Variable

Los algoritmos de PSO detallados anteriormente trabajan sobre una población de tamaño fijo, el mismo que debe definirse antes de empezar la búsqueda. Esto puede constituir una desventaja ya que si el tamaño de la población que se seleccione es pequeño, van a quedar zonas del espacio de búsqueda que no van a ser analizadas, mientras que si el tamaño de la población es grande, el tiempo de convergencia se incrementará considerablemente.

En (Lanzarini, Leza y De Giusti, 2008) se propone una extensión del algoritmo PSO, denominado PSO de Población Variable *VarPSO*, incorporando los conceptos de edad y vecindad para permitir la variación del tamaño de la población. La variación del tamaño de la población se basa en una modificación del proceso adaptativo permitiendo agregar y/o eliminar 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. Para la estimación del tiempo de vida de cada individuo de la población se los agrupa según su valor de aptitud en k grupos utilizando un método de clustering competitivo del tipo winner-take-all. Como resultado se obtiene 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 $G = (g_1, g_2, \dots, g_n)$. Luego, sobre el resultado de este agrupamiento se puede asignar el tiempo de vida *fijo por grupo*, o proporcional a la *cantidad de individuos de cada clase*. 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.

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. **varPSO** divide el problema original en dos partes: en primer lugar busca determinar cuantas partículas es necesario incorporar, para luego establecer donde deben posicionarse dentro del espacio de búsqueda.

El algoritmo 8 describe el pseudo código de **varPSO**

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 B.1 y B.2. La inercia utilizada para actualizar los vectores de velocidad es ajustada según (B.6):

$$W = W_{inicial} - \frac{(W_{inicial} - W_{final})}{Total\ Iteraciones} * Iteracion\ actual \quad (B.6)$$

Donde $W_{inicial}$ es el valor inicial de W y W_{final} 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 . 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

Algoritmo 8: Algoritmo **PSO** de Población Variable

```
begin  
  Pop=CrearPoblacion(N)  
  Calcular_Tiempos_de_Vida(Pop,2)  
   $w \leftarrow$  Inercia_Maxima;  
  while no se alcanza 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 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

de la población.

Bibliografía

- Abdou, Hussein A. y John Pointon (2011). «Credit scoring, statistical techniques and evaluation criteria: a review of the literature». En: *Intelligent Systems in Accounting, Finance and Management* 18.2-3, págs. 59-88.
- Abu-Mostafa, Y. (1986). «The complexity of information extraction». En: *IEEE Transactions on Information Theory* 32.4, págs. 513-525.
- Aggarwal, Charu C. (2015). *Data Mining: The Textbook*. Springer Publishing Company, Incorporated.
- Agrawal, Rakesh, Tomasz Imieliński y Arun Swami (1993). «Mining Association Rules Between Sets of Items in Large Databases». En: *SIGMOD Rec.* 22.2, págs. 207-216.
- Agrawal, Rakesh y Ramakrishnan Srikant (1994). «Fast Algorithms for Mining Association Rules in Large Databases». En: *Proceedings of the 20th International Conference on Very Large Data Bases. VLDB '94*. Morgan Kaufmann Publishers Inc., págs. 487-499.
- Andrews, Robert, Joachim Diederich y Alan Tickle (1995). «Survey and critique of techniques for extracting rules from trained artificial neural networks». En: *Knowledge-Based Systems* 8.6, págs. 373 -389.
- Arji, Goli y col. (2019). «Fuzzy logic approach for infectious disease diagnosis: A methodical evaluation, literature and classification». En: *Biocybernetics and Biomedical Engineering* 39.4, págs. 937 -955.
- Asadi, Shahrokh y Jamal Shahrabi (2016). «ACORI: a novel ACO algorithm for rule induction». En: *Knowledge-Based Systems* 97, págs. 175 -187.
- Baesens, Bart y col. (2003). «Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation». En: *Management Science* 49.3, págs. 312-329.
- Bagheri, Ahmad, Hamed Mohammadi Peyhani y Mohsen Akbari (2014). «Financial forecasting using ANFIS networks with Quantum-behaved Particle Swarm Optimization». En: *Expert Systems with Applications* 41.14, págs. 6235 -6250.
- Barakat, Nahla y Andrew Bradley (2010). «Rule extraction from support vector machines: A review». En: *Neurocomputing* 74.1, págs. 178 -190.
- Barkana, Buket D., Inci Saricicek y Burak Yildirim (2017). «Performance analysis of descriptive statistical features in retinal vessel segmentation via fuzzy logic, ANN, SVM, and classifier fusion». En: *Knowledge-Based Systems* 118, págs. 165 -176.

- Barua, Aditi, Lalitha Snigdha y Olga. Kosheleva (2014). «Why Trapezoidal and Triangular Membership Functions Work So Well: Towards a Theoretical Explanation». En: *Journal of Uncertain Systems* 8.
- Bekhet, Hussain Ali y Shorouq Fathi Kamel Eletter (2014). «Credit risk assessment model for Jordanian commercial banks: Neural scoring approach». En: *Review of Development Finance* 42.7, págs. 20-28.
- Bennouna, Ghita y Mohamed Tkiouat (2018). «Fuzzy logic approach applied to credit scoring for microfinance in Morocco». En: *Procedia Computer Science* 127, págs. 274-283.
- Bezdek, James C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers.
- Bezdek, James C., Robert Ehrlich y William Full (1984). «FCM: The fuzzy c-means clustering algorithm». En: *Computers & Geosciences* 10.2, págs. 191 -203.
- Bingül, Zafer y Oğuzhan Karahan (2011). «A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control». En: *Expert Systems with Applications* 38.1, págs. 1017 -1031.
- Blachnik, Marcin y Włodzisław Duch (2011). «LVQ algorithm with instance weighting for generation of prototype-based rules». En: *Neural Networks* 24.8, págs. 824 -830.
- Boicescu, V. y col. (1991). *Lukasiewicz-Moisil Algebras*. Annals of Discrete Mathematics. Elsevier Science.
- Borni, A. y col. (2017). «Fuzzy logic, PSO based fuzzy logic algorithm and current controls comparative for grid-connected hybrid system». En: *AIP Conference Proceedings* 1814.1, pág. 020006.
- Bouchon, Bernadette y Gérard Cohen (1986). «Partitions and fuzziness». En: *Journal of Mathematical Analysis and Applications* 116.1, págs. 166 -183.
- Brío, Bonifacio Martín-del y Alfredo Sanz (2006). *Redes neuronales y sistemas borrosos / B. Martín del Brío, A. Sanz Molina ; pról. de Lotfi A. Zadeh*.
- Butnariu, Dan (1983). «Additive fuzzy measures and integrals I». En: *Journal of Mathematical Analysis and Applications* 93.2, págs. 436 -452.
- Carvajal, Carlos (2015). «Extracción de reglas de clasificación sobre repositorio de incidentes de seguridad informática mediante programación genética». En: *Tecnura* 19.44, págs. 109+.
- Casillas, Jorge, Oscar Cordón y Francisco Herrera (2000). «Learning Fuzzy Rules Using Ant Colony Optimization Algorithms». En: págs. 1 -9.
- Castro, Juan y col. (2007). «Extraction of fuzzy rules from support vector machines». En: *Fuzzy Sets and Systems* 158.18, págs. 2057 -2077.
- Chang, Pei-Chann y Thunshun Liao (2006). «Combining SOM and fuzzy rule base for flow time prediction in semiconductor manufacturing factory». En: *Applied Soft Computing* 6.2, págs. 198 -206.

- Chaves, Adriana, Marley Vellasco y Ricardo Tanscheit (2005). «Fuzzy Rule Extraction from Support Vector Machines». En: *Proceedings of the Fifth International Conference on Hybrid Intelligent Systems*. HIS '05. IEEE Computer Society, págs. 335-340.
- Chen, Chia-Chong (2006). «A Pso-based Method for Extracting Fuzzy Rules Directly from Numerical Data.» En: *Cybernetics and Systems* 37, págs. 707-723.
- Collotta, Mario, Giovanni Pau y Vincenzo. Maniscalco (2017). «A Fuzzy Logic Approach by Using Particle Swarm Optimization for Effective Energy Management in IWSNs». En: *IEEE Transactions on Industrial Electronics* 64.12, págs. 9496-9506.
- Crainic, Teodor Gabriel y Michel Toulouse (2003). «Parallel Strategies for Meta-Heuristics». En: *Handbook of Metaheuristics*. Ed. por Fred Glover y Gary A. Kochenberger. Springer US, págs. 475-513.
- Córdova, Jaime Fabián Díaz, Edison Coba Molina y Paúl Navarrete (2017). «Lógica difusa y el riesgo financiero. Una propuesta de clasificación de riesgo financiero al sector cooperativo». En: *Contaduría y Administración* 62.5, págs. 1670 -1686.
- Cózar, Javier, Luis de la Ossa y José Gámez (2018). «Learning compact zero-order TSK fuzzy rule-based systems for high-dimensional problems using an Apriori + local search approach». En: *Information Sciences* 433-434, págs. 1 -16.
- Daqi, Gao y col. (2016). «Pseudo-inverse linear discriminants for the improvement of overall classification accuracies». En: *Neural Networks* 81, págs. 59 -71.
- Devedzic, Vladan (2000). «Knowledge Discovery and Data Mining in Databases». En: *Handbook Softw. Eng. Knowl. Eng.* 1, págs. 615-637.
- Dong, Gang, Kin Keung Lai y Jerome Yen (2010). «Credit scorecard based on logistic regression with random coefficients». En: *Procedia Computer Science* 1.1, págs. 2463 -2468.
- Drobics, Mario y col. (2001). «Data mining using synergies between self-organizing maps and inductive learning of fuzzy rules». En: *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*. Vol. 3, 1780-1785 vol.3.
- Dua, Dheeru y Efi Karra Taniskidou (2017). *UCI Machine Learning Repository*.
- Dua, Dheeru, Karra Taniskidou y Efi. *Ultima versión del código fuente del C4.5 publicada por su autor*.
- Dubois, Didier y Henri Prade (1997). «The three semantics of fuzzy sets». En: *Fuzzy Sets and Systems* 90.2, págs. 141 -150.
- Eberhart, Russell y Joseph Kennedy (1995). «A new optimizer using particle swarm theory». En: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, págs. 39-43.
- Einipour, Amin (2011). «A Fuzzy-ACO Method for Detect Breast Cancer». En: *lobal Journal of Health Science* 3.2, págs. 195 -199.
- Elizondo, Alan y Edward I Altman (2004). «Medición integral del riesgo de crédito». En: *Editorial Limusa*.
- Fayyad, Piatetsky-Shapiro y Smyth. (1996). «From Data Mining to Knowledge Discovery in Databases». En: vol. 17, págs. 141 -150.

- Fernández, Alberto, María José del Jesus y Francisco Herrera (2009). «Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets». En: *International Journal of Approximate Reasoning* 50.3, págs. 561 -577.
- Fernández, Alberto y col. (2018). «SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary». En: *Journal of Artificial Intelligence Research* 61, págs. 863-905.
- Filiberto, Yaima y col. (2011). «Algorithm To Learn Clasification Rules Based On The Extended Rough Set Theory». En: *DYNA* 78.169, págs. 62-70.
- Frank, Eibe y Ian H Witten (1998). «Generating accurate rule sets without global optimization». En: *University of Waikato, Department of Computer Science - Computer Science Working Papers*, págs. 144 -151.
- Giles, Robin (1976). «Łukasiewicz logic and fuzzy set theory». En: *International Journal of Man-Machine Studies* 8.3, págs. 313 -327.
- González, F (2012). «Las decisiones financieras en la incertidumbre en la empresa: usando Black-Scholes como estrategia». En: *Real Academia de Ciencias Económicas y Financieras*.
- Gottwald, Siegfried. (2013). *Fuzzy Sets and Fuzzy Logic: The Foundations of Application — from a Mathematical Point of View*. Computational Intelligence. Vieweg+Teubner Verlag.
- Han, Jiawei, Jian Pei y Yiwen Yin (2000). «Mining Frequent Patterns Without Candidate Generation». En: *SIGMOD Rec.* 29.2, págs. 1-12.
- Hanyu, E y col. (2019). «Enhancements of rule-based models through refinements of Fuzzy C-Means». En: *Knowledge-Based Systems* 170, págs. 43 -60.
- Haydemar, Núñez, Angulo Cecilio y Catalá Andreu (2002). «Rule extraction from support vector machines». En: *ESANN'2002 proceedings - European Symposium on Artificial Neural Networks Bruges (Belgium)*, págs. 107-112.
- Hernandez Orallo, J., M. Ramirez Quintana y C. Ramirez (2004). *Introducción a la Minería de datos*. Pearson Prentice Hall.
- Holden, Nicholas y Alex A. Freitas (2008). «A Hybrid PSO/ACO Algorithm for Discovering Classification Rules in Data Mining». En: *Artificial Evolution and Applications*, págs. 824 -830.
- Hsu, Arthur (2000). «Automatic clustering and rule extraction using a dynamic som tree». En: *Proceedings of the 6th International Conference on Automation, Robotics, Control and Vision*.
- Hüllermeier, Eyke (2015). «Does machine learning need fuzzy logic?» En: *Fuzzy Sets and Systems* 281, págs. 292-299.
- Jang, Jyh-Shing Roger (1993). «ANFIS: adaptive-network-based fuzzy inference system». En: *IEEE Transactions on Systems, Man, and Cybernetics* 23.3, págs. 665-685.
- Jang, Jyh-Shing Roger, Chuen-Tsai Sun y Eiji. Mizutani (1997). *Neuro-Fuzzy and Soft Computing*. Prentice-Hall.
- Jimbo, Patricia, Laura Lanzarini y Aurelio Fernandez (2018a). «Extraction of Knowledge with Population-Based Metaheuristics Fuzzy Rules Applied to Credit Risk».

- En: *Advances in Swarm Intelligence*. Ed. por Ying Tan, Yuhui Shi y Qirong Tang. Cham: Springer International Publishing, págs. 153-163.
- (2018b). «Fuzzy Credit Risk Scoring Rules using FRvarPSO». En: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 26.Suppl. 1, págs. 39-57.
- (2020). «Variations of Particle Swarm Optimization for Obtaining Classification Rules Applied to Credit Risk in Financial Institutions of Ecuador». En: *Risks* 8.2.
- Jimbo, Patricia. y col. (2016). «An exploratory analysis of methods for extracting credit risk rules». En: *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*, 834–841.
- Kennedy, James (2010). «Particle Swarm Optimization». En: *Encyclopedia of Machine Learning*. Ed. por Claude Sammut y Geoffrey I. Webb. Springer US, págs. 760-766.
- Kennedy, James y Russell Eberhart (1997). «A discrete binary version of the particle swarm algorithm». En: *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*. Vol. 5, 4104-4108 vol.5.
- Kohonen, Teuvo. (1990). «The self-organizing map». En: *Proceedings of the IEEE* 78.9, págs. 1464-1480.
- Kosko, Bart Isaka y Satoru (1993). «Lógica Borrosa. Investigación y Ciencia». En: págs. 60-65.
- Kotsiantis, S. B. (2007). «Supervised Machine Learning: A Review of Classification Techniques». En: *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*. IOS Press, págs. 3-24.
- Kubat, Miroslav (1999). «Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7.» En: *The Knowledge Engineering Review* 13.4, 409–412.
- Langdon, William B. y col. (2008). «Genetic Programming: An Introduction and Tutorial, with a Survey of Techniques and Applications». En: *Computational Intelligence: A Compendium*. Ed. por John Fulcher y L. C. Jain. Berlin, Heidelberg: Springer Berlin Heidelberg, págs. 927-1028.
- Lanzarini, Laura, Victoria Leza y Armando De Giusti (2008). «Particle Swarm Optimization with Variable Population Size». En: *Artificial Intelligence and Soft Computing – ICAISC 2008*. Ed. por Leszek Rutkowski y col. Springer Berlin Heidelberg, págs. 438-449.
- Lanzarini, Laura y col. (2011). «A New Binary PSO with Velocity Control». En: *Advances in Swarm Intelligence*. Ed. por Ying Tan y col. Springer Berlin Heidelberg, págs. 111-119.
- Lanzarini, Laura y col. (2015a). «Obtaining Classification Rules Using lvqPSO». En: *Advances in Swarm and Computational Intelligence*. Ed. por Ying Tan y col. Springer International Publishing, págs. 183-193.
- Lanzarini, Laura y col. (2015b). «Obtaining Classification Rules Using LVQ+PSO: An Application to Credit Risk». En: *Scientific Methods for the Treatment of Uncertainty*

- in Social Sciences*. Ed. por Jaime Gil-Aluja y col. Springer International Publishing, págs. 383-391.
- Lanzarini, Laura y col. (2017). «Simplifying credit scoring rules using LVQ + PSO». En: *Kybernetes* 46.1, págs. 8-16.
- Lessmann, Stefan y col. (2015). «Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research». En: *European Journal of Operational Research* 247.1, págs. 124 -136.
- Li, Zhiyong y col. (2017). «Reject inference in credit scoring using Semi-supervised Support Vector Machines». En: *Expert Systems with Applications* 74, págs. 105 -114.
- Lin, Chin Teng y C. S. George Lee (2019). *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. ePub eBook.
- Luca, A. De y S. Termini (1972). «A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory». En: *Information and Control* 20.4, págs. 301 -312.
- Luna, José M., José Raúl Romero y Sebastián Ventura (2012). «Design and behavior study of a grammar-guided genetic programming algorithm for mining association rules». En: *Knowledge and Information Systems* 32.1, págs. 53-76.
- Madrid, Nicolás y Sergio Díaz (2019). «Definición de particiones difusas condicionadas usando transformadas difusas». En: *XIX Congreso Español sobre Tecnologías y Lógica Fuzzy*, págs. 286 -290.
- Mahmoudi, Nader y Ekrem Duman (2015). «Detecting credit card fraud by Modified Fisher Discriminant Analysis». En: *Expert Systems with Applications* 42.5, págs. 2510 -2516.
- Malone, James y col. (mar. de 2006). «Data mining using rule extraction from Kohonen self-organising maps». En: *Neural Computing & Applications* 15.1, págs. 9-17.
- Mamdani, Ebrahim (1974). «Application of fuzzy algorithms for control of simple dynamic plant». En: *Proceedings of the Institution of Electrical Engineers* 121.12, págs. 1585-1588.
- Mamdani, Ebrahim., Jacob Østergaard y E. Lembessis (1983). «Use of Fuzzy Logic for Implementing Rule-Based Control of Industrial Processes». En: *Advances in Fuzzy Sets, Possibility Theory, and Applications*. Ed. por Paul P. Wang. Springer US, págs. 307-323.
- Mammadli, Sadig (2016). «Fuzzy Logic Based Loan Evaluation System». En: *Procedia Computer Science* 102, págs. 495 -499.
- Medina Hurtado Santiago Paniagua Gómez, Gisela (2008). «Modelo de inferencia difuso para estudio de crédito». En: *Dyna*.
- Melin, Patricia y col. (2013). «Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic». En: *Expert Systems with Applications* 40.8, págs. 3196 -3206.
- Nie, Guangli y col. (2011). «Credit card churn forecasting by logistic regression and decision tree». En: *Expert Systems with Applications* 38.12, págs. 15273 -15285.

- Omkar, S. y Raghavendra Karanth (2008). «Rule extraction for classification of acoustic emission signals using Ant Colony Optimisation». En: *Engineering Applications of Artificial Intelligence* 21.8, págs. 1381 -1388.
- Oreski, Stjepan y Goran Oreski (2014). «Genetic algorithm-based heuristic for feature selection in credit risk assessment». En: *Expert Systems with Applications* 41.4, Part 2, págs. 2052 -2064.
- Passino, Kevin y Stephen Yurkovich (1998). *Fuzzy Control*. Addison Wesley.
- Pateritsas, Christos. y Stylianos Modes (2000). «Extracting rules from trained self-organizing maps». En: *IADIS International Conference Applied Computing 2007*, págs. 183 -190.
- Pérez, José Muñoz. (2010). *Inteligencia computacional inspirada en la vida*. Lecciones Inaugurales. Servicio de Publicaciones e Intercambio Científico de la Universidad de Málaga.
- Quinlan, J. Ross. (1986). «Induction of decision trees». En: *Machine Learning* 1.1, págs. 81-106.
- Quinlan, J. Ross (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc.
- Reeves, Colin R., ed. (1993). *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, Inc.
- Reinke, Jens (1998). «How to lend like mad and make a profit: A micro-credit paradigm versus the start-up fund in South Africa». En: *The Journal of Development Studies* 34.3, págs. 44-61.
- Reyes, J., A. Morales-Esteban y F. Martínez-Álvarez (2013). «Neural networks to predict earthquakes in Chile». En: *Applied Soft Computing* 13.2, págs. 1314 -1328.
- Ritter, H., T. Martinetz. y K. Schulten (1993). *C4.5: programs for machine learning*. Addison-Wesley, New York, Morgan Kaufmann Publishers.
- Ruspini, Enrique H. (1969). «A new approach to clustering». En: *Information and Control* 15.1, págs. 22 -32.
- Saavedra, Maria y Máximo Saavedra (2010). «Modelos para medir el riesgo de crédito de la banca». En: 23.40, págs. 295 -319.
- Seguros, Ecuador Superintendencia de bancos y (2017). «Modelos para medir el riesgo de crédito de la banca». En: Resolución No JB-2003-602, págs. 818 -852.
- Setiono, Rudy, Bart Baesens y Christophe Mues (2008). «Recursive Neural Network Rule Extraction for Data With Mixed Attributes». En: *IEEE Transactions on Neural Networks* 19.2, págs. 299-307.
- Shen, Feng y col. (2018). «An extended intuitionistic fuzzy TOPSIS method based on a new distance measure with an application to credit risk evaluation». En: *Information Sciences* 428, págs. 105 -119.
- Shi, Maolin y col. (2019). «A fuzzy c-means algorithm guided by attribute correlations and its application in the big data analysis of tunnel boring machine». En: *Knowledge-Based Systems* 182, pág. 104859.
- Shihabudheen, K.V. y Gopinatha Pillai (2018). «Recent advances in neuro-fuzzy system: A survey». En: *Knowledge-Based Systems* 152, págs. 136 -162.

- Sik, Yong y Bongju Lee (2013). «THE ONE-SIDED QUADRANGULAR FUZZY SETS». En: 26.2, págs. 297-308.
- Simpson, Patrick K. (1992). «Fuzzy min-max neural networks. I. Classification». En: *IEEE Transactions on Neural Networks* 3.5, págs. 776-786.
- Sit, Emil y Robert Morris (2002). «Security Considerations for Peer-to-Peer Distributed Hash Tables». En: *Peer-to-Peer Systems*. Ed. por Peter Druschel, Frans Kaashoek y Antony Rowstron. Berlin, Heidelberg: Springer Berlin Heidelberg, págs. 261-269.
- Smith, Stephen Frederick. (1980a). «A learning system based on genetic adaptive algorithms». En: *PhD thesis*.
- Smith, Stephen Frederick (1980b). «A Learning System Based on Genetic Adaptive Algorithms». En:
- Soui, Makram y col. (2019). «Rule-based credit risk assessment model using multi-objective evolutionary algorithms». En: *Expert Systems with Applications* 126, págs. 144-157.
- Starczewski, Janusz T. (2009). «Efficient triangular type-2 fuzzy logic systems». En: *International Journal of Approximate Reasoning* 50.5, págs. 799-811.
- Straccia, Umberto (2014). *Foundations of Fuzzy Logic and Semantic Web Languages*. G. Q Zhang.
- Sugeno, Michio y Takahiro Yasukawa (1993). «A Fuzzy-Logic-Based Approach to Qualitative Modeling». En: *IEEE Transactions on Fuzzy Systems* 1.1, págs. 7-31.
- Tan, Pang-Ning, Michael Steinbach y Vipin Kumar (2005). *Introduction to Data Mining*. Addison Wesley.
- Teles, Gernmanno y col. (2019). «Machine learning and decision support system on credit scoring». En: *Neural Computing and Applications*.
- Terano, Toshiro, Kiyoji Asai y Michio Sugeno (1992). *Fuzzy Systems Theory and Its Applications*. Academic Press Professional, Inc.
- Thomas, Lyn C. (2000). «A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers». En: *International Journal of Forecasting* 16.2, págs. 149-172.
- Toivonen, Hannu (1996). «Sampling large databases for association rules». En: *Proceedings of the 22nd VLDB Conference, India*, págs. 3212-3218.
- Tsukamoto, Yahachiro (1993). «AN APPROACH TO FUZZY REASONING METHOD». En: *Readings in Fuzzy Sets for Intelligent Systems*. Ed. por Didier Dubois, Henri Prade y Ronald R. Yager. Morgan Kaufmann, págs. 523-529.
- Vaidyanathan, Sundarapandian y Ahmad Taher Azar (2016). «Takagi-Sugeno fuzzy logic controller for Liu-Chen four-scroll chaotic system». En: *International Journal of Intelligent Engineering Informatics* 4.2, págs. 135-150.
- Venturini, Gilles (1993). «SIA: A supervised inductive algorithm with genetic search for learning attributes based concepts». En: *Machine Learning: ECML-93*. Ed. por Pavel B. Brazdil. Springer Berlin Heidelberg, págs. 280-296.

- Wang, Gang y Jian Ma (2012). «A hybrid ensemble approach for enterprise credit risk assessment based on Support Vector Machine». En: *Expert Systems with Applications* 39.5, págs. 5325 -5331.
- Wang, Hong y col. (1995). «Advanced Adaptive Control».
- Witten, Ian y col. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier Science.
- Wong, Ching-Chang. y Chen Chia-Chong (2000). «A GA-based method for constructing fuzzy systems directly from numerical data». En: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 30.6, págs. 904-911.
- Wong, Shaw y Abdel Hamouda (2000). «Optimization of fuzzy rules design using genetic algorithm». En: *Advances in Engineering Software* 31.4, págs. 251 -262.
- Xu, Xin y col. (2016). «Forecasting tourism demand by extracting fuzzy Takagi-Sugeno rules from trained SVMs». En: *CAAI Transactions on Intelligence Technology* 1.1, págs. 30 -42.
- Yager, Ronald y Lotfi A. Zadeh (2012). *An Introduction to Fuzzy Logic Applications in Intelligent Systems*. The Springer International Series in Engineering and Computer Science. Springer US.
- Yingjie, Zhu y col. (2015). «Control of response reliability by parvalbumin-expressing interneurons in visual cortex». En: *Nature Communications* 6.1.
- Yun, Yong Sik y Lee Bongju (2013). «THE ONE-SIDED QUADRANGULAR FUZZY SETS». En: *Journal of the Chungcheong Mathematical Society* 26.2, págs. 297-308.
- Zadeh, Lotfi A. (1965). «Fuzzy sets». En: *Information and Control* 8.3, págs. 338 -353.
- Zeller, Manfred (1998). «Determinants of Repayment Performance in Credit Groups: The Role of Program Design, Intragroup Risk Pooling, and Social Cohesion». En: *Economic Development and Cultural Change* 46.3, págs. 599-620.
- Zhao, Zongyuan y col. (2015). «Investigation and improvement of multi-layer perceptron neural networks for credit scoring». En: *Expert Systems with Applications* 42.7, págs. 3508 -3516.
- Zhu, Quanmin y Ahmad Azar (2014). *Complex System Modelling and Control Through Intelligent Soft Computations*. Studies in Fuzziness and Soft Computing. Springer International Publishing.
- Zięba, Maciej, Sebastian Tomczak y Jakub Tomczak (2016). «Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction». En: *Expert Systems with Applications* 58, págs. 93 -101.