



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

UNIDAD ACADÉMICA PROFESIONAL TIANGUISTENCO

“ Generación de prototipos usando un algoritmo genético ”

Tesis
para obtener el grado de
Maestro en Ciencias de la Computación

Que Presenta:

Ing. Iván Hernández Martínez

Tutor Académico:
Dr. René Arnulfo García Hernández

Tutores adjuntos:
Dra. Yulia Nikolaevna Ledeneva
Dr. Ángel Hernandez Castañeda

Resumen

La clasificación es una tarea del reconocimiento de patrones que permite asignar una clase a un nuevo objeto o patrón. Para asignar la clase al patrón los algoritmos de clasificación deben generar un modelo de clasificación. El modelo de clasificación se genera con base en una muestra de datos. Sin embargo, algunos algoritmos de clasificación no generan un modelo de clasificación para generalizar la información de la muestra de datos. Los algoritmos que pertenecen a este enfoque tienen el nombre de algoritmos de aprendizaje perezoso. La regla del vecino más cercano o por sus siglas en inglés *Nearest Neighbor* (NN) pertenece a este tipo de algoritmos. El algoritmo NN usa toda la muestra de datos para clasificar nuevos patrones. No obstante, el usar toda la muestra de datos genera varios problemas por el tamaño de la muestra de datos y patrones ruidosos.

Para tratar de solucionar estos problemas se han propuesto métodos de selección de prototipos y Generación de Prototipos (GP). La selección de prototipos trata de encontrar un subconjunto de patrones representativos de la muestra de datos. Mientras, la GP además de seleccionar patrones de la muestra de datos generara nuevos patrones. La GP obtiene mejores resultados que la selección de prototipos, porque puede que dentro de la muestra de datos no se tengan los patrones representativos para generar la muestra de prototipos. Estos métodos tratan de reducir el número de patrones de la muestra de datos tratando de mantener la clasificación o mejorarla para nuevos patrones.

Para mantener o mejorar la clasificación de nuevos patrones se han propuesto diferentes métodos de GP. Los primeros métodos de GP se enfocaban en la fusión de patrones cercanos y con la misma clase. Estos métodos no requerían de una selección de patrones representativos de la muestra de datos. En cambio, otros métodos de GP requieren de una selección de una submuestra prototipos representativos. La submuestra de prototipos representativa es movida dentro del espacio de características para posicionarla en el mejor lugar. También, se tienen métodos de GP que no requieren de una selección de prototipos y no fusionan patrones de la ME. Estos métodos de GP usan enfoques evolutivos para generar y posicionar los prototipos dentro de la muestra de datos. Sin embargo, los métodos de GP evolutivos no usan los Algoritmos Genéticos (AG) y solo se han usado para la selección de prototipos representativos.

El método propuesto busca demostrar que un AG puede generar una submuestra de prototipos representativa con una métrica que evalúa los prototipos durante la generación para conocer la capacidad del método propuesto. Para realizar la generación de prototipos se usan 14 muestra de datos de diferentes tamaños, número de clases y número de características. Además, los resultados del método propuesto se comparan con 18 métodos de GP del estado del arte usando la métrica de exactitud. La comparación de los resultados obtenidos por el método propuesto revela que las submuestras de prototipos obtenidas en algunos casos mantienen y en otros casos mejoran la clasificación de nuevos patrones. En esta tesis se demuestra la capacidad del método propuesto para generar prototipos representativos.

Contenido

Página

LISTA DE FIGURAS.....	I
LISTA DE TABLAS.....	II
CAPÍTULO 1. INTRODUCCIÓN	3
1.1 Planteamiento del problema	8
1.2 Hipótesis.....	9
1.3. Objetivo general.....	9
1.3.1 Objetivos específicos	9
1.4 Justificación.....	9
1.5 Delimitación del problema.....	10
1.6 Estructura de la tesis	10
CAPÍTULO 2. MARCO TEÓRICO.....	11
2.1 Reconocimiento de objetos.	11
2.1.1 Reconocimiento de patrones.....	12
2.1.2 Clases	12
2.1.3 Patrones	12
2.1.4 Clasificación	14
2.1.5 Tipos de aprendizaje automático	15
2.1.6 Validación del modelo de clasificación	16
2.1.7 Sobreajuste.....	16
2.2 Clasificación con la regla del vecino más cercano (NN).....	16
2.2.1 Clasificación de nuevo patrones	17
2.2.2. Problema del clasificador NN.....	18
2.3 Métricas de evaluación.....	18
2.4 Validación cruzada	20
2.5 Optimización	21
2.5.1 Algoritmos genéticos	21
a) Codificación y población inicial.....	22
b) Función de aptitud	23
c) Condición de paro.....	23
d) Operadores de selección	24
e) Operadores genéticos.....	25
f) Mutación	26
2.6 Resumen.....	27
CAPÍTULO 3. ESTADO DEL ARTE.....	29

3.1 Descripción general de los métodos de GP.....	30
3.1.1 Inicialización.....	30
a) Selección aleatoria de un prototipo por clase.....	30
b) Selección de un centroide por clase como prototipo inicial.....	31
c) Selección de una submuestra de prototipos	33
3.1.2 Generación y optimización de los prototipos	33
a) Generación de prototipos.....	34
b) Optimización de prototipos.....	34
c) Generación y optimización de los prototipos.....	35
3.1.3 Validación de los prototipos.....	36
3.2 Ajuste de la posición de los prototipos con modelos vectoriales	37
3.2.1 <i>Learning Vector Quantization</i> (LVQ).....	37
a) LVQ1	38
b) LVQ2	39
c) LVQ3	39
3.2.2 Decision Surface Mapping (DSM)	40
3.2.4 Hybrid LVQ3 Algorithm (HYB)	41
3.3 Ajuste de la posición y generación de los prototipos con modelos vectoriales	41
3.3.1 <i>Learning Vector Quantization with Training Count</i> (LVQTC).....	41
3.3.2 <i>Learning Vector Quantization with Pruning</i> (LVQPRU)	42
3.4 Vector Quantization (VQ).....	42
3.4.1 Vector Quantization for Nearest Neighbor (VQ-NN).....	43
3.4.2 A Vector Quantization method for nearest neighbor classifier (AVQ)	43
3.5 Combinación de patrones de la ME para la generación de nuevos prototipos.....	43
3.5.1 Reduction by Space Partitioning (RSP3)	43
3.5.2 Bootstrap Technique (BTS3).....	44
3.5.3 MixtGauss.....	44
3.6 División de la ME para la generación de prototipos.....	45
3.6.1 Pairwise Opposite Class-Nearest Neighbor (POC-NN).....	45
3.6.2 Self-Generating Prototypes (SGP).....	45
3.7 Generación y optimización de prototipos con modelos evolutivos	46
3.7.1 Evolutionary Design of Nearest Prototype Classifiers (ENPC)	46
3.7.2 Iterative Prototype Adjustment based on Differential Evolution (IPADE)	47
3.8 Ajuste de la posición de los prototipos con modelos evolutivos.....	48
3.8.1 MSE.....	49
3.8.2 An Adaptive Michigan Approach PSO (AMPSO)	49
3.8.3 Particle Swarm Optimization for prototype reduction (PSO)	50
3.9 Resumen.....	51
CAPÍTULO 4. MÉTODO PROPUESTO.....	56

4.1 Descripción del método propuesto	57
4.1.1 Normalización.....	57
4.1.2 Generación de las <i>ME</i> y <i>MP</i>	57
4.2 Pasos del método propuesto	57
4.2.3 Generación de los prototipos	58
a) Inicialización.....	59
b) Optimización global	61
c) Inyección.....	62
d) Optimización global 2.....	63
e) Optimización local.....	64
f) Eliminación	66
g) Condición de paro del método	67
4.2.4 Evaluación de los prototipos con la <i>MP</i>	67
4.2.5 Evaluación de los resultados.....	69
4.3 Resumen.....	70
CAPÍTULO 5. EXPERIMENTACIÓN.....	71
5.1 Preprocesamiento de las muestras de datos.....	72
5.2 Generación de los prototipos.....	74
5.3 Resultados	75
5.3.1 Comportamiento de la aptitud con los parámetros del método propuesto	75
5.3.2 Resultados obtenidos del método propuesto	76
5.3.3 Evaluación de los resultados obtenidos.....	77
a) Evaluación de la exactitud en la <i>ME</i>	77
b) Evaluación de la exactitud en la <i>MP</i>	78
c) Reducción obtenida	79
5.3.4 Evaluación de los resultados obtenidos.....	80
5.3.5 Resultados de <i>ME</i> con menos de 500 patrones.....	81
5.3.6 Resultados de <i>ME</i> con más de 2,000 patrones	82
5.4 Resumen.....	83
CAPÍTULO 6. CONCLUSIONES	85
6.1 Aportaciones.....	86
6.2 Trabajo Futuro.....	87
REFERENCIAS BIBLIOGRÁFICAS	88

Lista de figuras

Figura 2.1 Patrones graficados en un plano de dos dimensiones.	14
Figura 2.2 División de dos clases con una línea recta.	15
Figura 2.3 Validación Cruzada.	21
Figura 2.4 Diagrama de flujo del algoritmo genético.	22
Figura 2.5 Codificación de un cromosoma de diez posiciones.	23
Figura 2.6 Cruza uniforme.	25
Figura 2.7 Representación de la mutación por inversión.	26
Figura 2.8 Representación de la mutación por inversión base- n	27
Figura 3.1 Representación gráfica de la muestra de datos de iris con dos características.	30
Figura 3.2 Selección aleatoria de prototipos iniciales por clase.	31
Figura 3.3 Prototipos iniciales usando el algoritmo de agrupamiento <i>k-means</i>	32
Figura 3.4 Prototipos iniciales usando información los patrones seleccionados.	33
Figura 3.5 Selección de una submuestra de prototipos.	33
Figura 3.6 Generación de nuevos prototipos con la información.	34
Figura 3.7 Optimización de la submuestra de prototipos en el espacio de características.	35
Figura 3.8 Generación y optimización de los prototipos.	36
Figura 3.9 Representación de la red LVQ con 2 clases.	38
Figura 3.10 Representación de los prototipos de ENPC.	47
Figura 3.11 Clasificación propuesta de los métodos de GP.	52
Figura 3.12 Mapa generación de prototipos propuesto por (Triguero et al., 2012).	53
Figura 4.1 Diagrama del método propuesto.	58
Figura 4.2 Diagrama de procesos de generación de prototipos usando la ME.	59
Figura 4.3 Representación gráfica de la muestra de datos “banana”.	60
Figura 4.4 Representación gráfica de los prototipos iniciales.	60
Figura 4.5 Representación gráfica de los prototipos después la primera optimización global.	62
Figura 4.6 Patrones mal clasificados de la ME inyectados en los prototipos.	63
Figura 4.7 Representación gráfica de los prototipos después la optimización global 2.	64
Figura 4.8 Posición inicial y final de los prototipos de la SMP al finalizar la optimización local.	65
Figura 4.9 Prototipos que forman la SMP y el prototipo a eliminar.	67
Figura 4.10 Resultados de la precisión obtenida de los prototipos y el clasificador KNN.	69
Figura 5.1 Herramienta KEEL Suite 3.0.	73
Figura 5.2 Aptitud del AG global con los parámetros del método propuestos.	75

Lista de tablas

Tabla 2.1 Patrones con dos características y su clase.	13
Tabla 2.2 Matriz de confusión para dos clases.	19
Tabla 4.1 Prototipos iniciales aleatorios de la ME.....	59
Tabla 4.2 Resultados obtenidos de la ME “banana”.....	68
Tabla 5.1 Muestra de datos usadas para la GP.....	72
Tabla 5.2 Métodos del estado del arte elegidos.	73
Tabla 5.3 Resultados obtenidos por el método propuesto y la métrica MfP	76
Tabla 5.4 Resultados de los métodos de GP del estado del arte.	77
Tabla 5.5 Resultados de exactitud para las ME ordenados de mayor a menor.....	78
Tabla 5.6 Resultados de exactitud para las MP ordenados de mayor a menor.....	79
Tabla 5.7 Porcentaje de reducción de la ME ordenados de mayor a menor.....	80
Tabla 5.8 Ranking de los métodos de GP.....	81
Tabla 5.9 Resultados obtenidos en la muestra de datos de apendicitis.	82
Tabla 5.10 Resultados obtenidos en la muestra de datos de banana.	83



Capítulo 1.

Introducción

Desde la creación de la lengua escrita, los humanos han registrado observaciones para monitorear el movimiento de los rebaños de animales y la alineación de los planetas y las estrellas. También, en las ciudades se registró la escritura en el pago de impuestos, actas de nacimiento y defunción (Lantz, 2015). Actualmente, este proceso se ha automatizado y tiene la capacidad de registrar, almacenar y procesar grandes cantidades de datos (Alpaydin, 2010). Los datos almacenados contienen información oculta que puede ser útil (Suthaharan, 2016). Sin embargo, clasificar, comprender y comprimir los datos de esta información es una tarea difícil (Kim & Oommen, 2003a).

Introducción

La disciplina encargada del diseño y desarrollo de métodos para el aprendizaje de datos tiene el nombre de reconocimiento de patrones (RP) (Theodoridis & Koutroumbas, 2003). El RP se describe como el descubrimiento automático de regularidades en los datos (Bishop, 2006). Las regularidades se obtienen de los datos al aplicar métodos que los describen y clasifican (Marques De Sa, 2001). En la clasificación, los datos son asignados a diferentes categorías o clases. Una clase se define como un conjunto de objetos que son similares, sin embargo, no son idénticos, y se distingue de otras clases (Theodoridis & Koutroumbas, 2003), (Dougherty, 2013).

El término patrón en RP se usa para referirse a los objetos o datos (Marques De Sa, 2001) que van a ser usados por los algoritmos de aprendizaje como, manzanas, naranjas, ondas de voz o huellas digitales (Dougherty, 2013). Los patrones deben tener diferencias medibles para realizar la clasificación, por ejemplo para clasificar pescado se pueden usar sus características físicas como: longitud, ligereza, ancho, posición de la boca, etc. (Duda et al., 2000). Para realizar la clasificación automática con algoritmos de aprendizaje la muestra de datos se debe dividir para formar una Muestra de Entrenamiento (ME) y una Muestra de Prueba (MP) (Triguero et al., 2012).

Para clasificar nuevos patrones los algoritmos de aprendizaje forman un modelo de clasificación en el entrenamiento a partir de la ME (Sánchez et al., 2003). Sin embargo, hay métodos de clasificación que no requieren que el algoritmo genere un modelo de clasificación (Triguero et al., 2010). Los algoritmos que pertenecen a este enfoque de clasificación tienen el nombre "algoritmos de aprendizaje perezosos", y la regla del vecino más cercano o por sus siglas en inglés *Nearest Neighbor* (NN) pertenece a este tipo de algoritmos (Cervantes et al., 2007). La regla de clasificación NN pertenece a la clasificación supervisada, fue propuesta por Cover y Hart en 1966 (Garain, 2008) y ha demostrado ser eficaz como algoritmo de clasificación en muchas aplicaciones de reconocimiento de patrones (Raicharoen & Lursinsap, 2005). Además, la regla NN es considerada uno de los 10 mejores algoritmos de reconocimiento de patrones (Triguero et al., 2010), porque es uno de los métodos más simples y fáciles de implementar para la clasificación no paramétrica (Yen et al., 2004). Por tal motivo es uno de los algoritmos más estudiados (Sánchez, 2004).

Introducción

La regla NN requiere almacenar toda la ME en memoria (Triguero et al., 2010) (Raicharoen & Lursinsap, 2005) porque requiere de todos los patrones para adquirir información estadística para clasificar un nuevo patrón (Yen et al., 2004). En cada clasificación, la regla NN debe comparar toda la ME con el nuevo patrón, y esto demanda requisitos computacionales altos en ME grandes (Yen et al., 2004). La comparación se realiza porque la regla NN plantea la hipótesis de proximidad en el espacio de características y, generalmente esto expresa la pertenencia a la misma clase (Decaestecker, 1997). Para determinar la proximidad de dos patrones o más patrones se usan medidas de similitud (Triguero et al., 2010). Los patrones que son similares son asignados a la misma clase (Fayed et al., 2007). Por lo que en ME grandes tiene problemas de complejidad computacional por el gran número de cálculos de distancia que se realizan (Kim & Oommen, 2003b). Además, las muestras de datos pueden contener ruido (Triguero et al., 2010) que afecta negativamente el proceso de clasificación (Sánchez, 2004).

La regla NN es bastante sensible a los patrones ruidosos y atípicos (Raicharoen & Lursinsap, 2005), (Garain, 2008). Una posible solución a los problemas anteriores es reducir los patrones de la ME (Xie et al., 1993) y tratar de obtener una precisión igual o mayor que la ME original (Bezdek & Kuncheva, 2001), (Kim & Oommen, 2003b). Al reducir el número de patrones de la ME se reduce la complejidad computacional, requisitos de almacenamiento y el ruido (Triguero et al., 2010). En la literatura se han propuesto y desarrollado métodos que reducen el número de patrones sin comprometer y afectar la precisión de clasificación (Cervantes et al., 2007), (Yen et al., 2004). Generalmente estos métodos se hacen en conjunto con la regla NN (Nanni & Lumini, 2009).

La reducción de patrones se ha investigado en dos categorías: selección de prototipos y Generación de Prototipos (GP) (Nanni & Lumini, 2009), (Triguero et al., 2010), (Cervantes et al., 2007). Los nuevos patrones generados o seleccionados tienen el nombre de prototipos, por lo que un prototipo es análogo a un patrón (Cervantes et al., 2009). Los prototipos deben representar eficientemente las distribuciones de las clases (Lozano et al., 2006). Los métodos de selección de prototipos solo se enfocan en la identificación (Nanni & Lumini, 2009) y selección de patrones de la ME (Sánchez, 2004) para formar un submuestra representativa (Triguero et al., 2010). Sin embargo, la selección de prototipos y la regla NN asumen que pueden encontrar patrones representativos en la ME. No obstante, puede que no se tenga

Introducción

ningún patrón ubicado en los puntos precisos que construirán un algoritmo de aprendizaje más preciso (Sánchez, 2004).

Los métodos de GP pueden iniciar con la selección inicial de una submuestra de prototipos de la ME (Nanni & Lumini, 2009), pero también pueden generar nuevos prototipos en donde se requieran, si las ubicaciones se determinan con precisión (Sánchez, 2004). Los prototipos generados tratarán de resumir las características representativas de patrones similares (Lam et al., 2002). En general, si los prototipos son generados correctamente se puede obtener una mejora, pero, no se garantiza la mejora de clasificación en patrones nuevos (Garain, 2008).

Para la generación de prototipos se define los siguientes elementos:

- La Muestra de Entrenamiento (ME), donde los patrones de $ME_t = (me_{t1}, me_{t2}, \dots, me_{tn})$ tienen un espacio m -dimensional de características $me_{ti} = (me_{ti1}, me_{ti2}, \dots, me_{tis}, me_{tiw})$, donde w es la clase a la que me_{ti} pertenece de un número posible de clases de $\Omega = (w_1, w_2, \dots, w_n)$ definidas en la ME (Lozano et al., 2006).
- La Muestra de Prueba (MP), donde los patrones de $MP_p = (mp_{p1}, mp_{p2}, \dots, mp_{pn})$ con un espacio m -dimensional de características $mp_{pn} = (mp_{pn1}, mp_{pn2}, \dots, mp_{pne}, mp_{pnd})$, en donde d no se conoce y se puede asignar a una posible clase de Ω .
- Los prototipos (PR), donde los prototipos de $PR_r = (pr_{r1}, pr_{r2}, \dots, pr_{rn})$ se generan a partir de ME_t , por lo que se espera que represente correctamente el espacio m -dimensional de características y las clases para realizar la clasificación de la ME y la MP (Yen et al., 2004), (Triguero et al., 2010).

El propósito de los métodos de GP es generar n prototipos donde $PR_r < ME_t$.

En el estado del arte se han propuesto diferentes métodos para la GP. Uno de los primeros métodos de GP es el método PNN (Chang, 1974) que está basado en la fusión de los patrones de la misma clase. Los prototipos se generan de la fusión de un patrón A con uno B de la ME que pertenezcan a la misma clase. La fusión de los patrones no debe afectar la clasificación (Chang, 1974) (Triguero et al., 2010). Se tienen otros métodos de GP que usan la técnica de

Introducción

divide y vencerás (Raicharoen & Lursinsap, 2005). La ME se divide en dos o más subespacios para simplificarla en cada división (Triguero et al., 2010). Los métodos basados en este enfoque son el método POC-NN (Raicharoen & Lursinsap, 2005) y el método SGP (Fayed et al., 2007). Estos métodos inician con solo un prototipo por clase y cada que genera una división en la ME crean uno o más prototipos con la misma clase que sus vecinos más cercanos. También, se tienen métodos como BTS3 (Hamamoto et al., 1997) y ICPL (Lam et al., 2002) que generan los prototipos de la fusión o agrupamiento de los patrones de la ME. Los métodos SPG, BTS3 y ICPL son sensibles a patrones con características similares y una clase diferente (Raicharoen & Lursinsap, 2005).

Otro enfoque para la GP se basa en la cuantificación vectorial o *Vector Quantization* (VQ), que se ha usado durante muchos años para generar prototipos (Bezdek & Kuncheva, 2001). Los métodos VQ representan con un número reducido de prototipos todos los patrones de la ME (Xie et al., 1993). Los métodos que usan VQ crean nuevos prototipos basados en la información obtenida de los patrones de la ME (Yen et al., 2004). Los métodos que usan VQ deben calcular el número de prototipos a generar para cada clase en las muestras de datos (Yen et al., 2004). Los enfoques de VQ que requieren una selección de patrones iniciales de la ME es *Learning Vector Quantization* (LVQ) (Geva & Sitte, 1991a). El método LVQ es una red neuronal donde cada neurona representa un prototipo y cada conexión representa una característica del prototipo (Kim & Oommen, 2003b). El método LVQ se entrena con la ME para ajustar y encontrar la mejor posición de cada prototipo dentro del espacio de características (Triguero & Herrera, 2012). LVQ es uno de los enfoques que presenta una gran cantidad de variantes en el estado del arte (Kim & Oommen, 2003a). Los métodos que hacen uso de LVQ requieren de una buena selección de prototipos iniciales por clase.

Los enfoques de GP evolutivos no requieren de una selección inicial representativa de prototipos. Los prototipos iniciales son seleccionados aleatoriamente de la ME, como en los métodos de ENPC (Fernández & Isasi, 2004) y PSCSA (Garain, 2008). En cada iteración estos métodos aplican a los prototipos iniciales operadores de reproducción, mutación y eliminación para generar más prototipos. Otros métodos evolutivos como AMPSO (Cervantes et al., 2007) y PSO (Nanni & Lumini, 2009) inician con una submuestra aleatoria de un cinco por ciento o dos por ciento de prototipos de la ME. La submuestra es movida dentro del espacio de características haciendo uso del algoritmo de enjambre de partículas (Fernández & Isasi, 2004),

Introducción

(Garain, 2008). La clase de los prototipos iniciales no se modifica en los métodos PSO. Además, existen otros algoritmos evolutivos como los Algoritmos Genéticos (AG); que solo se han usado para la selección de los prototipos iniciales (Nanni & Lumini, 2009).

Los métodos de GP presentados hacen uso de diferentes enfoques para la GP, usan reglas específicas para la creación y ajuste del prototipo dentro del espacio de características. Los prototipos obtenidos forman el modelo de clasificación. El modelo de clasificación se genera con tres diferentes orientaciones:

- Fusión o agrupación de patrones para generar prototipos.
- Cálculo del número y posición de prototipos con modelos VQ.
- Generación de prototipos con modelos evolutivos.

Además, solo se usa la regla de clasificación NN y una métrica de evaluación. La métrica utilizada evalúa la reducción de error (Cervantes et al., 2007), la mejora de precisión (Fernández & Isasi, 2004) o la calidad de los prototipos (Triguero et al., 2010). Sin embargo, los métodos de GP no permiten hacer uso de diferentes métricas para un mismo método de GP. Esto limita la aplicación de los métodos de GP para diferentes problemas de clasificación. Por lo que se requiere de un método de GP que permita usar diferentes métricas y reglas para la generación de prototipos.

1.1 Planteamiento del problema

Los trabajos encontrados en el estado del arte solo se han enfocado en iniciar, generar y validar de prototipos usando una métrica y reglas específicas para la generación de prototipos, es decir, no se tiene un método de GP que permita usar diferentes métricas y reglas para generar los prototipos. Por lo que se tiene el problema de cómo obtener prototipos útiles a partir de una ME.

Por lo que se tiene la siguiente pregunta de investigación:

¿Cómo obtener prototipos útiles a partir de una ME, que se puedan usar para clasificar nuevos patrones?

1.2 Hipótesis

Los métodos de GP del estado del arte hacen uso de diferentes reglas para la generación de los prototipos. Sin embargo, todos comparten tres procesos generales de inicialización, generación y validación. En la inicialización se debe seleccionar una submuestra de n -prototipos de la ME. Esta submuestra se usará en el proceso de generación y validación de prototipos con ayuda de la ME.

Si al simplificar y aplicar los procesos de inicialización, generación y validación con base en una ME, entonces es posible obtener prototipos útiles, con el fin de clasificar nuevos patrones en tareas de clasificación usando la regla NN.

1.3. Objetivo general

Simplificar y construir un método de GP para iniciar, generar y validar los prototipos para clasificar nuevos patrones usando la regla NN.

1.3.1 Objetivos específicos

- Investigar los métodos y resultados que obtienen los métodos de generación de prototipos.
- Conocer las muestras de datos que se han usado para la generación prototipos.
- Desarrollar el método para la generación de prototipos.
- Generar los prototipos usando el método propuesto
- Evaluar la calidad de los prototipos generados.
- Comparar los resultados obtenidos del método propuesto con los métodos de generación de prototipos investigados.

1.4 Justificación

Los métodos de GP obtienen una reducción en el número de patrones de la ME y mantienen una precisión similar en la clasificación de la MP. Sin embargo, estos métodos de GP solo usan una métrica de evaluación. Además, la asignación de prototipos por clases depende de los patrones iniciales o de la región que ocupa en la ME, limitando la adaptación de los prototipos obtenidos.

1.5 Delimitación del problema

La tesis presenta un método de GP para la generación, optimización y validación de prototipos usando la regla NN. Las muestras de datos usadas para la GP se obtienen de la herramienta *KeelPS* usada en (García et al., 2012) y (Triguero et al., 2012). El método tiene la capacidad para aplicarse en diferentes tareas de RP que hagan uso de muestras de datos. Sin embargo, en la presente tesis solo se usa para la inicialización, generación y validación de prototipos en muestras de datos de diferentes tamaños y dimensiones. El uso del método en otras tareas de RP queda fuera del alcance del problema.

1.6 Estructura de la tesis

En este capítulo se presenta la introducción al problema de investigación con los conceptos mínimos necesarios. Además, se da a conocer el planteamiento del problema, hipótesis, objetivos, justificación y delimitación de la tesis. En el capítulo 2 se presentan los conceptos del problema de investigación. En el capítulo 3 se presentan los trabajos del estado del arte para la GP. En el capítulo 4 se presenta el método propuesto para la solución al problema planteado. En el capítulo 5 se presentan los experimentos realizados y resultados obtenidos para validar el método propuesto. En el último capítulo 6 se describen las conclusiones obtenidas del trabajo de investigación.



Capítulo 2.

Marco Teórico

En el siguiente capítulo se definen los conceptos usados para el desarrollo de la presente tesis. Se describe el reconocimiento de patrones, clasificación y las métricas de evaluación. Además de la descripción del AG con sus operadores y la regla de clasificación NN.

2.1 Reconocimiento de objetos.

El reconocimiento de objetos es de gran importancia para la mayoría de los seres vivos (Logothetis, 1996), porque les ayuda a sobrevivir en su medio ambiente. Les ayuda en la identificación de alimento, pareja, migración e identificación de depredadores (Marques De Sa, 2001). Los humanos tienen tan desarrollado el reconocimiento de objetos que lo hacen

inconscientemente (Dougherty, 2013). La mayoría de las actividades que realizan los humanos se basa en la clasificación o reconocimiento de una gran cantidad de objetos (Logothetis, 1996). El reconocer un objeto es un proceso más complejo que simplemente ver un objeto, porque esto sirve como base para el razonamiento y la inferencia que aporta más información del objeto como; la longitud, peso, o función que desempeña (Hummel, 2013). Los humanos reconocen rápidamente y sin esfuerzo un objeto sin importar la orientación, iluminación, color, textura y olor (Hummel, 2013), (Logothetis, 1996). Sin embargo, los humanos no tienen la capacidad de explicar cómo lo realizan (Dougherty, 2013). Lo que ha generado las siguientes preguntas: ¿Cómo logran esto los humanos en su cerebro?, ¿Qué tipo de información usan para construir descripciones de un conjunto de características de los objetos que capturen las propiedades invariantes de los objetos? ¿Cómo activa, representa y almacenan las descripciones del objeto visto? (Logothetis, 1996).

2.1.1 Reconocimiento de patrones

El reconocimiento de objetos se intenta realizar de manera automática con métodos que emulan la capacidad humana para describir y clasificar objetos. La disciplina encargada de tratar con métodos para la descripción y clasificación de objetos se llama reconocimiento de patrones (RP) (Marques De Sa, 2001). El RP usa algoritmos para encontrar regularidades en los objetos (Bishop, 2006). Las regularidades son usadas para asignar una clase a los objetos (Bishop, 1995), (Duda et al., 2000), (Theodoridis & Koutroumbas, 2003).

2.1.2 Clases

Las clases son generalmente palabras que capturan propiedades funcionales presentes en los objetos de una clase. Los humanos hacen este proceso para referirse a los objetos cotidianos, por ejemplo, piñas, monedas, perros, puertas, etc. (Logothetis, 1996). Una clase tiene objetos que son similares, pero no idénticos, y estos se distinguen de otras clases (Dougherty, 2013). Las características de los objetos en una clase son similares, mientras que los objetos de otras clases tienden a valores diferentes en sus características (Duda et al., 2000).

2.1.3 Patrones

Los objetos que se usan para clasificar, como por ejemplo: manzanas, naranjas o huellas dactilares se denominan patrones y no implica una repetición (Dougherty, 2013). Los patrones están formados de un número determinado de características (Gasca & Barandela, 2000). Sea

x_p un patrón con un x_{pi} número de características representadas de la forma; $x_p = (x_{p1}, x_{p2}, \dots, x_{pi}, x_{pw})$, en donde cada x_p pertenece a una clase w de todas las posibles de clases de $\Omega = (w_1, w_2, \dots, w_m)$, y que está dada por x_{pw} en un espacio m -dimensional con x_{pi} que representa un valor en la característica i -th de un patrón x_{p-th} (Triguero et al., 2012). Como ejemplo, si un patrón x_p tiene solo dos características x , y se pueden representar en un plano de dos dimensiones, donde $x_p = [x_{p1}, x_{p2}]$. El valor de $x=x_{p1}$ y el valor de $y=x_{p2}$. Estas características pueden tener diferentes tipos de valores; valores simbólicos, cualitativos, como "negro", "blanco", "etc.", valores numéricos y cuantitativos, que pueden ser continuos o discretos (Kasabov, 2013). Como ejemplo en la tabla 2.1 se muestran siete patrones con su respectiva clase.

Tabla 2.1 Patrones con dos características y su clase.

x_p	x_{p1}	x_{p2}	x_{pw}
x_1	0.455	0.6636	1
x_2	0.355	0.3874	1
x_3	0.788	0.6688	0
x_4	0.9032	0.7975	0
x_5	0.4915	0.1318	0
x_6	0.2112	0.2605	1
x_7	0.7426	0.5133	0

Los patrones de la tabla 2.1 se representan gráficamente en la figura 2.1, donde se muestra el espacio de características ocupado por las clases de los patrones.

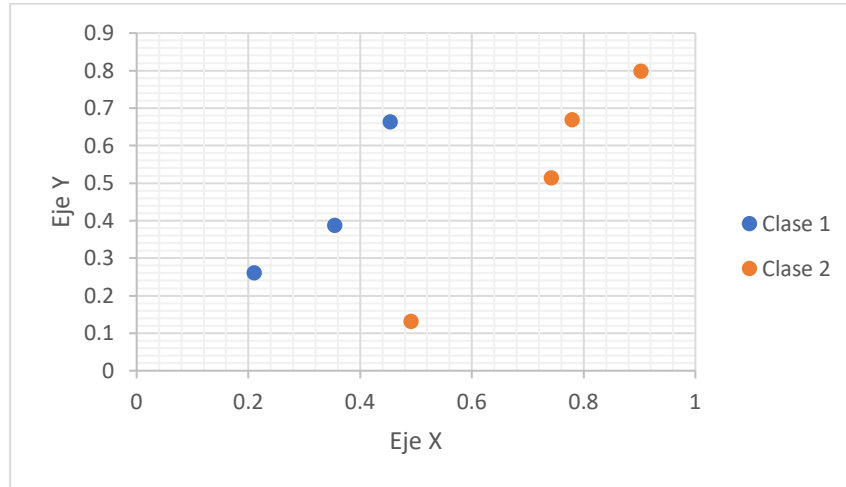


Figura 2.1 Patrones graficados en un plano de dos dimensiones.

2.1.4 Clasificación

La clasificación trata de asignar la clase correcta a los nuevos patrones de todas las posibles clases de $\Omega = (w_1, w_2, \dots, w_n)$ (Marques De Sa, 2001). Esto implica dividir los nuevos patrones a clasificar en diferentes clases de Ω , que ya existen en la muestra de datos (Dougherty, 2013), (Dougherty, 2013). Para realizar la clasificación se usan los valores de las características de cada patrón; en donde las características se representan de la forma $x_{pm} = (x_{p1}, x_{p2}, \dots, x_{pm}, x_{pw})$. Los valores de cada x_{pm} se evalúan para tomar una decisión con respecto a la clase w que se debe asignar al patrón. La asignación de la clase para un nuevo patrón depende de los valores presentes en x_{pm} , estos valores determinan si el patrón está dentro o fuera de la tolerancia de una de las posibles clases de Ω . La clasificación se usa, por ejemplo, para clasificar lesiones como benignas o malignas (Dougherty, 2013). Supongamos que los patrones de la figura 2.1 pertenecen a este tipo de lesiones, y para poder clasificar nuevos patrones se pueden dividir con una línea recta como en la figura 2.2, porque los patrones así lo permiten. Como ejemplo, para clasificar dos nuevos patrones usando como referencia los patrones de la tabla 2.2; $x_8 = (0.444, 0.4636)$ representado con ■ en la figura 2.2, cae dentro de los límites de la clase 1, como resultado se le asigna la clase 1. El siguiente $x_9 = (0.555, 0.444)$ representado con ▲ está dentro de los límites de la clase 2, por lo que se le asigna la clase 2.

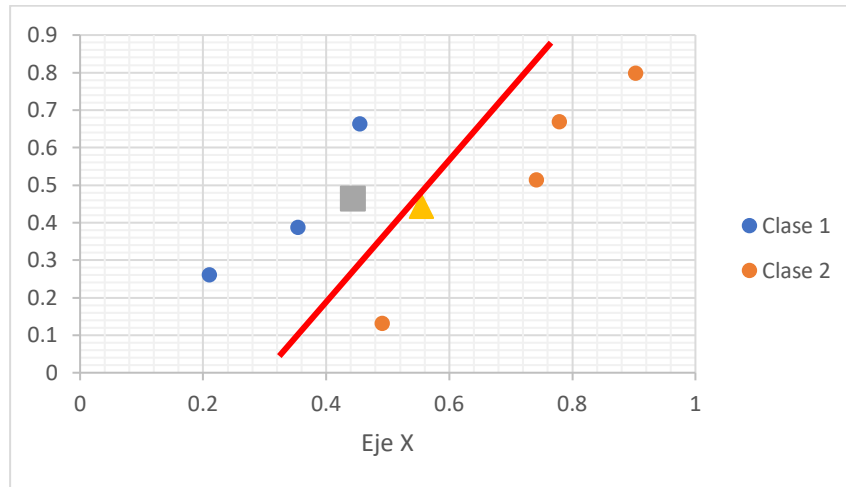


Figura 2.2 División de dos clases con una línea recta.

El clasificar patrones de manera automática implica que el clasificador aprenda de los patrones (Dougherty, 2013), es decir, el clasificador aprende que puede dividir los patrones como en la figura 2.2 con una línea; no obstante, esto no será siempre posible para la mayoría de muestras de datos. Para solucionar esto, los clasificadores aprenden de diferentes formas. El aprendizaje se refiere a una forma de adaptar el algoritmo de clasificación para obtener la mejor respuesta, y esto se logra al extraer información como sea posible de los patrones (Conway & White, 2012). Por lo que no se necesita escribir un programa que siga reglas preprogramadas (The Royal Society, 2017) para resolver un problema de clasificación dado, no obstante, el programa necesitará patrones (Alpaydin, 2010). Los algoritmos de aprendizaje usan los patrones o experiencia para mejorar el rendimiento y hacer predicciones precisas (Mehryar Mohri, Afshin Rostamizadeh, 2012). Esto se logra porque los algoritmos de aprendizaje automático generan un modelo para clasificar nuevos patrones (Hackeling, 2014).

2.1.5 Tipos de aprendizaje automático

El aprendizaje automático se divide en aprendizaje supervisado y no supervisado. En el aprendizaje supervisado las clases de los patrones de la ME se conocen (Dougherty, 2013), y se usan en el entrenamiento para corregir el modelo de aprendizaje automático en cada iteración del entrenamiento (Theodoridis & Koutroumbas, 2003). En el aprendizaje no supervisado los patrones no tienen asignada la clase a la que pertenecen (Theodoridis & Koutroumbas, 2003). En la etapa de entrenamiento, el aprendizaje no supervisado intentará

encontrar un modelo útil dentro del espacio de características (Marques De Sa, 2001). El modelo generado tratará de separar los patrones en clases (Dougherty, 2013). Los modelos obtenidos de estos dos enfoques deben aproximarse a la mayoría de los patrones de la ME para clasificar nuevos patrones (Marques De Sa, 2001).

2.1.6 Validación del modelo de clasificación

Para validar el modelo creado por los algoritmos de clasificación la muestra de datos se debe dividir en una ME y una MP (Cervantes et al., 2007). La ME se usará para entrenar (Hackeling, 2014) y crear el modelo de clasificación. Se espera que la ME esté formada por los patrones y las características que representen las clases de la ME (Francone, 1998). La MP se usará para evaluar el rendimiento del clasificador en nuevos patrones (Hackeling, 2014). La precisión obtenida en la MP mide la generalización alcanzada por los algoritmos de clasificación. La generalización es la capacidad de convertir una gran cantidad de información de la ME (Bishop, 2006) en un conocimiento reducido (Lantz, 2015) que clasificará nuevos patrones en un clase (Gasca & Barandela, 2000), (Kasabov, 2013). Sin embargo, existe un problema de sobreajuste que afecta el modelo de clasificación y afecta su rendimiento.

2.1.7 Sobreajuste

El sobreajuste se produce cuando el algoritmo ha alcanzado un punto en el que para mejorar la clasificación de la ME aprende el ruido de los patrones (Francone, 1998). El sobreajuste se debe evitar, porque se intentará modelar cada variación de los patrones de la ME (Murphy, 2012); produciendo un modelo más complejo que pueda clasificar perfectamente toda la ME (Duda et al., 2000). Sin embargo, en nuevos patrones de la MP no se obtendrá una buena generalización, es decir, se clasifican nuevos patrones incorrectamente (Francone, 1998), (Duda et al., 2000), (Dougherty, 2013). Uno de los algoritmos que tiene la capacidad de clasificar correctamente toda la ME es el clasificador del vecino más cercano (Geva & Sitte, 1991a).

2.2 Clasificación con la regla del vecino más cercano (NN)

Algunos algoritmos de clasificación no crean un modelo ni modifican la ME en la etapa de entrenamiento, como resultado no se sobreajustan, ya que la ME es usada como el modelo de clasificación. La regla de clasificación del vecino más cercano o por sus siglas en inglés K-

Nearest Neighbors (KNN) pertenece a este enfoque. Los clasificadores KNN hacen uso de la regla NN y se llaman métodos de aprendizaje perezoso (Cervantes et al., 2007), (Triguero et al., 2010) que pertenecen a la clasificación supervisada (Garain, 2008). La regla NN fue propuesta por Cover y Hart en 1966 (Raicharoen & Lursinsap, 2005). La regla NN ha demostrado funcionar bien en diferentes problemas de clasificación (Lam et al., 2002). Se ha usado ampliamente en el reconocimiento de patrones y aprendizaje automático (Sánchez et al., 2003). La regla NN es simple y fácil de implementar (Raicharoen & Lursinsap, 2005), y sin embargo, es una de las reglas de clasificación más científicas en la práctica (Kim & Oommen, 2003b). Por lo que se ha convertido en uno de los métodos más estudiados (Sánchez, 2004) desde los años 70 para mejorar su eficiencia y generalización (Decaestecker, 1997) y es considerado uno de los 10 mejores métodos en RP (Triguero et al., 2010).

2.2.1 Clasificación de nuevo patrones

La regla NN clasifica un nuevo patrón en función de sus vecinos o patrones más cercanos de la ME (Kim & Oommen, 2003b). La clase de los n vecinos o el vecino más cercano clasificará al patrón en su misma clase (Fernández & Isasi, 2004). Como resultado, los patrones que son similares son asignados a la misma clase (Fayed et al., 2007). Para calcular los vecinos más cercanos se hace uso de una función de similitud. La función de similitud f_i de un patrón x y los n patrones de la ME se representa como; $ME = (me_1, me_2, \dots, me_n)$ con un espacio ocupando de m -características (Chang, 1974), (Singh et al., 2013) con la función de similitud f_i de la forma:

$$f_1(x) = \min\{d(x, me_1), \dots, d(x, me_n)\}$$

Donde $d(x, me_1)$ es el valor de similitud entre un patrón x y me_1 , $f_1(x)$ es la función de similitud del patrón x con los patrones de la $ME = (me_1, me_2, \dots, me_n)$ para una de las clases de Ω (Chang, 1974) que define el espacio de características de los patrones (Cervantes et al., 2007), (Triguero et al., 2010).

Una de las funciones de similitud más usada por la regla NN es la distancia euclidiana. La distancia euclidiana calcula la raíz de la diferencia cuadrada entre las coordenadas de un par de objetos (Singh et al., 2013).

$$d(a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_m - b_m)^2}$$

Dónde: a, b son los vectores con m -características en el espacio euclidiano.

En donde los a_m, b_m son el número total de características de los patrones. Las características numéricas se definen como $d(a, b) = (a_m - b_m)^2$ (Aha et al., 1991).

2.2.2. Problema del clasificador NN

La regla NN tiene diferentes problemas que afectan la precisión en la clasificación de nuevos patrones. Los problemas de la regla NN se relacionan con el ruido presente en la ME; con patrones mal etiquetados, atípicos y ruidosos (Xie et al., 1993). Además, en ME grandes el número de cálculos para clasificar un patrón aumentan los recursos computacionales considerablemente (Decaestecker, 1997). Para solucionar estos problemas se han propuesto técnicas de reducción de patrones en la ME (Kim & Oommen, 2003a). La reducción se realiza con la selección de prototipos o la Generación de Prototipos (GP). La selección de prototipos identifica y selecciona los patrones de la ME para obtener una submuestra representativa de prototipos (Cervantes et al., 2007). Mientras, la GP selecciona, modifica y si es necesario puede generar nuevos patrones (Garain, 2008), (Nanni & Lumini, 2009). La submuestra de prototipos obtenida es un modelo de clasificación con una menor complejidad y una eficiencia similar o mejor a la ME (Raicharoen & Lursinsap, 2005). Al usar un modelo de clasificación reducido para clasificar nuevos patrones se denomina Clasificador de Prototipos Más Cercanos (CPMC) (Fernández & Isasi, 2004). Los prototipos actúan como representantes de las clases con lo cual se reducen y eliminan los patrones mal etiquetados, atípicos y ruidosos (Fayed et al., 2007). Esto es una mejora para los clasificadores basados en la regla NN que usan como modelo de clasificación toda la ME.

2.3 Métricas de evaluación

Para evaluar el rendimiento de la regla NN y del CPMC se usan patrones nuevos de la MP y una métrica que evalúa el número de patrones clasificados correctamente (Brodersen et al., 2010), (Fernández Hernández et al., 2015). Para calcular la exactitud se usa la siguiente expresión:

$$Exactitud = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Dónde: TP son verdaderos positivos, TN son verdaderos negativos, FP son falsos positivos y FN son falsos negativos, lo cuales se obtienen de la matriz de confusión de tabla 2.2.

Tabla 2.2 Matriz de confusión para dos clases.

	+	-
Clase +	TP	FP
Clase -	FN	TN

En donde TP y TN son el número de patrones que se clasificaron correctamente para cada clase. Mientras, FP y FN son los patrones que se clasificaron incorrectamente (Chicco & Jurman, 2020).

Sin embargo, se tienen ME desbalanceadas que tienen un número mayor de patrones para una clase en comparación con la otra u otras clases (Ertekin et al., 2007). El desbalance en dominios tales como llamadas telefónicas fraudulentas, gestión de telecomunicaciones, clasificación de textos y la detección ciertos objetos de imágenes de satélite (Ertekin et al., 2007) representa un problema. El desbalance en estos dominios es de 100 a 1 por cada clase en algunos casos (Chawla et al., 2002) y el usar la métrica de precisión para evaluar el rendimiento no es fiable. Para evaluar los clasificadores entrenados con ME desbalanceadas se usan métricas como Exactitud Balanceada (EB) y medida f . Para calcular la EB se usa la siguiente expresión:

$$Exactitud Balanceada = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right)$$

Donde el número de clasificaciones correctas es dividido por número de clasificaciones totales (Brodersen et al., 2010).

Para calcular la medida f se usa la siguiente expresión (Huang et al., 2015):

$$F1 = 2 * \frac{(precisión * recuerdo)}{(precisión + recuerdo)}$$

Donde el recuerdo se calcula usando la siguiente expresión (Chicco & Jurman, 2020):

$$recuerdo = \frac{TP}{TP + FN}$$

La precisión se calcula usando la siguiente expresión (Chicco & Jurman, 2020):

$$precisión = \frac{TP}{TP + FP}$$

Sin embargo, se hace uso de la Medida f Ponderada (MfP) que evalúa cada clase por separado y calcula su promedio ponderado por soporte (Scikit learn, 2021).

2.4 Validación cruzada

La complejidad, generalización y reducción de los modelos generados por los métodos de CPMC se evalúa usando la validación cruzada (Odorico, 1997), (Fernández & Isasi, 2004). La validación cruzada es un método para remuestrear datos y evaluar la generalización de modelos de clasificación y evitar el sobreajuste (Berrar, 2018). La validación cruzada divide la muestra de datos n -veces. La muestra de datos se divide aleatoriamente en subconjuntos del mismo tamaño (Kohavi, 1995). Por ejemplo, cuando $n = 10$, es decir, la validación cruzada es igual a 10, la primera división del primer subconjunto sirve como MP y los nueve restantes son usados como la ME (Berrar, 2018). En la figura 2.3 se muestra la división de la ME en $n=10$.

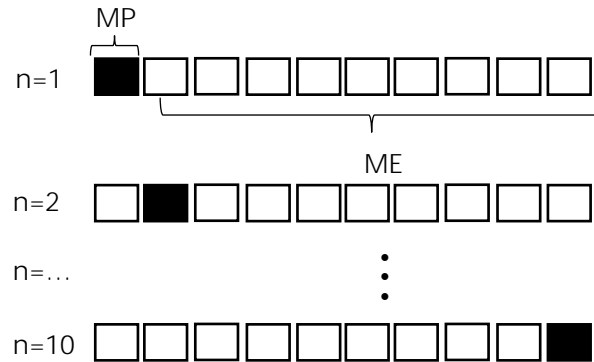


Figura 2.3 Validación Cruzada.

Cada cuadro de la figura 2.3 representa una n cantidad de patrones de la muestra de datos. Como ejemplo, si la validación cruzada es con $n=10$ y la métrica de evaluación es precisión, el resultado de la validación cruzada será el promedio de la precisión de todas las submuestras de ME o MP (Brodersen et al., 2010).

2.5 Optimización

La naturaleza sigue un camino interesante para encontrar una solución óptima a cualquier problema (Samanta, 2014). Una solución óptima tiene una mejor calidad que otras posibles soluciones. La optimización trata de encontrar una solución óptima en el espacio de soluciones (Kramer, 2017). En la computación se tienen algoritmos basados en conceptos biológicos de evolución para problemas de optimización (Kramer, 2017).

2.5.1 Algoritmos genéticos

Los algoritmos genéticos (AG) son modelos computacionales inspirados en la evolución, búsqueda y supervivencia de los individuos más aptos (Whitley, 1994), (Ribeiro Filho et al., 1994), (Adeli & Sarma, 2006). Los AG inician con un conjunto inicial de soluciones aleatorias llamado población que está dentro de los límites o restricciones del problema (Gen et al., 2008). La población está formada por individuos que representan estructuras similares a un cromosoma y se codifican para el problema en específico (Whitley, 1994). Los individuos de la población son mejorados gradualmente por un proceso de evolución (Kendal & Creen, 2007) que tiene los siguientes pasos; selección, reproducción, cruce y mutación (Adeli & Sarma, 2006). Los

individuos más aptos tienen una posibilidad de sobrevivir y generar descendencia, transmitiendo su herencia a la nueva generación (Ribeiro Filho et al., 1994). El proceso es cíclico y se repite para obtener una solución óptima, a cada iteración se le llama generación. En la figura 2.4 se muestra el proceso del AG.

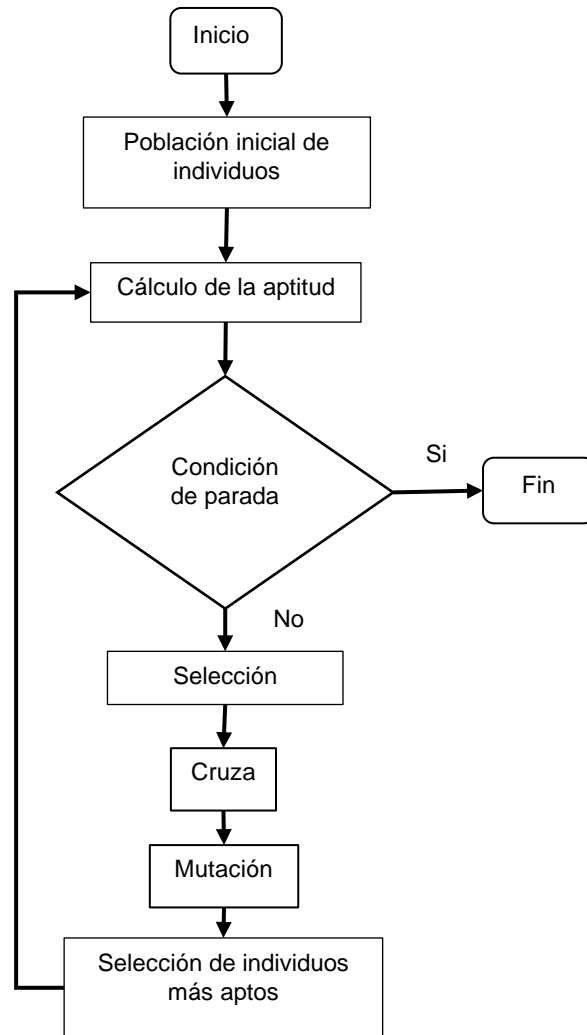


Figura 2.4 Diagrama de flujo del algoritmo genético.

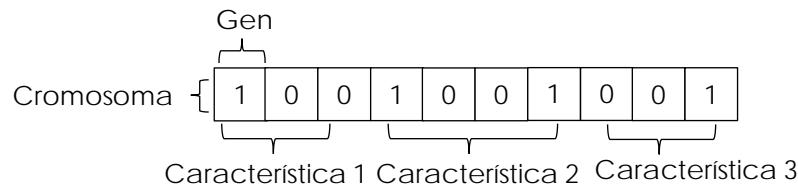
a) Codificación y población inicial

El AG inicia con la codificación de la solución potencial de un problema. Generalmente la solución potencial tiene el nombre de cromosoma. El cromosoma está formado por genes y cada gen o grupo de genes representa una característica determinada en particular del cromosoma (Mitchell, 1998), también llamada fenotipo. La codificación se puede representar con valores binarios y números reales (Ribeiro Filho et al., 1994). La inicialización del AG es

simple: se crea una población de cromosomas, donde cada cromosoma es de un tamaño específico y los genes generalmente se inicializan aleatoriamente (Michalewicz, 1996). Un cromosoma de diez posiciones y una combinación aleatoria inicial como una cadena de genes se muestra en la figura 2.5 (Whitley, 1994), (Kramer, 2017).

Figura 2.5 Codificación de un cromosoma de diez posiciones.

En donde el cromosoma tiene un tamaño de diez y en total representa 3 características de la posible solución.



b) Función de aptitud

Una vez que se crea la población inicial se debe evaluar cada cromosoma o individuo de la toda la población. La evaluación de los cromosomas se hace con la función de aptitud (Kramer, 2017). La aptitud es un indicador de supervivencia y reproducción los organismos en generaciones posteriores (Adeli & Sarma, 2006). En los AG la aptitud de un cromosoma evalúa la calidad alcanzada que depende de que tan bien resuelve el problema (Mitchell, 1998). En cada generación se producen cromosomas con una aptitud relativamente buena y mala. En el proceso de evaluación se busca que solo se reproduzcan cromosomas con una buena función de aptitud y eliminar cromosomas con una baja aptitud (Gen et al., 2008). Una aptitud alta del cromosoma le permite influir en el desarrollo de posteriores generaciones. Cuando varios descendientes de un cromosoma dado sobreviven para reproducirse, muchos cromosomas de la próxima generación portaran genes de ese individuo (Holland, 1992).

c) Condición de paro

La condición de paro en un AG se puede realizar de acuerdo con diferentes criterios. El AG finaliza cuando deja de encontrar mejores soluciones (Ahn, 2006). Generalmente se ejecutan durante un número predefinido de generaciones. Mientras, más cerca este de la solución óptima, el progreso de mejora en la aptitud disminuye significativamente (Kramer, 2017). Algunas de las condiciones de paro propuestas por (Samanta, 2014) son:

- Número máximo generaciones.

- Límite de tiempo máximo.
- Aptitud máxima alcanzada.

d) Operadores de selección

Los AG juegan al casamentero al seleccionar dos cromosomas de la población para generar nuevos descendientes (Ahn, 2006). La selección dirige la búsqueda genética explorando y reduciendo el espacio de búsqueda hacia las regiones prometedoras de solución (Gen et al., 2008). Lo que permite la convergencia del AG en soluciones óptimas. En cada generación se elegirán los cromosomas con mejor aptitud para generar la próxima generación (Ribeiro Filho et al., 1994). En problemas de minimización se eligen valores bajos de aptitud y en caso de problemas maximización viceversa (Kramer, 2017). Cuando la aptitud de un cromosoma es mejor que la del resto de la población tiene una probabilidad más alta de ser seleccionado (Mitchell, 1998). Para realizar la selección de cromosomas se puede hacer uso del método de selección por torneo o ruleta.

1) Selección por torneo

Selección por torneo: imita la competencia de apareamiento en la naturaleza, elige al azar un subconjunto de cromosomas (dos o tres) de la población, y el cromosoma con la mejor aptitud se convierte en padre (Gen et al., 2008). La selección de torneo permite que cada cromosoma sobreviva, incluidos los cromosomas con valores de aptitud bajos (Kramer, 2017). Sin embargo, en torneos grandes tienen una probabilidad baja de ser seleccionados (Gen et al., 2008). El torneo se repite para cada padre necesario, en general esta selección genera buenas parejas y no requiere de ordenar la población. Además, en poblaciones grandes funciona mejor (Ahn, 2006).

2) Selección por ruleta

Selección por ruleta: determina la probabilidad de selección para cromosoma en función de su valor de aptitud. Donde cada cromosoma tiene una probabilidad de ser seleccionado (Kramer, 2017). Lo que genera un modelo de rueda de ruleta mostrando estas probabilidades. La selección se basa en girar la rueda el número de veces igual al tamaño de la población, en cada giro se selecciona solo un cromosoma. La selección de ruleta no permite seleccionar cromosomas duplicados de la población (Gen et al., 2008).

e) Operadores genéticos

En la naturaleza, la mayoría de las especies tiene dos padres que combinan su material genético y lo heredan a su descendencia. En los AG los cromosomas representan partes exitosas de la solución que, cuando se combinan, incluso superan a sus padres (Kramer, 2017). El operador de cruce del AG toma dos cromosomas y los combina para intercambiar información genética para producir nuevas cromosomas (Gen et al., 2008). La forma más sencilla de realizar una cruce es elegir un punto al azar de cruce, las porciones de las cadenas principales p_1 y p_2 se intercambian para producir nuevas cadenas descendientes o_1 y o_2 (Ribeiro Filho et al., 1994). Por ejemplo, si los padres p_1 y p_2 tienen un tamaño de cinco genes como $p_1 (a_1, b_1, c_1, d_1, e_1)$ y $p_2 (a_2, b_2, c_2, d_2, e_2)$ al cruzar los cromosomas después del segundo gen producirá la descendencia $o_1 (a_1, b_1, c_2, d_2, e_2)$ y $o_2 (a_2, b_2, c_1, d_1, e_1)$ (Michalewicz, 1996).

1) Cruza uniforme

En la cruce uniforme se toma cada gen de los cromosomas padres y asigna aleatoriamente el gen de un padre a un hijo y el gen del otro padre al otro hijo. Para hacer esto se genera una máscara aleatoria que actúa como un colador (Adeli & Sarma, 2006). La máscara es un vector de unos y ceros aleatorios con la misma longitud de los padres. Los valores de las máscaras segmentarán los genes que portaran los hijos (Ahn, 2006). Los n -segmentos de los padres se deben alternar entre los hijos (Michalewicz, 1996). En la siguiente figura se muestra la cruce de dos cromosomas y sus cromosomas resultantes.

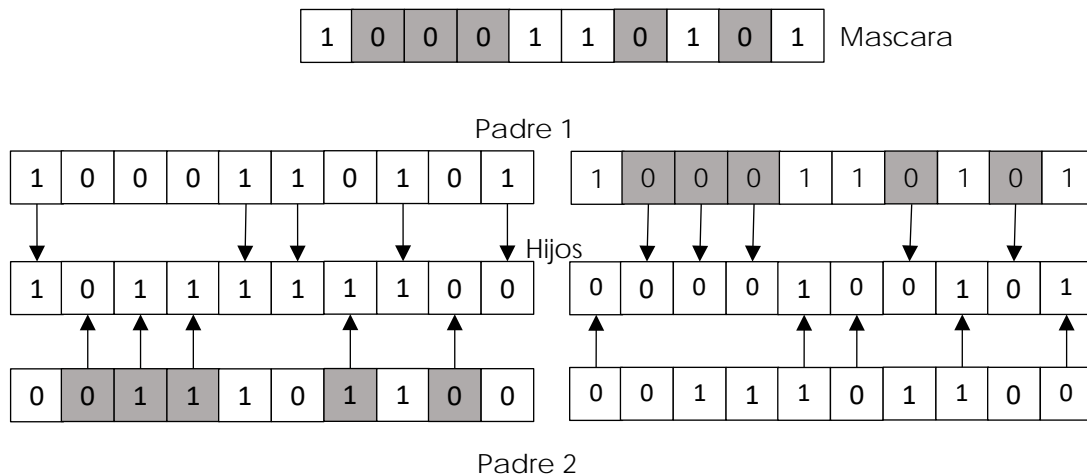


Figura 2.6 Cruza uniforme.

f) Mutación

La mutación es el siguiente operador en un AG. La mutación tiene una pequeña probabilidad de ocurrir en el mundo real y en los AG tienen un papel importante (Adeli & Sarma, 2006). La mutación previene la pérdida de valores en ciertos genes de los cromosomas causados por los operadores de selección y cruce. Si esto ocurre, el AG convergerá prematuramente (Whitley, 1994). La mutación trata de mantener una diversidad genética en los cromosomas (Mitchell, 1998). La mutación introduce rasgos que no se tienen en la población original para que el AG explore todos los posibles espacios de solución (Ahn, 2006). La mutación se aplica a cada cromosoma de la nueva población (Ribeiro Filho et al., 1994) alterando arbitrariamente uno o más genes del cromosoma elegido (Michalewicz, 1996). El cromosoma se elige con base en una probabilidad de mutación (Gen et al., 2008). Una forma de realizar la mutación en un cromosoma basa en el cambio aleatorio de un gen (Kramer, 2017).

1) Mutación por inversión

La mutación por inversión parte el cromosoma en tres subcadenas y solo muta una de las tres subcadenas (Eiben & Smith, 2015). La partición se hace al seleccionar aleatoriamente dos posiciones a lo largo del cromosoma (Gen et al., 2008), (Michalewicz, 1996), y los genes dentro de las dos posiciones seleccionados se invierte (Ahn, 2006). La inversión se realiza al cambiar las posiciones de los genes del cromosoma (Whitley, 1994). En la figura 2.7 se muestra gráficamente la mutación por inversión de un cromosoma.

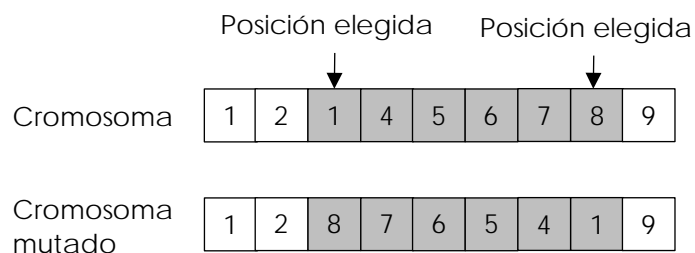


Figura 2.7 Representación de la mutación por inversión.

Sin embargo, la mutación por inversión para cromosomas con una gran cantidad de genes, en donde un valor puede no volver a presentarse y perderse como en la figura 2.7.

En esta figura el valor que ya no se presenta es el número 3 y la mutación por inversión no ayudará a prevenir la pérdida de valores en ciertos genes.

2) Mutación por inversión base- n

La mutación por inversión de base- n trata de prevenir la pérdida de valores de la mutación por inversión tradicional. La mutación por inversión base- n genera la mutación de los genes de acuerdo con la base de la codificación usada en el AG. El gen puede tomar un valor de b de la codificación usada. Como ejemplo, en la figura 2.8 se muestra un cromosoma con una codificación $b = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ por lo que el gen mutado puede tomar algún valor de b . El valor asignado al gen mutado de la figura 2.8 es 3 que ocupa la tercera

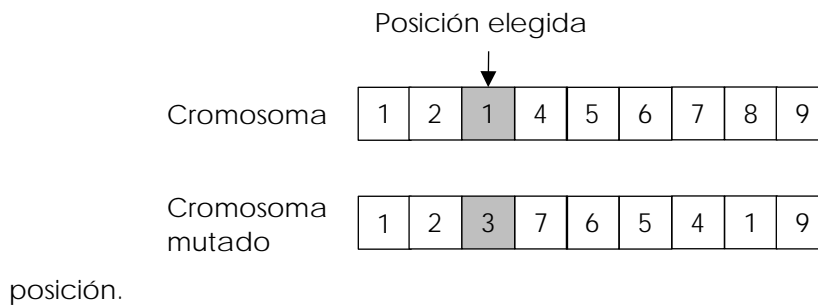


Figura 2.8 Representación de la mutación por inversión base- n .

2.6 Resumen

En este capítulo se presentaron los conceptos necesarios para entender el problema de investigación. Se inicia con el reconocimiento de objetos que todos los seres vivos realizan para sobrevivir. El reconocimiento de objetos se trata de imitar con algoritmos y modelos computacionales. Estos son desarrollados por el RP para problemas de clasificación. Además, se explica la división de la clasificación supervisada y no supervisada que depende de las muestras de datos a usar para entrenar los algoritmos de clasificación.

Se describen los elementos necesarios para la entrenamiento, clasificación, evaluación y sobreajuste de los algoritmos de clasificación. El clasificador presentado fue la regla NN que es uno de los más simples de usar e implementar. Este algoritmo no realiza ninguna modificación de la ME en el entrenamiento. Por lo que se considera un algoritmo de clasificación perezoso. El proceso de clasificación del NN se describe como el cálculo de

Marco Teórico

similitud entre 1 patrón o n -patrones de la ME para clasificar un nuevo patrón. La similitud entre los patrones se calcula con la distancia euclidiana. Estos conceptos son necesarios por que el problema de investigación de la tesis pertenece al RP, ya que busca generar prototipos para generar un modelo de clasificación que se usará clasificar nuevos patrones.

Para generar el modelo de clasificación y dar la solución al problema planteado en la presente investigación se presentó la definición de los AG. Los AG se basan en la selección y supervivencia del más apto. Esto se aplica para buscar la mejor solución posible dentro de un espacio de soluciones. Se da una breve definición para representar la codificación y función de aptitud en el AG. Además, de los operadores que forman el AG como son los operadores de selección, cruza, mutación y paro que simulan una evolución de las posibles soluciones.



Capítulo 3.

Estado del Arte

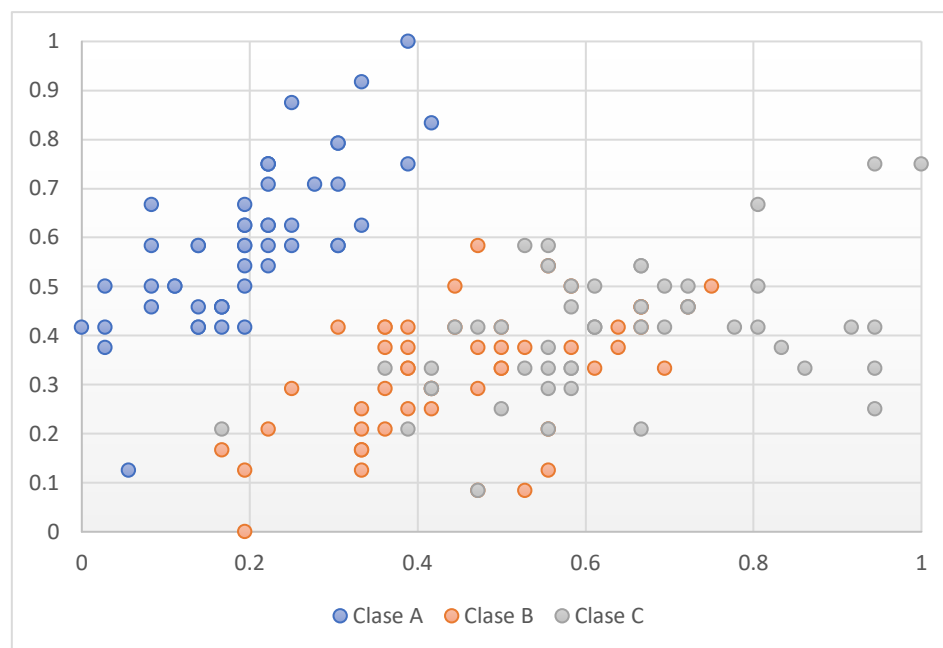
Los métodos en el estado del arte para la GP se enfocan en mejorar los resultados obtenidos para regla 1-NN. La mejora de los resultados se obtiene al generar prototipos de la ME original. A continuación, se presentan los pasos generales con una breve descripción de cómo los métodos de GP inician, generan, optimizan y validan los prototipos. Después se continuará con la descripción de que pasos usa cada método de GP.

3.1 Descripción general de los métodos de GP

Los métodos de GP comparten un proceso general para iniciar, generar, optimizar y validar los prototipos. Sin embargo, este proceso no es similar para todos los métodos de GP, por lo que se presenta una descripción general para la mayoría de los métodos de GP.

3.1.1 Inicialización

En algunos métodos se solo inician con 1 prototipo por clase y en otros métodos seleccionan un subconjunto de prototipos aleatorio o representativo. A continuación, se describen los diferentes tipos de inicialización. La Inicialización en los métodos de GP es un paso importante para algunos de los métodos de GP y para otros métodos de GP no tanto. Para mostrar los diferentes tipos de inicialización se usa la muestra de datos de "iris". La muestra de datos de iris contiene 150 patrones con 3 clases diferentes. En la figura 3.1 se muestran gráficamente la distribución de los patrones de la muestra de datos de iris usando solo la característica de



longitud del sépalo y ancho del sépalo.

Figura 3.1 Representación gráfica de la muestra de datos de iris con dos características.

a) Selección aleatoria de un prototipo por clase

La selección de prototipos aleatorios generalmente inicia con solo un prototipo por clase, y la posición que ocupan estos prototipos no afecta el rendimiento del método de GP. En la figura

3.2 se muestra la selección de varios prototipos aleatorios por clase. Los prototipos seleccionados de la figura 3.2 se marcan con un rectángulo verde. No obstante, cualquier patrón de la muestra de datos de iris se puede seleccionar. En este paso ya no se agregan más prototipos por clase. Como resultado, la muestra de datos de iris solo inicia con una submuestra de 3 prototipos.

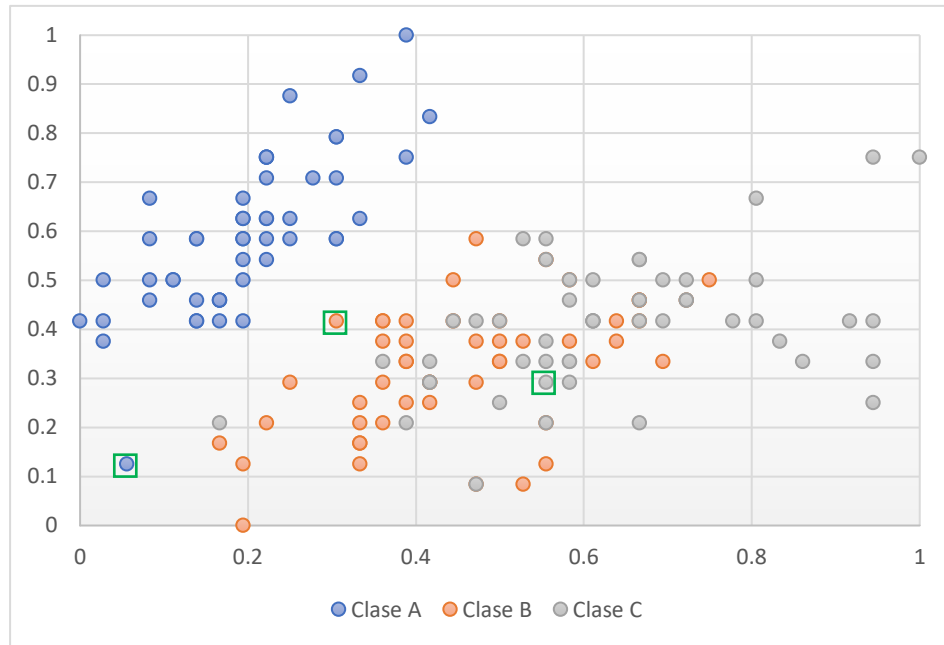


Figura 3.2 Selección aleatoria de prototipos iniciales por clase.

b) Selección de un centroide por clase como prototipo inicial

La selección de centroides como prototipos iniciales busca cubrir un área más grande dentro del espacio de características con un prototipo por clase. Tomando como ejemplo la muestra de datos de iris y el algoritmo de agrupamiento *k-means* usado en (Decaestecker, 1997) se generan los centroides iniciales para cada clase. En la figura 3.3 se señalan con un cuadro color rojo los centroides generados por el algoritmo *k-means*. Sin embargo, los centroides iniciales se pueden generar de diferentes formas, como;

- La selección de los patrones más alejados de la muestra.
- La selección de los patrones más cercanos a los límites de clase.
- La selección aleatoria de patrones.

Estado del Arte

Una vez seleccionados los patrones se calcula el centroide o la media de los vecinos cercanos para obtener los prototipos iniciales. Como ejemplo, en la figura 3.4 se grafican con un triángulo los prototipos iniciales. Los triángulos marcan el lugar que se espera que ocupen los nuevos prototipos.

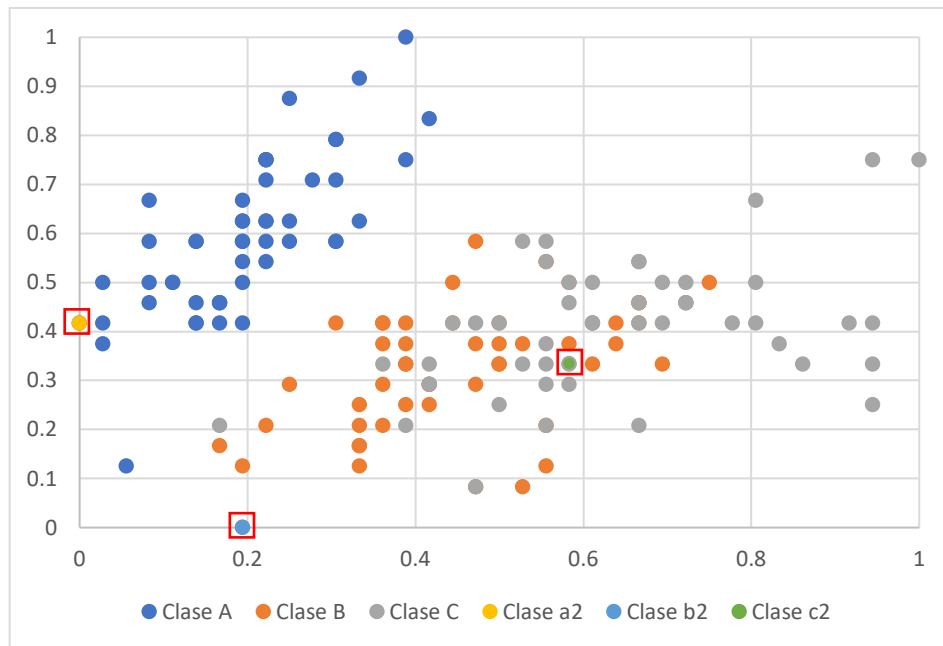


Figura 3.3 Prototipos iniciales usando el algoritmo de agrupamiento *k-means*.

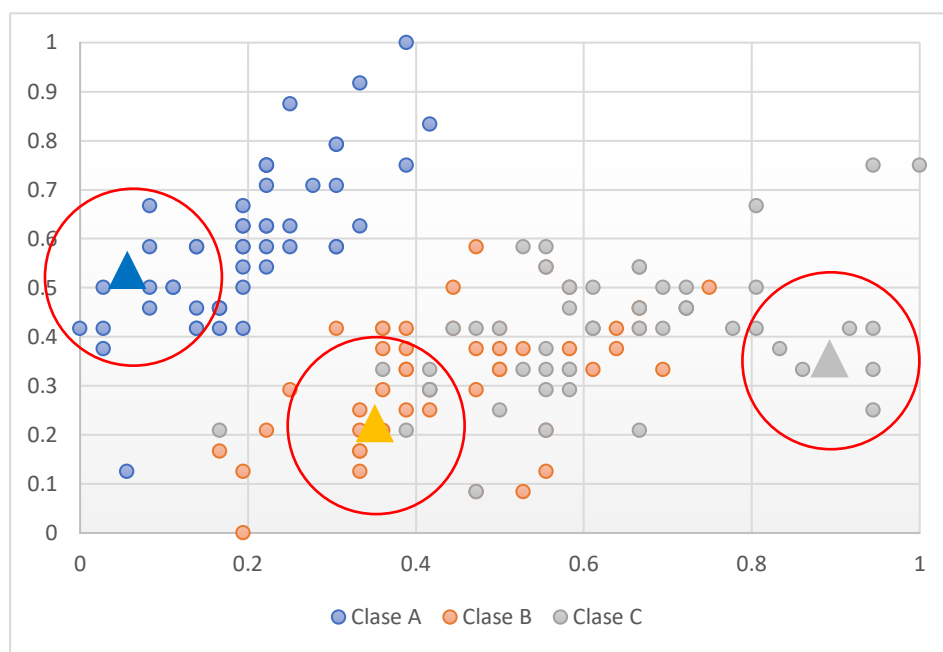


Figura 3.4 Prototipos iniciales usando información los patrones seleccionados.

c) Selección de una submuestra de prototipos

La selección de una submuestra de prototipos requiere de más de un patrón por clase. La submuestra se elige aleatoriamente si el método no requiere de un número específico de patrones por clase. No obstante, algunos métodos necesitan de un número de prototipos por clase, y el número de patrones por clase no puede ser aleatorio. La submuestra tiene un tamaño aproximado de un 5% a 20% del tamaño de la ME (Cervantes et al., 2007). En la figura 3.5 se muestran con círculos color rojo los prototipos seleccionados para una submuestra inicial. La submuestra de la figura 3.5 tiene el mismo número de prototipos por clase.

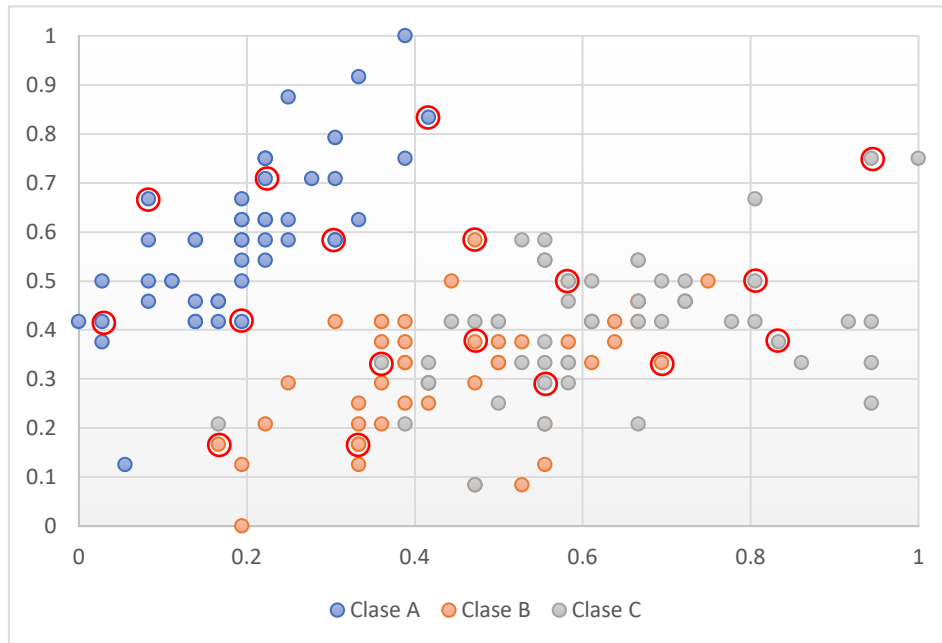


Figura 3.5 Selección de una submuestra de prototipos.

3.1.2 Generación y optimización de los prototipos

Los métodos de GP se pueden dividir en métodos de GP que generan los prototipos con la información de sus vecinos más cercanos en la ME, y métodos de GP que optimizan la posición de los prototipos. Algunos métodos combinan la generación y optimización para obtener los prototipos. A continuación, se hace una descripción generalizada de la generación, optimización y de la combinación de estos dos procesos.

a) Generación de prototipos.

Los métodos de GP que usan la información de sus vecinos más cercanos para generar nuevos prototipos, generan centroides, combinan o calculan la media de los vecinos más cercanos para generar un nuevo prototipo. El nuevo prototipo se trata de colocar cerca de los patrones mal clasificados de la ME. Usando como ejemplo, la muestra de datos de iris de la figura 3.3 se muestra el proceso de generación de nuevo prototipos. En la figura 3.6 se muestra los nuevos prototipos con la forma de una estrella. Estos nuevos prototipos se generan con la información de sus vecinos más cercanos. La generación de los prototipos puede variar en cada método y usar técnicas diferentes para posicionarlo.

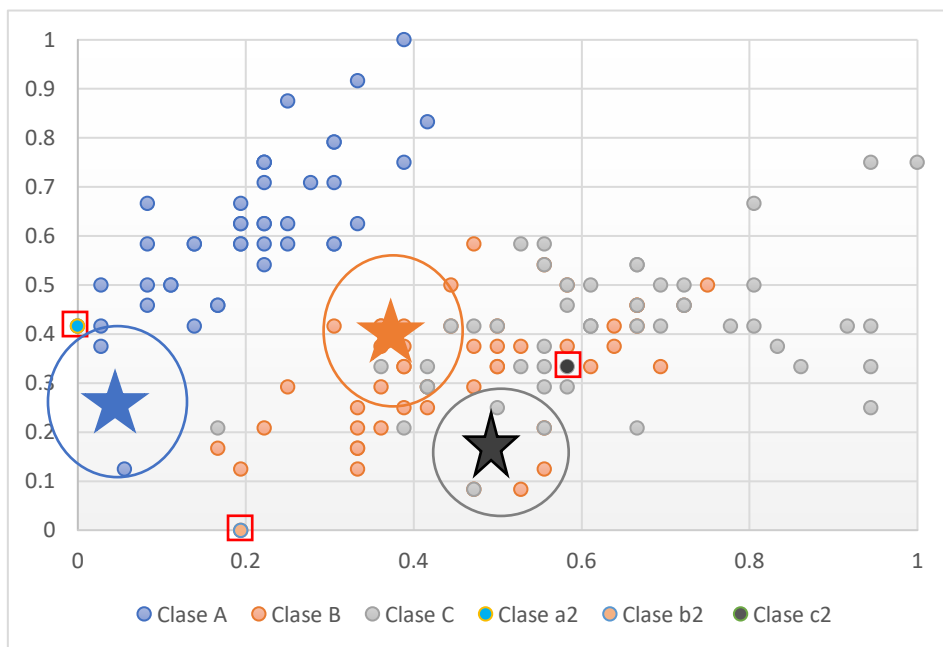


Figura 3.6 Generación de nuevos prototipos con la información de los vecinos más cercanos.

Sin embargo, en la mayoría de los métodos de GP los patrones mal clasificados de la ME se usan para generar nuevos prototipos en la nueva iteración.

b) Optimización de prototipos

La optimización de los prototipos se usa generalmente en las submuestras de los prototipos. La posición de los prototipos se optimiza para que cubran un espacio de características más amplio dentro la ME. La optimización se realiza al ajustar los valores de las características de

los prototipos. El ajuste de los prototipos se puede realizar al sumar o restar los valores en sus características. En otros casos los patrones de la ME se usan para modificar la posición de los prototipos. Algunos métodos cambian la clase del prototipo o generan nuevos prototipos para una clase. La optimización de la posición trata de colocar los prototipos cerca de patrones de su misma clase y alejarlos patrones de clases diferentes (Geva & Sitte, 1991b). En la figura 3.7 se muestra un ejemplo hipotético de optimización en la posición de la submuestra. Este tipo de métodos inician con grandes cambios en los prototipos, y en cada iteración la modificación es menor (Triguero et al., 2012).

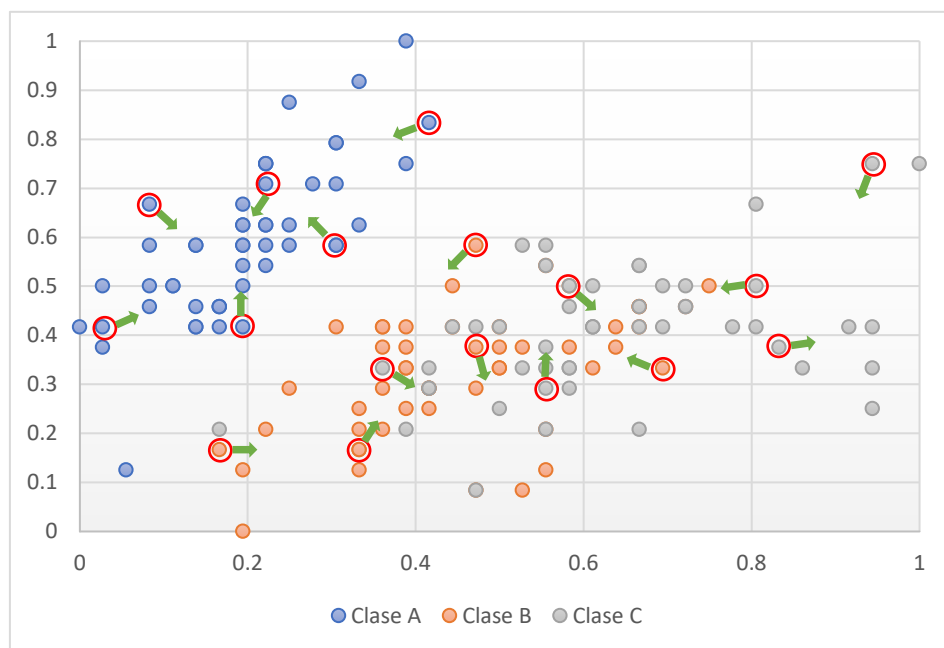


Figura 3.7 Optimización de la submuestra de prototipos en el espacio de características.

c) Generación y optimización de los prototipos

La generación y optimización de prototipos generalmente inicia con una selección aleatoria de un prototipo por clase de la ME. La generación de los prototipos se inicia con la modificación o combinación de los prototipos iniciales, esto se hace para generar nuevos prototipos. La optimización trata de encontrar los mejores prototipos generados. Los mejores prototipos son los que clasifican correctamente un mayor número de patrones de la ME. Estos prototipos se agregan a una submuestra de prototipos que aumenta iterativamente, y pueden reemplazar a los prototipos actuales de la submuestra o solo agregarse para aumentar el número de prototipos. En la figura 3.8 se muestra un ejemplo hipotético de los prototipos

generados se representan con una estrella, y los prototipos que no son seleccionados se eliminan.

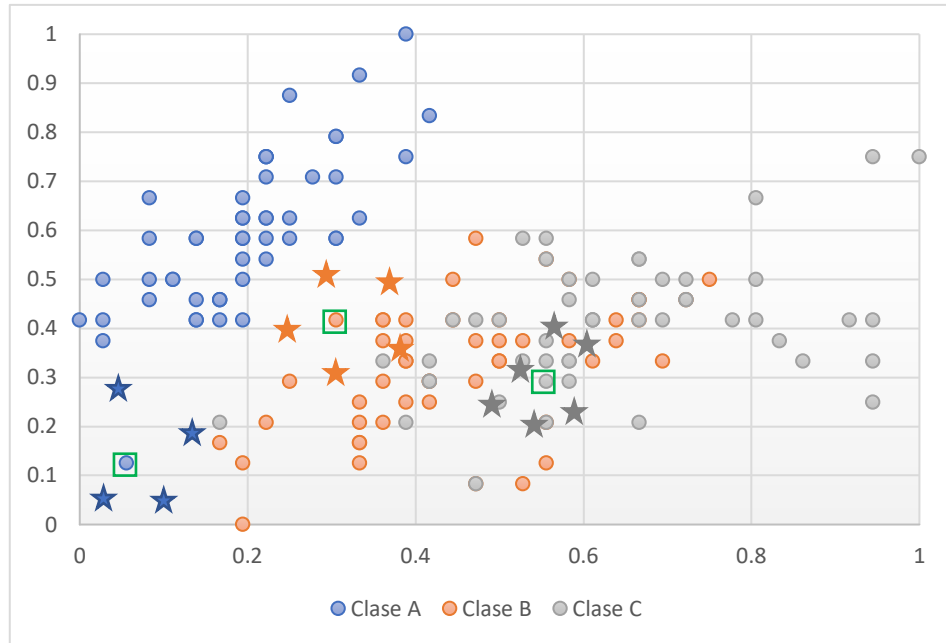


Figura 3.8 Generación y optimización de los prototipos.

3.1.3 Validación de los prototipos

Para la validación de los prototipos se usa generalmente una medida de distancia, la regla de clasificación y una muestra de datos. Para validar los prototipos se siguen los siguientes pasos;

- Clasificar la ME con los prototipos.
- Evaluar la clasificación obtenida.
- Asignar la precisión alcanzada que generalmente va 0 a 1.
- Eliminar los prototipos que no mejoran la clasificación.

Los métodos de GP generalmente hacen uso de la distancia euclidiana, la regla 1-NN y de una ME. La eliminación de los prototipos reduce el número de prototipos que no aportan una mejora en la clasificación. Generalmente, se eliminan los prototipos que tienen por vecinos más cercanos a patrones de otras clases. En algunos métodos se hace solo una modificación de la clase. Como resultado, se elimina un prototipo de una clase y se obtiene un nuevo prototipo para otra clase. La validación trata de mantener los prototipos que clasifican un

mayor número de patrones de su misma clase. Sin embargo, algunos métodos no realizan una eliminación de los prototipos durante todo el proceso de generación, optimización y validación.

3.2 Ajuste de la posición de los prototipos con modelos vectoriales

Uno de los métodos más estudiados y que más variantes presenta para la generación de prototipos es *Learning Vector Quantization* (LVQ) (Bezdek & Kuncheva, 2001). El método LVQ genera un número determinado de prototipos y no genera, elimina o cambia la clase de los prototipos iniciales. El determinar el número correcto de prototipos iniciales es el principal problema de los métodos LVQ (Fernández & Isasi, 2004).

3.2.1 Learning Vector Quantization (LVQ)

El método LVQ ajusta la posición de los prototipos, mientras reduce el error de clasificación (Lozano et al., 2006). El método LVQ es una red neuronal de tres capas: la primera capa de entrada tiene tantas neuronas como características tienen los patrones; la segunda capa intermedia contiene un número definido de neuronas LVQ (prototipos); la capa de salida tiene tantas neuronas como clases la muestra de datos (Odorico, 1997). En la segunda capa es donde se ajusta la posición de los prototipos. El ajuste se realiza al entrenar la red LVQ con la ME para modificar las conexiones entre las neuronas de la segunda capa y las neuronas de salida (Triguero et al., 2010). En la figura 3.9 se muestra la representación de la red LVQ.

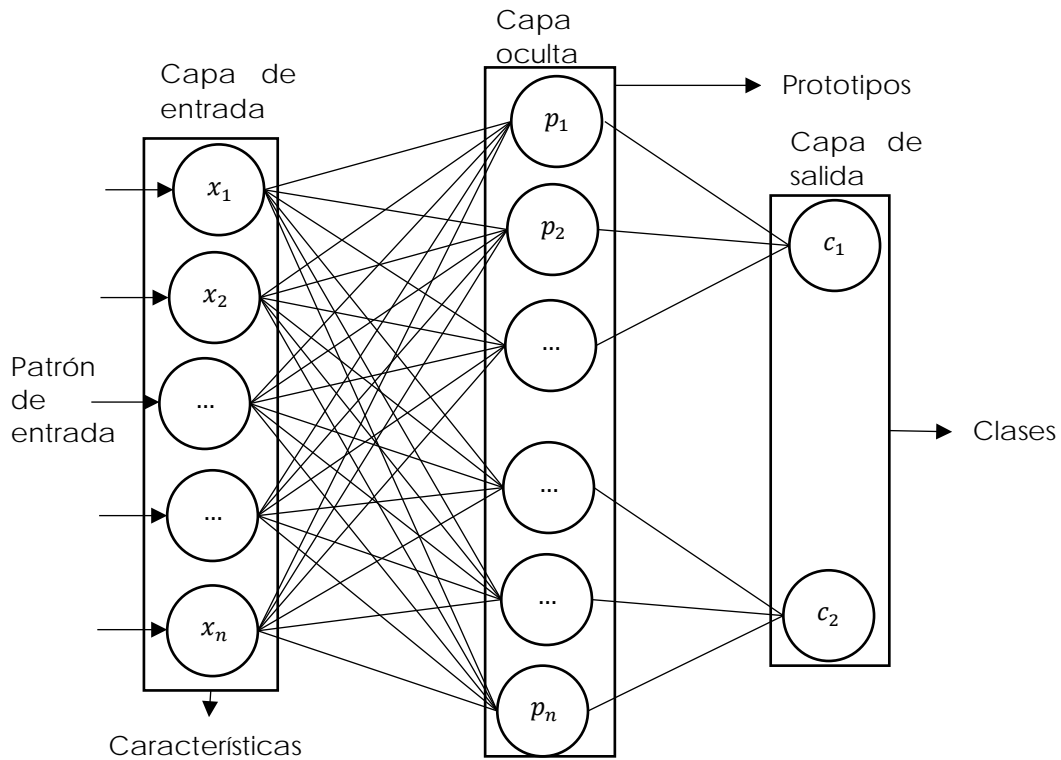


Figura 3.9 Representación de la red LVQ con 2 clases.

El método LVQ tiene tres variantes llamadas LVQ1, LVQ2 y LVQ3. La estructura de la red neuronal no cambia en estas variantes, solo cambia la regla para modificar a un prototipo (Kohonen, 1990). La descripción de las diferentes variantes de LVQ se realiza porque varios métodos de GP hacen uso de estas técnicas y realizan una modificación a los métodos originales para mejorar la clasificación (Odorico, 1997). A continuación, se describe la forma en que cada método de LVQ hace el ajuste de los prototipos.

a) *LVQ1*

El método de LVQ1 ajusta la posición de los prototipos con una recompensa o castigo (Geva & Sitte, 1991a). La recompensa se aplica cuando el prototipo más cercano clasifica correctamente al patrón actual. En caso contrario se realiza un castigo al prototipo más cercano. El método LVQ1 ajusta la posición del prototipo de acuerdo con la siguiente regla

general. Que acerca el prototipo al patrón si la clasificación es correcta y lo aleja en caso contrario.

$$\begin{cases} y_i(t+1) = y_i + \alpha(t)[x_p(t) - y_i(t)] & \text{si la clase}(x_p) == \text{clase}(y_i) \\ y_i(t+1) = y_i - \alpha(t)[x_p(t) - y_i(t)] & \text{si la clase}(x_p) \neq \text{clase}(y_i) \\ y_k(t+1) = y_k(t) & \text{si } k \neq \text{clase}(x_p) \end{cases}$$

t es la época de la red LVQ

y_i corresponde al prototipo a modificar

$x_p(t)$ es el patrón de entrada

$\alpha(t)$ es la tasa de aprendizaje

En donde $0 < \alpha(t) < 1$ que puede ser constante o disminuir uniformemente en el tiempo de entrenamiento (Triguero & Herrera, 2012).

b) LVQ2

El método LVQ2 ajusta la posición del prototipo si la clasificación es correcta o incorrecta (Li et al., 2005). El ajuste se realiza si el prototipo más cercano al patrón tiene una clase diferente, pero si el siguiente prototipo más cercano tiene la misma clase del patrón. Como se muestra en la siguiente regla general (Geva & Sitte, 1991a).

$$\begin{cases} y_i(t+1) = y_i + \alpha(t)[x_p(t) - y_i(t)] & \text{si la clase}(x_p) \neq \text{clase}(y_i) \\ y_j(t+1) = y_j - \alpha(t)[x_p(t) - y_j(t)] & \text{si la clase}(x_p) == \text{clase}(y_j) \end{cases}$$

El regla la describe el autor Kohonen (Kohonen, 1990) como; si el prototipo y_i es la más cercano, pero el patrón x_p pertenece a la clase del prototipo y_j , en donde el prototipo y_j es la siguiente clase más cercana, entonces, para actualizar el patrón x_p debe caer en una ventana definida. En caso contrario aplicar la siguiente regla.

$$y_k(t+1) = y_k(t)$$

c) LVQ3

En LVQ3 la ajusta la posición del prototipo haciendo una combinación de LVQ1 y LVQ2, y lo realiza siguiendo la siguiente regla general.

$$\begin{cases} y_i(t+1) = y_i - \alpha(t)[x_p(t) - y_i(t)] \\ y_j(t+1) = y_j + \alpha(t)[x_p(t) - y_j(t)] \end{cases}$$

Donde el prototipo y_i y el prototipo y_j son los prototipos más cercanos al patrón x_p . El patrón x_p y el prototipo y_j pertenecen a la misma clase, mientras el patrón x_p y el prototipo y_i pertenecen a diferentes clases. En caso contrario, donde el patrón x_p , el prototipo y_i y el prototipo y_j pertenecen a la misma clase se sigue la siguiente regla de ajuste.

$$y_k(t+1) = y_k - \epsilon\alpha(t)[x_p(t) - y_k(t)]$$

Para $k = \{i, j\}$, si x, y_i y y_j pertenecen a la misma clase. El valor puede tomar ϵ puede ser de 0.1 a 0.5 (Kohonen, 1990).

Los métodos basados en LVQ son sensibles al número inicial por clase y la tasa de aprendizaje (Fayed et al., 2007). Sin embargo, determinar la tasa de aprendizaje, la cantidad de prototipos iniciales y la ubicación de estos, es una tarea difícil para cada ME (Li et al., 2005).

3.2.2 *Decision Surface Mapping (DSM)*

- Selección de una submuestra representativa
- Optimización de los prototipos

El método *Decision Surface Mapping* (DSM) propuesto por (Geva & Sitte, 1991b) ajusta la posición de los prototipos usando una variante de LVQ1. El método DSM inicia con la selección aleatoria de una submuestra y ajusta los prototipos en cada iteración usando toda la ME. Los prototipos se ajustan, si el prototipo más cercano clasifica incorrectamente a un patrón de la ME. Los prototipos más cercanos que tienen la misma clase del patrón se ajustan acercándolos al patrón, mientras, el prototipo que clasificó incorrectamente al patrón se aleja (Triguero & Herrera, 2012). El método DSM inicia con grandes cambios en la posición de los prototipos al inicio y en cada época la modificación que se realiza es menor, para refinar errores de clasificación más pequeños. El método DSM trata de mantener un equilibrio en la submuestra entre los prototipos y clases. El método DSM finaliza si no ocurre una clasificación errónea en el entrenamiento (Geva & Sitte, 1991b).

3.2.4 *Hybrid LVQ3 Algorithm (HYB)*

- Selección de una submuestra representativa
- Optimización de los prototipos

El método *Hybrid LVQ3 algorithm* (HYB) hace uso de diferentes métodos para seleccionar la submuestra de prototipos iniciales (Triguero et al., 2012). El método de selección de prototipos que obtiene los mejores resultados es *Support Vector Machines* (Khoshgoftaar et al., 2005). La submuestra de prototipos se ajusta con el método LVQ3. Para realizar el ajuste de los prototipos la ME se parte en dos. Una parte se usa para buscar los mejores parámetros para LVQ3, y la otra parte de la ME se usa para ajustar la posición de la submuestra de prototipos. (Triguero & Herrera, 2012), (Kim & Oommen, 2003a).

3.3 *Ajuste de la posición y generación de los prototipos con modelos vectoriales*

Los métodos LVQ de ajuste y generación de prototipos no calculan el número de prototipos iniciales. Estos métodos generan, eliminan y cambian la clase de los prototipos con diferentes reglas que se describen a continuación.

3.3.1 *Learning Vector Quantization with Training Count (LVQTC)*

- Selección de una submuestra
- Optimización de los prototipos
- Eliminación y cambio de clase

El método *Learning Vector Quantization with Training Count* (LVQTC) es una modificación del método LVQ1. El método LVQTC modifica la clase del prototipo y en algunos casos lo elimina. El método LVQTC fue propuesto por (Odorico, 1997) e inicia con una submuestra de un 5 % de patrones de la ME. A cada prototipo de la submuestra se le agrega un contador de entrenamiento. El contador de entrenamiento permite la modificación de la clase del prototipo o la eliminación. Los contadores de entrenamiento de cada patrón inician en cero y se modifican en cada época del entrenamiento. El ajuste, modificación y eliminación se hace acuerdo a la siguiente regla (Odorico, 1997).

$$\begin{cases} y_q(t+1) = y_q + \alpha_r(t)/Pot[x_p(t) - y_q(t)] & \text{si la clase}(x_p) == \text{clase}(y_q) \\ y_q(t+1) = y_q - \alpha_w(t)/Pot[x_p(t) - y_q(t)] & \text{si la clase}(x_p) \neq \text{clase}(y_q) \end{cases}$$

Donde α_r y α_w son dos tasas de aprendizaje diferentes y P_{tot} almacena el número de veces que los patrones de una misma clase han entrenado el prototipo. La eliminación de una neurona se realiza cuando alcanza número definido de clasificaciones erróneas en P_{tot} . Además, si P_{tot} alcanza un valor máximo en los contadores de entrenamiento de la clase incorrecta se crea un nuevo prototipo para esa clase (Triguero & Herrera, 2012).

3.3.2 *Learning Vector Quantization with Pruning (LVQPRU)*

- Selección de una submuestra
- Optimización de los prototipos
- Eliminación y cambio de clase

El método *Learning Vector Quantization with Pruning* (LVQPRU) propuesto por (Li et al., 2005) inicia con una submuestra grande de prototipos. Usa un gran número de prototipos iniciales porque calcular el número de prototipos de la submuestra una tarea difícil para cada muestra de datos. El método LVQPRU usa LVQ2 para ajustar la posición de los prototipos. Cada prototipo tiene un registro de cuantos patrones clasifica correctamente de la ME. Durante el entrenamiento, el método LVQPRU cambia la clase de los prototipos de acuerdo con el número de patrones que clasifica de la ME. Para eliminar un prototipo su registro de clasificación no debe ser mayor a cero, y la eliminación de este prototipo no debe aumentar el error de clasificación más de un veinte por ciento. La eliminación de prototipos se detiene cuando se alcanza un número mínimo de prototipos (Li et al., 2005).

3.4 *Vector Quantization (VQ)*

Los métodos de *Vector Quantization* (VQ) ajustan la posición de los prototipos de manera similar a los métodos LVQ. La diferencia entre estos métodos radica en la inicialización de la submuestra. Los métodos de VQ no requieren una submuestra inicial de prototipos. Los métodos de VQ crean los prototipos iniciales con toda la información de los patrones de la ME (Xie et al., 1993). Los métodos VQ crean un número definido de prototipos iniciales por clase llamados "codewords" (Yen et al., 2004). Los prototipos son separados en clases y no tienen ninguna interacción con patrones y prototipos de clases diferentes durante el ajuste de posición (Triguero et al., 2012).

3.4.1 Vector Quantization for Nearest Neighbor (VQ-NN)

- Selección de un centroide por clase como prototipo inicial
- Optimización de los prototipos

El método VQ-NN propuesto por (Xie et al., 1993) es un método basado en cuantificación vectorial y la regla NN. El método VQ-NN inicia con un prototipo por clase y los aumenta iterativamente. Sin embargo, calcular el número de prototipos por clase es difícil y los prototipos generados no forman buenos límites de decisión por la falta de interacción durante la cuantificación. Para clasificar los patrones de la ME usa la regla de bayes. Además, las clases de los prototipos son asignadas al finalizar el método VQ-NN. (Xie et al., 1993).

3.4.2 A Vector Quantization method for nearest neighbor classifier (AVQ)

- Selección de un centroide por clase como prototipo inicial
- Optimización de los prototipos

El método propuesto por (Yen et al., 2004) trata de solucionar el problema del número de prototipos con su método propuesto AVQ. El método AVQ inicia con solo un prototipo por clase, que es el centroide de cada clase. Los patrones de la ME son clasificados con los prototipos, y el grupo que tenga mayor error es dividido y el prototipo es reemplazado por otros dos prototipos que son los nuevos centroides. El error lo calcula de la suma de la regla de bayes y la regla NN. Los patrones son divididos usando el método LBG. El método AVQ termina cuando la generación de un nuevo prototipo no mejora la precisión (Yen et al., 2004).

3.5 Combinación de patrones de la ME para la generación de nuevos prototipos

Los métodos de GP que combinan los patrones de la ME para generar nuevos prototipos tratan de agrupar los patrones de diferentes formas. Estos métodos se describen a continuación.

3.5.1 Reduction by Space Partitioning (RSP3)

- Selección de un centroide por clase como prototipo inicial
- Generación de prototipos

El método *Reduction by Space Partitioning* (RSP3) propuesto por (Sánchez, 2004) inicia con un prototipo por clase. El método elige los patrones más alejados de la ME y creando un centroide para cada clase usando los patrones más cercanos. Los prototipos son usados para formar

agrupamientos dentro de la ME. Todos los agrupamientos son medidos para ver cual presenta un mayor diámetro. El agrupamiento con mayor diámetro se selecciona para buscar los patrones más lejanos para generar nuevos centroides. Este proceso se repite hasta que ningún agrupamiento formado por los prototipos contengan patrones de clases diferentes. El método RSP3 requiere una eliminación de patrones ruidosos, atípicos y mal etiquetados (edición). Si esto no se realiza el método generará un número mayor de prototipos (Sánchez, 2004).

3.5.2 *Bootstrap Technique (BTS3)*

- Selección aleatoria de un prototipo por clase
- Generación de prototipos

El método propuesto por (Hamamoto et al., 1997) basado en Bootstrap que tiene el nombre de BTS3. El método BTS3 combina localmente los patrones de la ME con Bootstrap. El método BTS3 inicia con la selección de un patrón aleatorio de la ME por clase. El siguiente paso es buscar los vecinos más cercanos al patrón para iniciar la generación de un nuevo prototipo. Los patrones de la ME no se deben seleccionar más de una vez. Además, el número de prototipos se debe determinar para saber cuándo detener el método (Hamamoto et al., 1997). La fórmula para la generación de los prototipos se muestra a continuación.

$$y_i = \frac{1}{k} * \sum_{j=1}^k x_i^j$$

Donde y_i es el prototipo y x_i^j son los patrones más cercanos (Triguero & Herrera, 2012).

3.5.3 *MixtGauss*

- Selección de una submuestra representativa
- Generación y optimización.

El método MixtGauss propuesto por (Sotoca et al., 2004) hace uso de la condensación adaptativa del modelo de mezclas de distribuciones gaussianas. El método MixtGauss asume una independencia estadística de las características (Lozano et al., 2006). El método MixtGauss inicia con una submuestra de prototipos. Los prototipos son modificados con valores aleatorios en las características para generar nuevos prototipos. La generación de nuevos prototipos se detiene cuando ninguna clase obtenga un aumento de rendimiento con respecto a una

iteración anterior (Sotoca et al., 2004). El proceso puede verse como un esquema de condensación adaptativo, en el que los prototipos del conjunto resultante corresponderán a los centros de los gaussianos finales (Triguero & Herrera, 2012).

3.6 División de la ME para la generación de prototipos

Otros enfoques que se tienen son los que dividen la ME para iniciar la generación de prototipos. Los métodos son divididos en dos grupos que inician solo con un prototipo por clase y los que inician con una submuestra de prototipos.

3.6.1 Pairwise Opposite Class-Nearest Neighbor (POC-NN)

- Selección de un centroide por clase como prototipo inicial
- Generación de prototipos

El método propuesto por (Raicharoen & Lursinsap, 2005) llamado POC-NN inicia con un prototipo por clase. El método POC-NN busca los límites de las clases de la ME para calcular el valor del límite de clase. El límite de clase divide la ME en región 1 y región 2. En cada región se buscan los patrones más cercanos al límite de clase para generar un prototipo para cada región. El siguiente paso es clasificar la ME usando los prototipos creados, y buscar la región con un mayor número de patrones mal clasificados. En la región seleccionada se busca un nuevo límite para generar nuevos prototipos. Sin embargo, en ME que tienen patrones con características iguales o similares con una clase diferente, el método tiene problemas para generar nuevos prototipos (Raicharoen & Lursinsap, 2005).

3.6.2 Self-Generating Prototypes (SGP)

- Selección de un centroide por clase como prototipo inicial
- Generación de prototipos
- Eliminación

El método propuesto por (Fayed et al., 2007) llamado SGP forma varios agrupamientos con los patrones de la ME. La media de cada agrupamiento se usa para generar los prototipos iniciales. El número de agrupamientos iniciales depende del número de clases de la ME. Los agrupamientos iniciales son divididos sucesivamente para formar nuevos agrupamientos y prototipos. Los patrones de la ME pueden cambiar a otro agrupamiento, si el prototipo más

cercano pertenece a otro grupo. Además, si algún patrón de la ME es clasificado incorrectamente el agrupamiento se divide, y se eliminan los patrones mal clasificados para formar un nuevo agrupamiento y calcular dos nuevos prototipos (Fayed et al., 2007).

3.7 Generación y optimización de prototipos con modelos evolutivos

Los métodos de generación con modelos evolutivos inician con solo un prototipo por clase que usarán para clasificar la ME. El siguiente paso es generar nuevos prototipos usando los prototipos iniciales. Los prototipos se generan y optimizan siguiendo reglas específicas de cada método de GP.

3.7.1 Evolutionary Design of Nearest Prototype Classifiers (ENPC)

- Selección aleatoria de un prototipo por clase
- Generación y optimización de prototipos
- Eliminación de prototipos

El método ENPC propuesto por (Fernández & Isasi, 2004) trata de encontrar el número óptimo de prototipos por clase. El método ENPC se basa en un proceso de evolución. El método ENPC inicia con un prototipo por clase e introduce nuevos prototipos usando operadores de reproducción, mutación, lucha, eliminación y movimiento. El operador de reproducción busca cubrir nuevas regiones en el espacio de características. La clase de los prototipos puede cambiar de acuerdo con los patrones más cercanos a este.

1. El método inicia con un prototipo por clase seleccionado aleatoriamente.
2. Evalúa los prototipos seleccionados clasificando la ME, si es necesario la clase del prototipo es actualiza.
3. La reproducción del prototipo solo se realiza usando ruleta, y si el descendiente es mejor que el prototipo padre se conserva. Además, si el prototipo tiene como vecinos más cercano patrones de otra clase se crea otro prototipo con su misma clase.
4. Los prototipos realizan una lucha para mantener más patrones dentro de su región. Los prototipos más alejados del prototipo y más cercano a otro cambian de prototipo. En cambio, si los prototipos pertenecen a clases diferentes la lucha

por los prototipos se realiza con ruleta. La división con ruleta trata de ser equilibrada para dividir patrones y prototipos de la misma clase.

5. Los prototipos son movidos al centro de su región que forman sus patrones más cercanos.
6. Los prototipos son eliminados si dentro de su región no contienen un número determinado de patrones.
7. El método finaliza al cumplir un número de iteraciones, alcanzar una precisión o un número de prototipos establecido.

En el método ENPC todos los operadores están integrados y no se pueden modificar por otro operador. Esto permite que el método no necesite de muchos parámetros para iniciar su ejecución. Los prototipos son representados en una matriz que contiene las características, la clase a la que pertenece y el número de patrones que clasifican. En la figura 3.1 se muestra un ejemplo de la matriz del método ENPC (Fernández & Isasi, 2004).

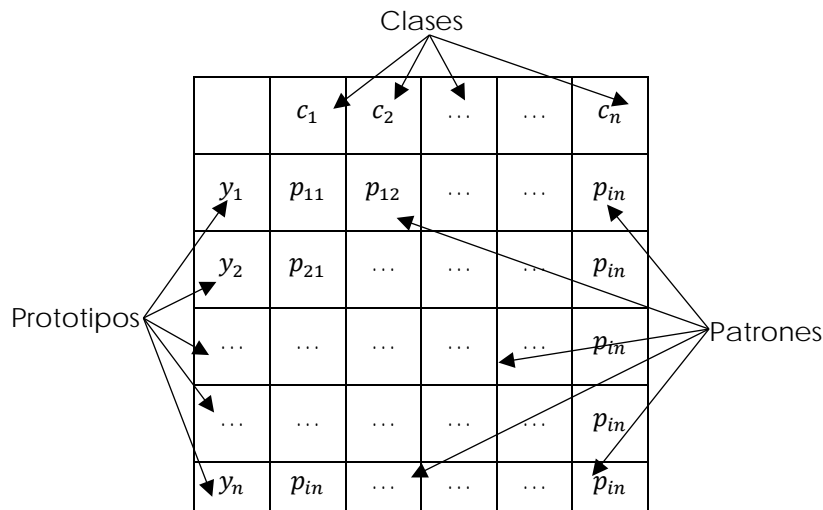


Figura 3.10 Representación de los prototipos de ENPC.

3.7.2 Iterative Prototype Adjustment based on Differential Evolution (IPADE)

El método IPADE propuesto por (Triguero et al., 2010) hace uso de evolución diferencial o fórmulas matemáticas. El método IPADE inicia solo con un prototipo por clase. Cada individuo del método de IPADE codifica solo un prototipo, y usa la regla NN para realizar la clasificación de la ME. La generación y optimización de los prototipos se realiza con operadores de mutación y cruza. La optimización inicia con la creación de un vector mutante para cada prototipo. El vector mutante se cruzará con su respectivo prototipo. El prototipo obtenido se

usa para clasificar los patrones de la ME, y si se obtiene una mejora el prototipo reemplaza al prototipo padre, pero si no se obtiene una mejora es descartado. Para agregar un nuevo prototipo el método IPADE selecciona un patrón aleatorio de la ME. El patrón seleccionado es agregado a la muestra de prototipos, para realizar una iteración usando los operadores de mutación y cruza, si se obtiene una mejora el patrón se agrega a la muestra de prototipos. En caso contrario, el patrón se elimina y la clase se marca como no optimizable (Triguero et al., 2010).

El algoritmo de IPADE es el siguiente:

- 1) GS = Inicialización (ME)
- 2) Optimización de (GS, ME)
- 3) Precisión_Global = Evaluar (GS, ME)
- 4) Registrar_clase [0... Ω] = optimizable
- 5) **MIENTRAS** Precisión_Global \neq 1.0 o todas las clases no son optimizables
 - a) Menor_precisión = ∞
 - b) **PARA** (i = 1 a Ω i)
 - SI** Registrar_clase [i] == Clase_de_precisión optimizable [i] = Evaluar (GS, patrones de clase i en ME) ii)
 - SI** Clase_de_precisión [i] < Menor_precisión
Menor_precisión = Clase_de_precisión [i]; clase_de_destino = i;
 - c) GStest = GS u Muestra_aleatoria_para_clase (ME, clase_de_destino)
 - d) Optimización DE (GStest, ME)
 - e) Precisión_MP = Evaluar (GStest, TR) f)
 - SI** prueba_de_precisión > Precisión_Global
Precisión_Global = precisionTest; GS = GStest
 - f) **de otro modo** Registrar_clase [clase_de_destino] = no optimizable

3.8 Ajuste de la posición de los prototipos con modelos evolutivos

Los métodos evolutivos para ajustar la posición de los prototipos inician con una submuestra de la ME. Generalmente es del cinco por ciento y se trata de mantener un equilibrio de las clases. El ajuste de la posición se realiza con un vector del mismo tamaño que los prototipos que suma o resta valores a las características de los prototipos.

3.8.1 *MSE*

- Selección de una submuestra representativa
- Optimización de los prototipos
- Eliminación

El método MSE propuesto por (Decaestecker, 1997) inicia con un conjunto de prototipos de la ME. Los prototipos iniciales se buscan de manera no supervisada, y se eligen por clase con el algoritmo de agrupación *k-means*. Los centroides encontrados por *k-means* se usan como los prototipos iniciales, generalmente son de diez a veinte por clase. Con esto busca que cada clase y agrupamiento este representado por un prototipo (Triguero & Herrera, 2012). La optimización de los prototipos se realiza con descenso de la gradiente. La optimización permite realizar grandes cambios en los prototipos al inicio, y conforme avanza la optimización los cambios son más pequeños. Para eliminar los prototipos no representativos hacen uso de la regla con $k=3$, mientras, la clasificación no se vea afectada. El método MSE presenta problemas en ME con una deficiente construcción o ruido.

3.8.2 *An Adaptive Michigan Approach PSO (AMPSO)*

- Selección de una submuestra de prototipos
- Optimización de prototipos
- Eliminación

El método AMPSO propuesto por (Cervantes et al., 2007) usa *Particle Swarm Optimization* (PSO) con un enfoque diferente de optimización. La optimización PSO se inspira en la naturaleza y la analogía social, donde cada miembro del enjambre ayuda a encontrar la solución a un problema. En la optimización PSO tradicional una partícula representa la solución potencial al problema. En cambio, en el método AMPSO que usa un enfoque Adaptativo de Michigan y cada partícula representa solo una parte de la solución, por lo que cada partícula representa un prototipo y su clase. Por lo que el tamaño de cada partícula es igual al número de características y clase de los patrones de la ME. La codificación propuesta por AMPSO es más sencilla y flexible al número de prototipos que PSO. El método inicia con una selección de una submuestra aleatoria de los prototipos iniciales. Los prototipos son optimizados con PSO que permite la eliminación y reproducción en caso de ser necesario. Sin embargo, la clase de las

partículas no cambia durante la optimización de la posición. Para clasificar los patrones de la ME hacen uso de la regla NN (Cervantes et al., 2007).

El pseudocódigo del algoritmo AMPPO es:

1. Inicializar Swarm. (La dimensión de las partículas es igual al número de atributos)
2. Inserte N prototipos de cada clase de la ME.
3. Si el Criterio de parada no se cumple:
 - a) Verifique la reproducción y eliminación de partículas.
 - b) Para cada partícula haga:
 - i) Calcule la aptitud local.
 - ii) Calcule su siguiente posición
 - c) Mueva las partículas.
 - d) Asignar clases a los prototipos de la ME utilizando las partículas más cercanas.
 - e) Evaluar la precisión de la clasificación.
4. Eliminar de la mejor Swarm encontrada, las partículas (prototipos) que pueden eliminarse sin reducir la precisión de clasificación.

3.8.3 Particle Swarm Optimization for prototype reduction (PSO)

- Selección de una submuestra de prototipos
- Optimización de prototipos

El método PSO propuesto por (Cervantes et al., 2009) se basa en la optimización PSO para encontrar la posición de una muestra de prototipos. El método PSO inicia con un 5% de patrones de la ME. Las partículas contienen la solución completa, es decir cada partícula contiene todos los prototipos. Para mover a los prototipos hacen uso de un vector con las mismas dimensiones que un patrón. El vector controla la velocidad en la que los prototipos son movidos dentro del m-espacio de caracterizas. Al inicio PSO realizan cambios grandes en los prototipos, pero cada que se acerca al número final de iteraciones los cambios son menores (Cervantes et al., 2009)

1. Inicialización: seleccione un número aleatorio de prototipos de la ME.
2. Mientras se alcanza la condición final:
 - a. Evaluar partículas.
 - b. Procedimiento de actualización: Para cada partícula, calcule su velocidad y actualice el posicionamiento, de acuerdo con las siguientes ecuaciones:

$$v_1 = wv_1 + c_1 * Rand()(p_i - x_i) + c_2 Rand()(p_g - x_i)$$
$$x_i(t + 1) = x_i(t) + v_i$$

donde i varía de 1 al tamaño de la población; c_1 y c_2 se denominan constantes de aceleración; p_i es la partícula actual (una muestra de prototipos) y p_g son las mejores posiciones; $Rand()$ es una función aleatoria en el rango $[0,1]$; w es el peso de inercia.

3.9 Resumen

Los métodos de GP se pueden clasificar de diferentes formas. Una de las posibles clasificaciones se presentó anteriormente con siete grupos. Los métodos de cada clase comparten de una forma general el proceso de ajuste o la generación de prototipos. En los métodos de ajuste y generación que hacen uso de VQ hacen las modificaciones con base en los patrones de la ME. Los patrones de la ME se usan para atraer o alejar a los prototipos dentro del espacio de características. Además, dentro de estos métodos VQ se tienen métodos con parámetros que permiten generar y eliminar prototipos. La eliminación y generación de estos métodos les permite iniciar con una submuestra no representativa. Los métodos de GP que hacen uso de enfoques evolutivos también se pueden dividir en dos clases diferentes. En la primera clase los métodos realizan un ajuste con base en vectores de velocidad. Los vectores de velocidad tienen las mismas características que los prototipos y patrones, estos vectores son sumados o restados en los prototipos para moverlos dentro del espacio de características. La segunda clase de los métodos evolutivos de GP tienen la capacidad de generar nuevos prototipos. Los prototipos son generados con operadores especializados de selección, cruce y mutación. Mientras, en los métodos de generación de nuevos prototipos se generan con los patrones de la ME. Los patrones son usados para calcular nuevos centroides o realizar una mezcla de varios patrones al obtener el promedio. El centroide o promedio se elige un nuevo prototipo. En la figura 3.11 se muestra el esquema de la clasificación propuesta.

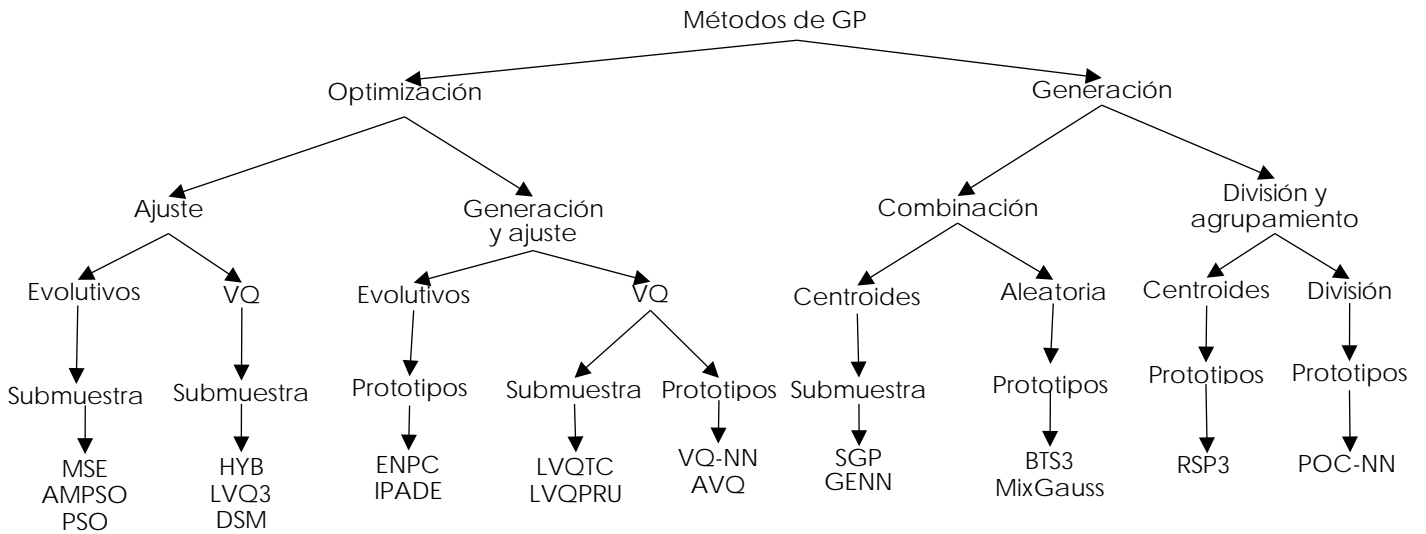


Figura 3.11 Clasificación propuesta de los métodos de GP.

Además, se incluye el mapa de generación de prototipos que fue propuesto en el trabajo de (Triguero et al., 2012) en donde divide en cuatro categorías principales los métodos de GP. Cada categoría tiene diferentes subcategorías que aportan información del tipo de inicialización, generación y optimización. En la figura 3.12 se muestra el mapa de generación de prototipos propuesto por (Triguero et al., 2012). Además, se incluye un resumen de las subcategorías del mapa de generación de prototipos.

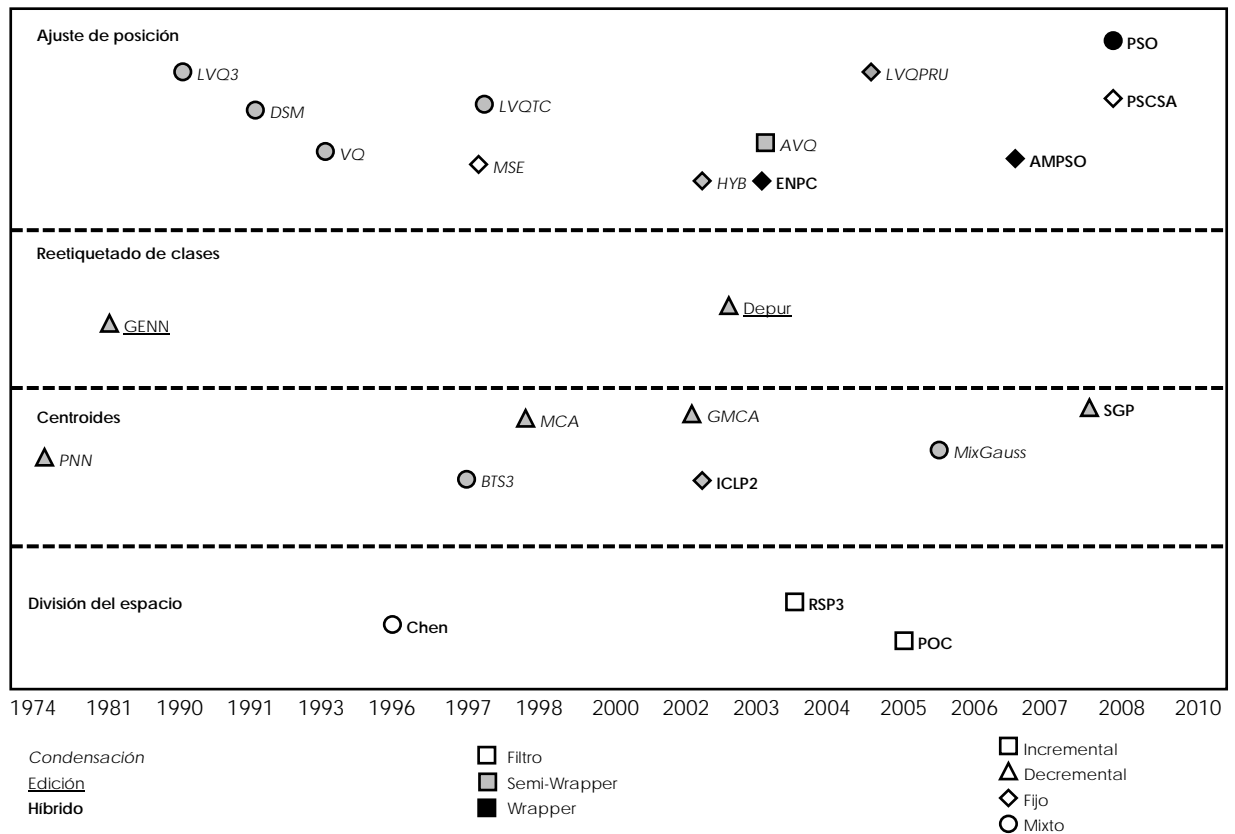


Figura 3.12 Mapa generación de prototipos propuesto por (Triguero et al., 2012)

Generación de prototipos:

- Ajuste de posicionamiento: los prototipos se generan de la selección de una submuestra de la ME. La submuestra de prototipos se modificará y ajustará en el espacio de características sumando o restando cantidades a las características de los prototipos.
- Reetiquetado de clases: modifican la clase los patrones o prototipos que puedan pertenecer a otras clases (ruidosos), sin eliminar ningún patrón.
- Centroides: los prototipos se generan de la combinación de varios patrones similares, dando como resultados los centroides. Los algoritmos basados en centroides obtienen una alta reducción que da como resultado pérdidas en la precisión.
- División del espacio: los prototipos se generan de la división del espacio de características (regiones) y en cada región se asignará a un prototipo representativo, que reemplazará a todos los patrones de la región.

Estado del Arte

Conjunto de prototipos obtenidos:

- Condensación: obtiene prototipos cerca de los límites de decisión y una generalización alta en la ME. Sin embargo, la generalización en la MP es menor. Además, la reducción es alta.
- Edición: obtiene prototipos alejados de los límites de decisión, al modificarlos o eliminarlos. La generalización en la MP es mayor en comparación con la ME. La reducción obtenida es menor.
- Híbrido: trata de obtener un número reducido de prototipos al combinar la modificación de prototipos cercanos y alejados de los límites de decisión.

Evaluación de la búsqueda:

- Filtro: La regla NN no es usada para evaluar los prototipos en su lugar se usa otra heurística. La evaluación puede ser más rápida pero no garantiza la precisión.
- Semiwrapper: la regla NN es usada para evaluar los prototipos y otros criterios son usados para tomar una decisión en la evaluación. Es un enfoque intermedio de eficiencia y precisión.
- Wrapper: la regla NN es usada para la evaluación y la toma de decisiones de la generación de los prototipos. Es un enfoque para obtener la mejor precisión en la MP. Sin embargo, cada evaluación puede ser costosa.

Tipo de reducción:

- Incremental: inicia con una submuestra de prototipos o con algunos prototipos representativos por clase. En cada iteración se agregan nuevos prototipos o se modifican. Es una de las técnicas más rápidas y que requiere de menos almacenamiento. Además, le permite establecer el número de prototipos necesarios para la ME. Sin embargo, un número elevado de prototipos producirá un sobreajuste.
- Decremental: inicia con la ME como submuestra de prototipos y en cada iteración reducirá y modificará los prototipos. La reducción y modificación se realiza con diferentes operaciones, como fusionar, mover, cambiar la clase o eliminar prototipos. La principal ventaja de esta reducción es que tiene disponible todos los patrones de la ME. Sin embargo, esto aumenta el costo computacional.
- Fijo: inicia con un número definido de prototipos en relación con un porcentaje de la ME. Uno de los principales inconvenientes de esta reducción es la variación del número

Estado del Arte

de prototipos para cada ME. Además, solo se enfoca en aumentar precisión de la clasificación.

- Mixta: inicia con una submuestra de prototipos seleccionados de la ME, que luego agregara, modificara y eliminara prototipos en la submuestra. Este tipo de reducción combina las ventajas de las anteriores reducciones, lo que le permitirá corregir los problemas que tenga la submuestra. Sin embargo, presenta problema de sobreajuste y en general tienen un alto costo computacional.



Capítulo 4.

Método Propuesto

Recordando el problema planteado

¿Cómo obtener prototipos útiles a partir de una ME, que se puedan usar para clasificar nuevos patrones?

En este capítulo se presenta el método propuesto para la generación de prototipos representativos de Muestra Entrenamiento (ME), para dar la solución al problema planteado. Cada paso del método propuesto es descrito con un diagrama y una explicación de las tareas realizadas en cada paso.

4.1 Descripción del método propuesto

Los métodos de GP expuestos en el capítulo 3 muestran los diferentes enfoques para la generación de prototipos. Cada método de GP del estado del arte usa diferentes reglas y una métrica definida para validar los prototipos. Por tal motivo, se propuso un nuevo método que permita usar diferentes métricas de evaluación y el clasificador basado en la regla NN. Las muestras de datos que se usan en el método deben estar normalizadas y divididas en ME y MP.

4.1.1 Normalización

Los métodos de GP buscan en los espacios de características de la muestra de datos. Cada muestra de datos tiene patrones con m -características y el espacio de características a explorar que puede ser muy grande. Una forma de reducir el espacio de características de los patrones es normalizando los valores de las características. Las características de los patrones de la muestra de datos se normalizarán con valores de 0 a 1. La normalización de las características se hace en todos los métodos de GP del estado del arte (Nanni & Lumini, 2009).

4.1.2 Generación de las ME y MP

Los métodos de GP requieren de muestras de datos con patrones diferentes para generar y validar los prototipos. Los patrones que se usan para generar los prototipos no deben usar para validarlos y viceversa. Para determinar los patrones que forman la ME y MP se usa la validación cruzada. El número de ME y MP generalmente usado en los métodos de GP en el estado del arte es de diez. Lo que permite una comparación de los resultados obtenidos con los resultados de los métodos del estado del arte.

4.2 Pasos del método propuesto

Los pasos del método propuesto se muestran en la figura 4.1 con 3 pasos principales. El método propuesto de GP inicia con una ME que puede tener n -patrones con n -clases y n -características. En el paso de generación de prototipos se optimizan y evalúan los prototipos con la ME. El último paso es la evaluación de la MP con los prototipos generados para la evaluación de la generalización alcanzada por los prototipos. Además, de evaluar la reducción alcanzada en comparación con la ME.

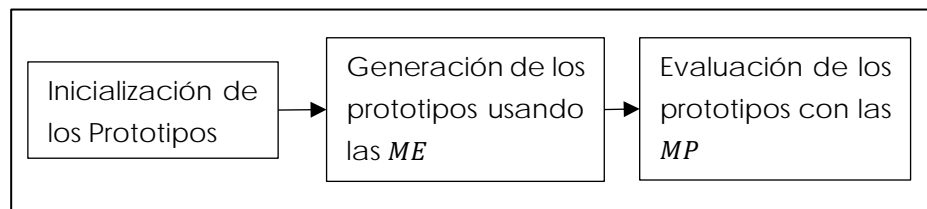


Figura 4.1 Diagrama del método propuesto.

La reducción y la generalización son los puntos más importantes en los métodos de GP por lo que se busca generar prototipos útiles. Los prototipos útiles se definen como prototipos con una alta precisión en la clasificación y una baja cantidad número de prototipos (Fernández & Isasi, 2004). Sin embargo, una alta precisión y un número bajo de prototipos entra en conflicto de manera natural. Esto se produce porque a mayor número de prototipos la precisión aumenta y viceversa un menor número de prototipos reduce la precisión (Bezdek & Kuncheva, 2001). El método propuesto se diseñó para generar prototipos útiles con diferentes métricas y un clasificador de la regla NN.

4.2.3 Generación de los prototipos

La generación de los prototipos se realiza para cada ME obtenida en la validación cruzada. La MP no se utiliza en esta etapa de generación de prototipos. El método se aplica para cada ME_1, ME_2, \dots, ME_n y se obtiene un número diferente de prototipos de cada una de las ME. Uno de los problemas de los métodos de GP es saber el número de prototipos iniciales. Además, del número de prototipos se debe considerar la clase que cada prototipo para no generar un desequilibrio en los prototipos. En los métodos de GP del estado del arte las clases de los prototipos son definidas como en SGP, LVO3, HYB, etc. y en otros como AMPPO, ENPC, IPADE, etc. se modifica de acuerdo con los patrones cercanos de la ME. El método propuesto inicia con dos, tres o cuatro prototipos dependiendo del número total de clases de la ME. Además, los prototipos iniciales son generados aleatoriamente. Los prototipos iniciales son usados para generar nuevos prototipos y solo se conservan los prototipos con una clase útil que definen los patrones de la ME y métrica de evaluación. En la figura 4.2 se muestra el diagrama del proceso completo de generación de prototipos que se propone en esta tesis.

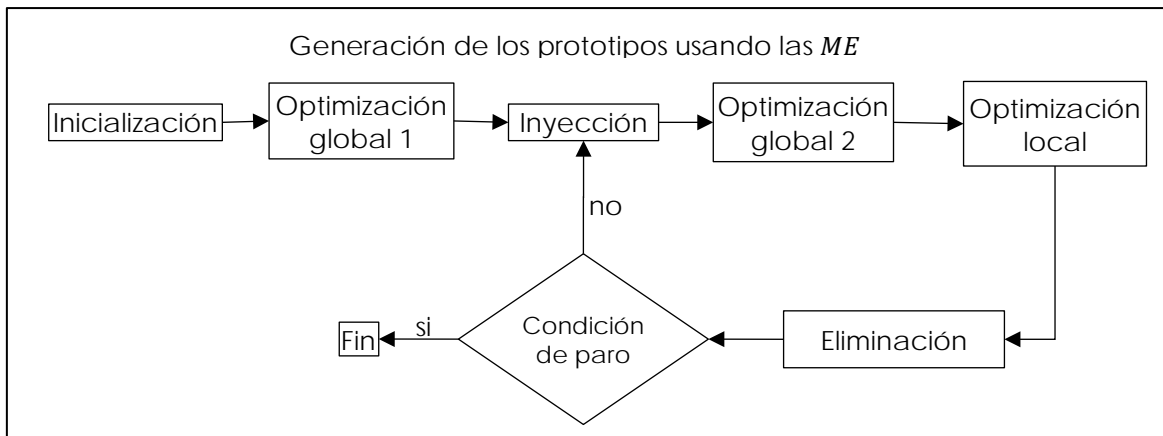


Figura 4.2 Diagrama de procesos de generación de prototipos usando la ME.

Los procesos de inicialización, optimización global, inyección, optimización local, eliminación y condición de paro del diagrama de procesos de la figura 4.2. se describen a continuación usando la muestra de datos “banana”.

a) Inicialización

El número de prototipos iniciales está relacionado con el total de clases de la ME. Sin embargo, esto no garantiza que cada clase tenga un prototipo que la represente. La clase y características de cada prototipo no se definen debido a que se generan aleatoriamente en el proceso de inicialización.

Como ejemplo, en la tabla 4.1 se muestra seis prototipos generados aleatoriamente de la ME “banana”. En la figura 4.3 se muestran el espacio de características ocupado por los patrones de la ME “banana”.

Tabla 4.1 Prototipos iniciales aleatorios de la ME.

No.	Característica uno	Característica dos	Clase
1	0.605533	0.555933	1
2	0.850655	0.943022	0
3	0.795159	0.252154	1
4	0.382322	0.29352	0
5	0.549843	0.115958	1
6	0.024936	0.148586	1

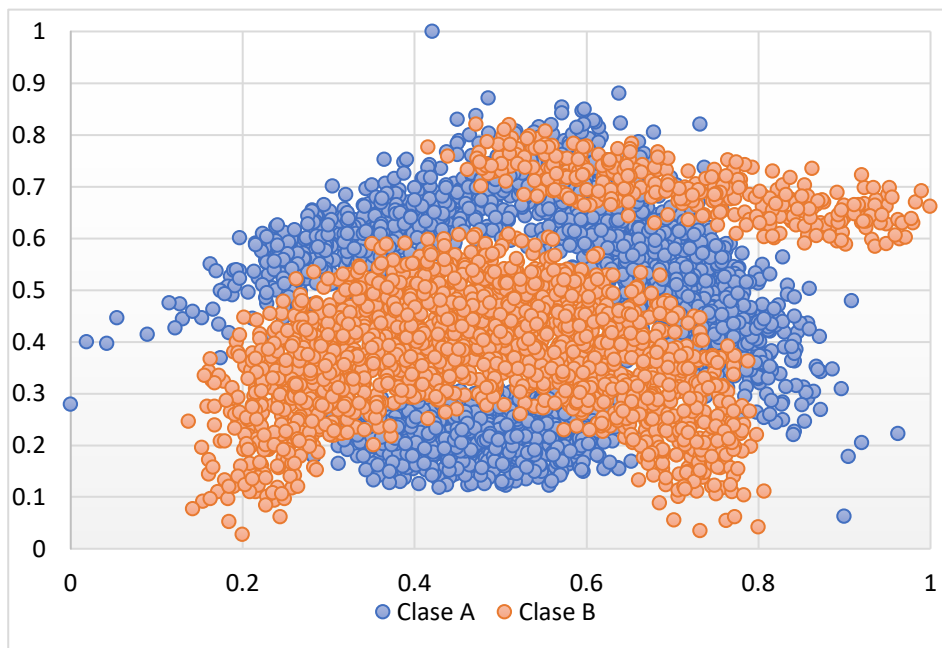


Figura 4.3 Representación gráfica de la muestra de datos "banana".

Para mostrar el espacio de características ocupado por los prototipos iniciales y compararlos con los patrones de la ME, en la figura 4.4 se muestran 400 prototipos iniciales.

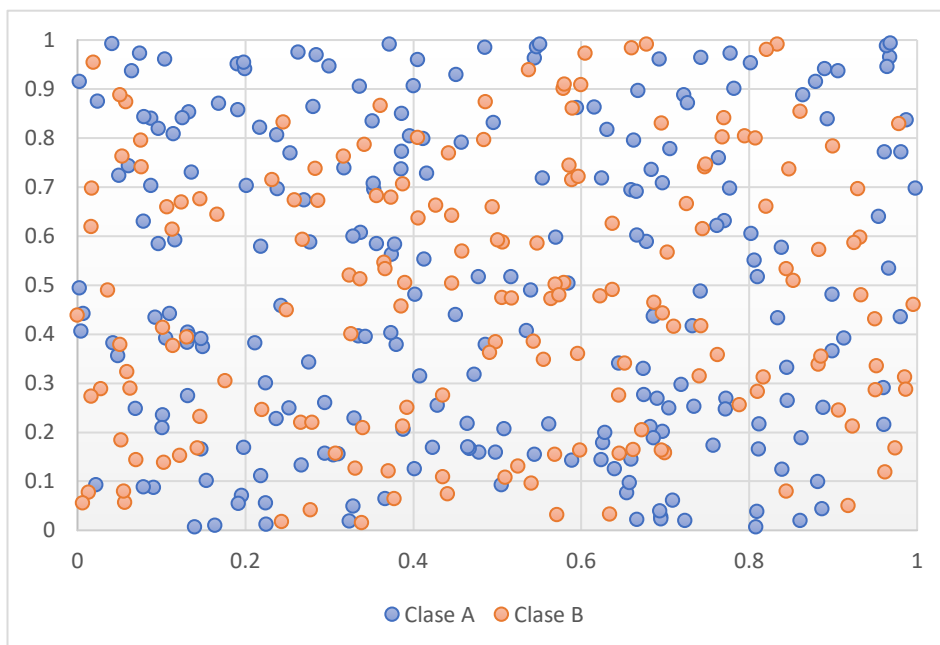


Figura 4.4 Representación gráfica de los prototipos iniciales

Método Propuesto

Los prototipos iniciales ocupan diferentes lugares en el espacio de características de la ME "banana". Para encontrar, validar y colocar dentro del espacio de características representativos de la ME "banana" se debe iniciar la optimización global.

b) Optimización global

El problema de cómo optimizar los prototipos de la figura 4.5 y obtener los mejores dentro del espacio de características es uno de los problemas principales de los métodos de GP. La optimización debe tener la capacidad de validar y colocar los prototipos en la mejor posición dentro del espacio de características. Algunos métodos de GP evolutivos del estado del arte hacen uso de operadores de reproducción, mutación y eliminación para buscar dentro del espacio de características de la ME. Sin embargo, existen otros algoritmos evolutivos como los Algoritmos Genéticos (AG); que solo se han usado para la selección de los prototipos iniciales (Nanni & Lumini, 2009). Para realizar la optimización global el método propuesto usará un AG que tratará de buscar dentro del espacio de características de la ME, la mejor posición de los prototipos iniciales.

1) Función de aptitud

El problema de validar los prototipos con la ME se puede realizar de diferentes maneras. La validación de los prototipos guiará el proceso de generación de los prototipos. La validación se realiza con la función de aptitud del AG. La función de aptitud se compone por la regla NN y de una métrica de evaluación. La métrica propuesta es Medida f Ponderada (MfP). Además, se calcula la distancia que tiene cada prototipo con el patrón más cercano de la ME y se suma a una distancia total. Al final se suma la distancia total y el error de clasificación obtenido de la métrica de evaluación.

2) Operador de selección

El operador de selección usado por el AG es ruleta. También, se hace uso de un operador elite para mantener en cada generación el cromosoma con la aptitud más alta.

3) Operador de cruce.

El operador propuesto para la generación de nuevos cromosomas es cruce uniforme.

4) Operador de mutación

El operador propuesto para la mutación es generación de base- n para mantener una variedad genética en los cromosomas.

Como ejemplo, en la figura 4.5 se muestra el espacio de características ocupado por los prototipos iniciales al finalizar la optimización global. Además, se muestran los mejores prototipos de cada clase con la clase AA y la clase BB señalados con un círculo color rojo. Estos son los prototipos que obtienen la mejor evaluación de la función de aptitud del AG. Estos prototipos se guardan en la Submuestra de Prototipos (SMP).

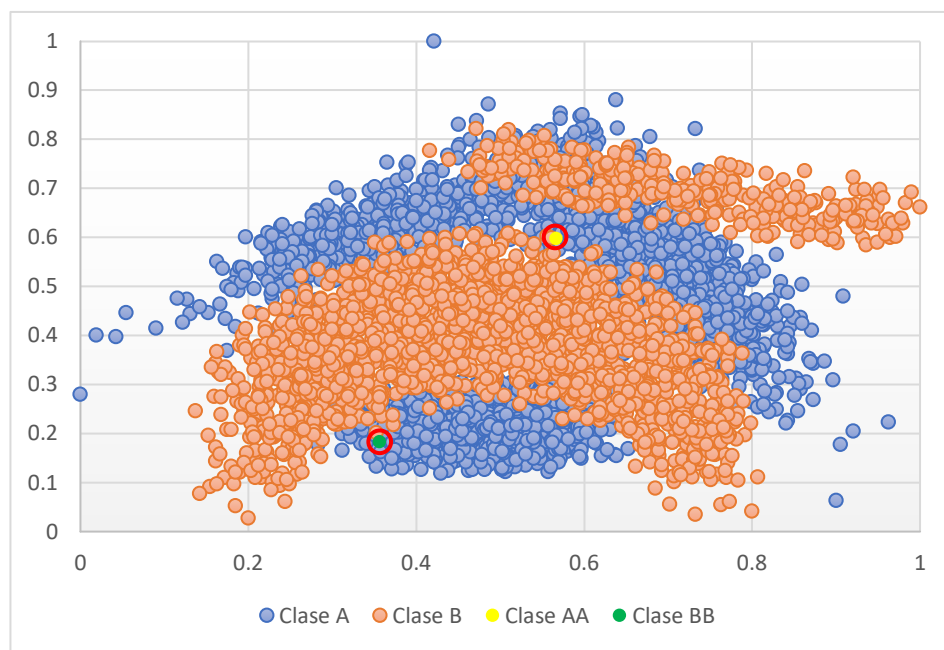


Figura 4.5 Representación gráfica de los prototipos después la primera optimización global.

c) Inyección

La SMP generada en el paso anterior se usa para clasificar toda la ME y obtener los patrones que se clasifican incorrectamente de la ME. Algunos de los patrones de la ME que son clasificados incorrectamente se usarán para reemplazar a todos los prototipos de la optimización global. Esto se hace para explorar nuevos espacios de características de la ME y mejorar la clasificación con nuevos prototipos.

Método Propuesto

Como ejemplo, en la figura 4.6 se muestran algunos de los prototipos mal clasificados de la ME “banana”. Para esta muestra con dos clases se inyectan 400 prototipos mal clasificados. Esto genera que el método propuesto inicie una nueva optimización global. La siguiente optimización global es similar a la anterior y solo agrega un nuevo parámetro que considerar.

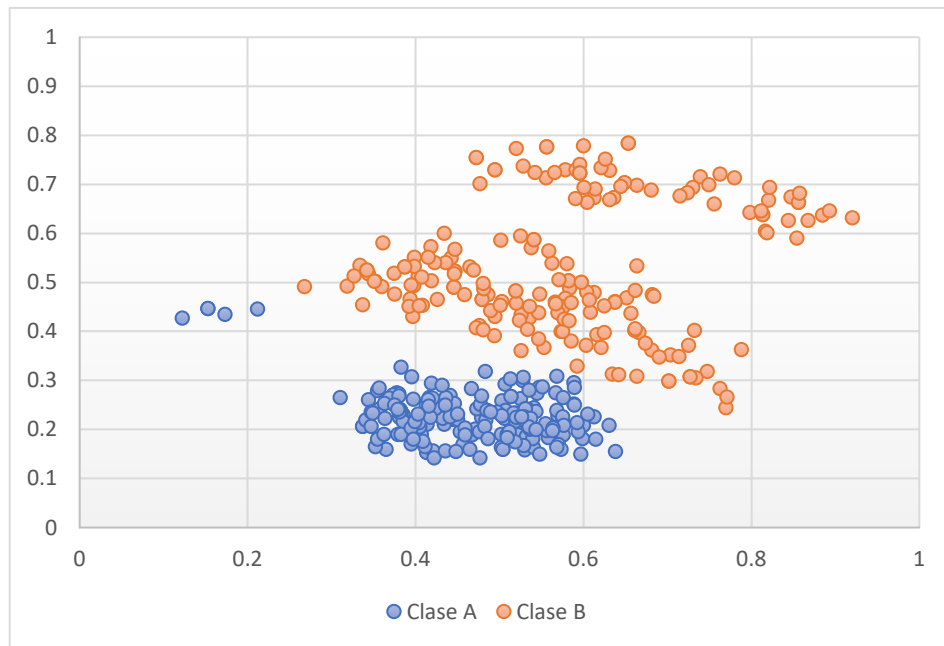


Figura 4.6 Patrones mal clasificados de la ME inyectados en los prototipos.

La exploración del método propuesto tiene como propósito iniciar una generación de prototipos, que no se ha explorado con los prototipos actuales. El número de patrones inyectados se puede modificar para inyectar solo uno o más patrones por clase de la ME.

d) Optimización global 2

La optimización global 2 del método propuesto optimizará los prototipos de la figura 4.6 para encontrar la mejor posición dentro del espacio de características. Usa la misma optimización global 1 del AG sin cambiar ningún operador. La única diferencia con la anterior optimización es la consideración de los prototipos de la SMP al momento de evaluar los prototipos de la optimización global 2.

Como ejemplo, en la figura 4.7 se muestran los prototipos al finalizar la optimización global 2 con un total de 4 prototipos. Los mejores prototipos de la optimización global 2 son señalados

Método Propuesto

con un círculo color rojo. Estos prototipos se agregan a la SMP para incrementar el número de prototipos optimizados. Además, los prototipos de la optimización global 1 son señalados con color verde. El siguiente paso del método propuesto es buscar localmente la mejor posición para los cuatro prototipos de SMP.

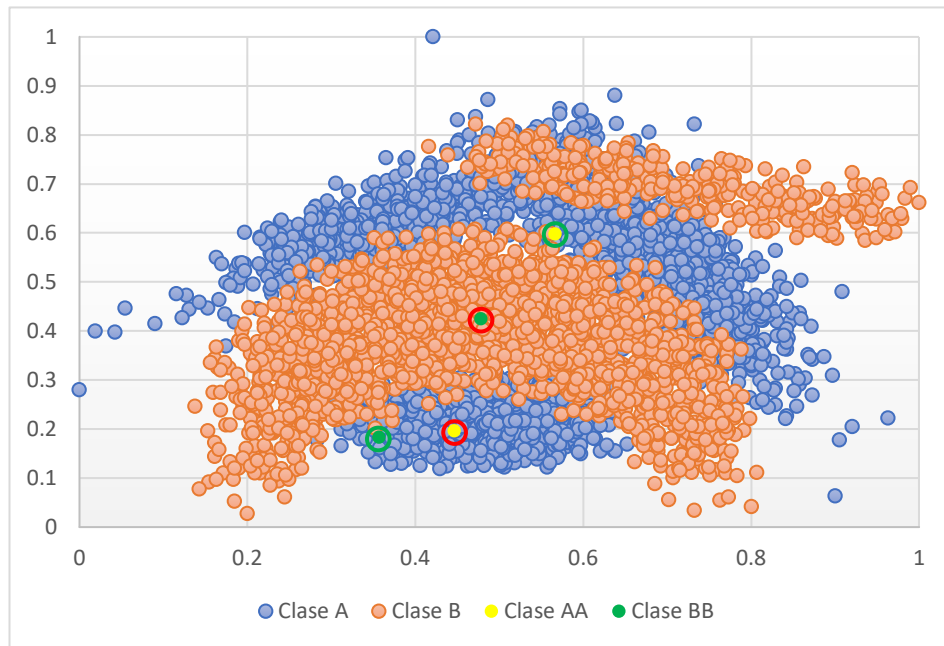


Figura 4.7 Representación gráfica de los prototipos después la optimización global 2.

e) Optimización local

La optimización local usa los prototipos que forman hasta el momento la SMP y tratará de mejorar su posición dentro del espacio de características. La optimización local divide los prototipos de la SMP en varias partes que se representan de la siguiente manera $SMP = \{smp_1, smp_2, \dots, smp_n\}$. Los prototipos de smp_1 exploraran su espacio de características cercano y al encontrar una mejor posición en el espacio de características se guardará en smp_1 . Cuando finaliza la optimización de smp_1 se continua iterativamente con las partes restantes de smp_n . El proceso se repite varias veces para cada una de las partes smp_n . La optimización local usa la misma optimización global 1 y 2 del AG sin cambiar ningún operador y parámetro.

Método Propuesto

Como ejemplo, los prototipos de la figura 4.7 señalados con los círculos color verde y rojo tienen una optimización local. En la figura 4.8 se muestran el movimiento de los prototipos dentro de la ME de "banana". La flecha color rojo muestra la posición inicial y la posición final de cada prototipo. Los prototipos de la SMP inician clasificando correctamente solo 3471 patrones de la ME. Para la "Clase A" se clasifican 2077 patrones correctamente y para la "Clase B" se clasifican correctamente 1394 patrones. Al finalizar la optimización local los prototipos de la SMP clasifican correctamente 3619. Para la "Clase A" ahora clasifican 2527 patrones correctamente y para la "Clase B" se clasifican 1092 patrones. La mejora de los prototipos de la SMP es de un 3.1 por ciento. Solo un prototipo de la SMP cambio de clase y fue movido considerablemente. El prototipo mejoro la clasificación de la clase A, pero clasifíco menos patrones de la clase B y como resultado aumento la precisión en toda la ME.

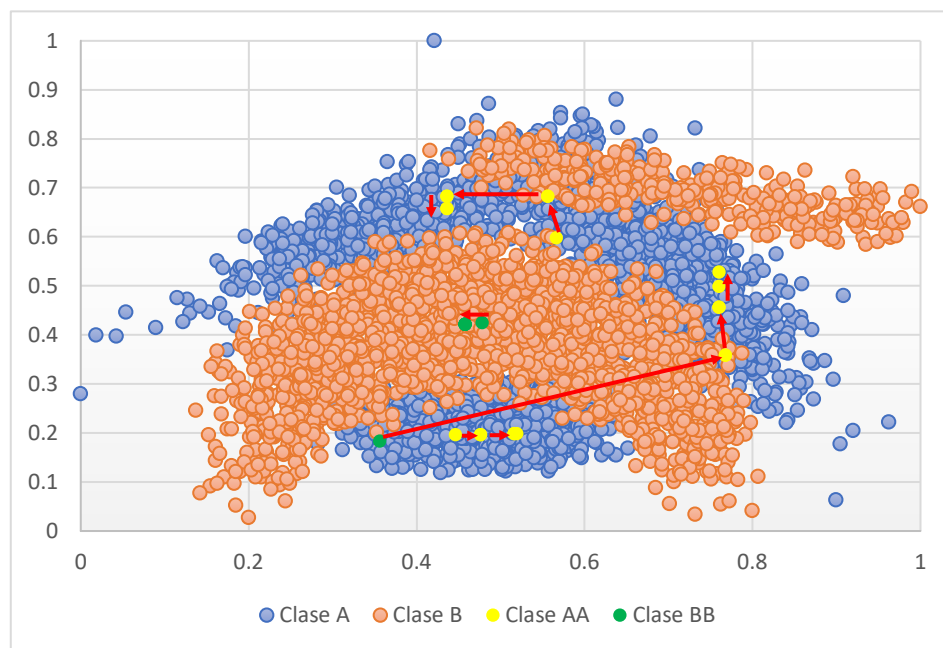


Figura 4.8 Posición inicial y final de los prototipos de la SMP al finalizar la optimización local.

Uno de los problemas del método propuesto es mantener prototipos útiles dentro de la SMP, es decir, se eliminan prototipos que no mejoran la clasificación.

f) Eliminación

La eliminación de prototipos de la SMP solo se realiza si el tamaño es mayor a cinco y menor a treinta. Para decidir qué prototipos no son necesarios se usa la regla NN, la métrica de evaluación es medida f ponderada (MfP) y una submuestra de SMP. Para crear la submuestra se elimina un prototipo y los prototipos restantes se usan para clasificar la ME. Si la eliminación de este prototipo mejora o mantiene la precisión de clasificación, este prototipo se elimina de la submuestra. La submuestra reemplaza a SMP, y se inicia una nueva eliminación de prototipos. Si en cambio, la clasificación empeora el prototipo se agrega de nuevo a la submuestra, y se procesa el siguiente prototipo para seguir el mismo proceso de eliminación. Este proceso de eliminación puede eliminar varios prototipos o no eliminar ninguno.

Los pasos para eliminar un prototipo de la SMP se muestran a continuación:

- Se crea Submuestra (S) = SMP
- Se inicia $i = 0$
- Se elimina de S un *prototipo _{i}*
- Se clasifica con S la ME.
- Si $S \geq$ SMP
 - $i=0$
- Si no
 - Se agrega *protitipo _{i}* a S
 - $i=i+1$
- Al finalizar $SMP = S$

Este proceso se repite para todos los prototipos de la SMP.

Como ejemplo, en la figura 4.9 se muestra los prototipos de la SMP y el prototipo que se van a eliminar se marca con un círculo de color rojo.

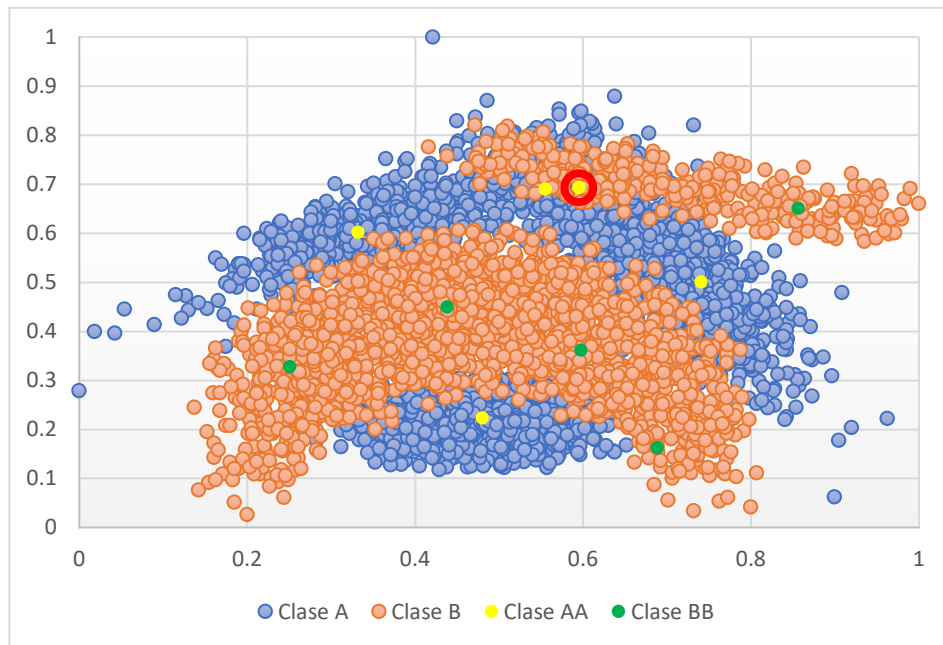


Figura 4.9 Prototipos que forman la SMP y el prototipo a eliminar.

g) Condición de paro del método

El método propuesto al iniciar con pocos prototipos y al agregar nuevos prototipos debe tener una condición de paro. El método propuesto se detiene cuando alcanza una aptitud igual o menor a cero, que indica que los prototipos de SMP clasifican correctamente todos los patrones de la ME. Sin embargo, alcanzar el cero puede conducir a un sobre ajuste en la ME y obtener una baja generalización en la MP. En caso contrario el método se detiene cuando alcanza un número máximo de iteraciones. Sin embargo, si ninguna de estas dos condiciones de paro se cumple el método inicia de nuevo en la inyección de patrones mal clasificados de la ME.

4.2.4 Evaluación de los prototipos con la MP

La generalización que tiene los métodos de GP se evalúa con la Muestra de Prueba (MP). Los patrones de la MP no se usaron durante la generación de los prototipos. La evaluación solo se realiza cuando se generan todas las SMP para todas las ME, y se procede a clasificar las MP con sus respectivos prototipos. La precisión alcanzada en MP se promedia para conocer la generalización para esa muestra de datos.

Método Propuesto

Como ejemplo, los resultados obtenidos de las ME "banana" se muestran en la tabla 4.3. En esta tabla se muestran los resultados obtenidos en exactitud para cada ME y MP, y el número de prototipos generados por el método propuesto. Además, se muestra la exactitud de la regla NN en la MP. Este proceso se debe repetir para las todas las muestras de datos que se usen. En todas las MP el método propuesto superó la regla NN que usa toda la ME completa para clasificar la MP.

Tabla 4.2 Resultados obtenidos de la ME "banana".

No	ME	MP	Prototipos	Re	NN-MP	Patrones
1	0.895	0.902	13	99.727	0.898	4770
2	0.892	0.871	14	99.706	0.862	4770
3	0.895	0.873	14	99.706	0.857	4770
4	0.903	0.894	13	99.727	0.846	4770
5	0.898	0.888	14	99.706	0.876	4770
6	0.901	0.873	12	99.748	0.872	4770
7	0.895	0.893	12	99.748	0.862	4770
8	0.89	0.872	13	99.727	0.849	4770
9	0.89	0.905	12	99.748	0.9	4770
10	0.902	0.896	13	99.727	0.872	4770
Promedio	0.896	0.887	13	99.727	0.869	4770

Los resultados de la tabla 4.3 se muestran en la figura 4.10 para apreciar gráficamente la diferencia de los resultados de los prototipos y la ME al clasificar los patrones de la MP.

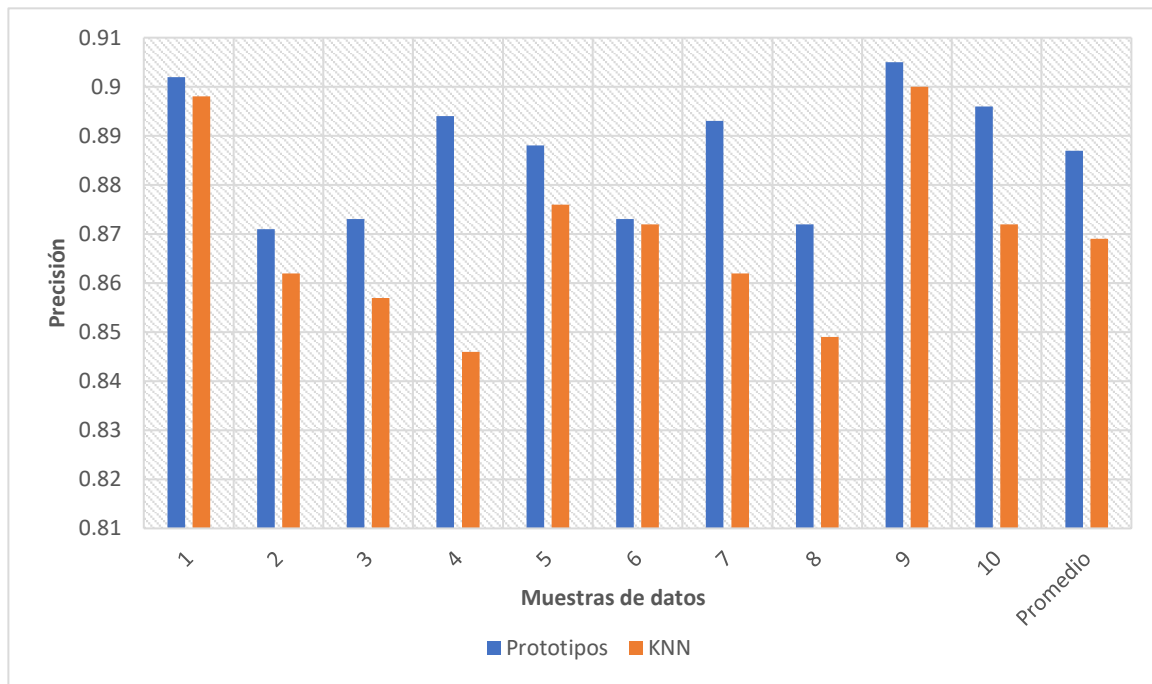


Figura 4.10 Resultados de la precisión obtenida de los prototipos y el clasificador KNN.

4.2.5 Evaluación de los resultados

Para comparar los resultados obtenidos por el método propuesto con los métodos de GP se debe evaluar la generalización en la ME y MP. Además, se debe calcular la reducción alcanzada. Para evaluar los resultados del método propuesto se deben usar diferentes muestras de datos. A cada muestra de datos se le aplicará el método propuesto para generar sus prototipos. La información que se obtiene de los prototipos se usa para comparar el método propuesto con los métodos de GP del estado del arte. La información que se usa para comparar los métodos de GP es la cantidad y calidad de clasificación de los prototipos. La calidad de los prototipos en la GP se basa en reducir el tamaño de la ME sin disminuir drásticamente la precisión de clasificación.

La reducción se calcula con la fórmula usada en el trabajo de (Kim & Oommen, 2003a);

$$Re = \left(\frac{ME_t - P_t}{ME_t} \right) * 100$$

Método Propuesto

En donde ME_t es el número total de patrones de la ME y P_t es el número total de prototipos para una determinada ME. Para todas las ME se calcula la Re para al final sumar y promediar la reducción obtenida por cada muestra de datos.

Como ejemplo, de la tabla 4.3 de resultados de la muestra de datos de banana solo se usa la marcada en color verde. Para sumar y promediar la precisión alcanzada en todas las muestras de datos en ME, MP y Re .

4.3 Resumen

En este capítulo se describió el método propuesto de GP que hace uso de la regla NN y el algoritmo de GP que usa el AG para optimizar los prototipos. Se describieron las etapas del método propuesto con un ejemplo de la muestra "banana". Por lo que se intenta demostrar que una muestra se puede adaptar al problema de clasificación con los prototipos generados. El método propuesto consta de tres etapas esenciales para obtener los prototipos. La inicialización, optimización y validación de los prototipos con la métrica y la regla NN. La validación que se realiza con la regla NN no necesita de reglas estrictas para la selección y eliminación de los prototipos. Lo que permitirá generar prototipos para diferentes clasificadores que hagan uso de la regla NN en tareas de RP.



Capítulo 5.

Experimentación

Para iniciar con el capítulo 5, se recuerda la hipótesis planteada en la presente tesis: “Si al simplificar y aplicar los procesos de inicialización, generación y validación con base en una ME, entonces es posible obtener prototipos útiles, con el fin de clasificar nuevos patrones en tareas de clasificación usando la regla NN”.

En este capítulo se presentan los diferentes experimentos realizados de acuerdo con el método propuesto del capítulo anterior. Los experimentos se realizaron para comprobar y validar la hipótesis planteada.

Experimentación

A continuación, se describen los parámetros y las muestras de datos usados para la experimentación. También, se muestran los diferentes parámetros usados en el AG, la regla NN y las métricas de evaluación usadas.

5.1 Preprocesamiento de las muestras de datos

Las muestras de datos y los métodos de GP son tomados de la herramienta Keel Suite 3.0. Las muestras de datos seleccionadas de la herramienta se muestran en la tabla 5.1 con su nombre, características, número de clases y número total de patrones.

Tabla 5.1 Muestra de datos usadas para la GP.

No.	Nombre	Características	Clases	No. P
1	Australian	14	2	690
2	Appendicitis	7	2	106
3	Balance	4	3	625
4	Banana	2	2	5,300
5	Bupa	6	2	345
6	Breast Cancer Wisconsin (Diagnostic)	32	2	569
7	Iris	4	3	150
8	Phoneme	5	2	5,404
9	Pima	8	2	768
10	Ring	20	2	7,400
11	Titanic	3	2	2,201
12	Vehicle	18	4	846
13	Wine	13	3	178
14	Wisconsin	9	2	699

Las características de los patrones de las 14 muestras de datos se normalizaron de 0 a 1. En la figura 5.1 se muestra la pantalla de inicio de esta herramienta. La herramienta *Keel Suite 3.0* cuenta con diferentes métodos de selección de prototipos y GP.

Experimentación



Figura 5.1 Herramienta KEEL Suite 3.0.

Las muestras de datos de tabla 5.1 se dividieron usando *Keel Suite 3.0* con validación cruzada con $n=10$. Para realizar la validación cruzada con *Keel Suite 3.0* se elige *Data Management*, y en la nueva vista se elige *Partition*. Finalmente, dentro de la ventana *Partition* de *Keel Suite 3.0* se elige la muestra de datos que se va a dividir, y si el proceso finalizó correctamente se mostrará el siguiente mensaje *Your partitions have been correctly generated!!!*. El preprocesamiento asegura que ninguna muestra de datos usada por el método propuesto o los métodos estado del arte tenga una partición diferente o características normalizadas con un proceso diferente.

Los métodos usados del estado del arte se presentan en la tabla 5.2 que contiene el año, referencia y nombre abreviado del método. Los parámetros usados por los métodos se tomaron del trabajo de (Triguero et al., 2012).

Tabla 5.2 Métodos del estado del arte elegidos.

No.	Año	Trabajo	Abreviatura
1	1981	(Koplowitz & Brown, 1981)	GENN
2	1990	(Kohonen, 1990)	LVQ3
3	1991	(Geva & Sitte, 1991b)	DSM
4	1993	(Xie et al., 1993)	VQ-NN
5	1997	(Odorico, 1997)	LVQTC
6	1997	(Decaestecker, 1997)	MSE
7	1997	(Hamamoto et al., 1997)	BTS3
8	2003	(Kim & Oommen, 2003a)	HYB

Experimentación

9	2004	(Fernández & Isasi, 2004)	ENPC
10	2004	(Sánchez, 2004)	RSP3
11	2004	(Yen et al., 2004)	AVQ
12	2005	(Li et al., 2005)	LVQPRU
13	2005	(Raicharoen & Lursinsap, 2005)	POC-NN
14	2006	(Lozano et al., 2006)	Mixt Gauss
15	2007	(Fayed et al., 2007)	SGP
16	2007	(Cervantes et al., 2007)	AMPSO
17	2009	(Nanni & Lumini, 2009)	PSO
18	2010	(Triguero et al., 2010)	IPADE

5.2 Generación de los prototipos

A continuación, se presentan los parámetros generales usados en el método propuesto para generar, optimizar y validar todos los prototipos de las muestras de datos de la tabla 5.1.

Parámetros usados para todas las muestras de datos:

Método propuesto:

- Iteraciones: 1050
- Prototipos iniciales: 1 prototipo aleatorio por clase
- Inyección: 1 patrón mal clasificado de la ME por clase.
- Métricas de Evaluación: MfP .

Regla NN:

- No. Vecinos: 1
- Función de similitud: distancia euclidiana

Algoritmo Genético Local, Global 1 y 2:

- No. Elite: 1
- Operador de selección: ruleta.
- Cruza: uniforme.
- Mutación: Inversión base N .
- Mutación: 8%
- No. Individuos: 200
- No. Generaciones: 10

Experimentación

- Probabilidad de cruce: 15

Los resultados que se presentan se obtuvieron de 140 experimentaciones con la métrica de *MfP*. El método propuesto tiene el nombre de AG-*MfP*.

5.3 Resultados

Las secciones que se muestran a continuación siguen un orden y presentan los resultados obtenidos por el método propuesto. Los resultados del método propuesto primero se compararon usando la regla NN con la ME. Después, los resultados del método propuesto se compararon con todos los métodos de GP de la tabla 5.2.

5.3.1 Comportamiento de la aptitud con los parámetros del método propuesto

La función de aptitud del AG minimiza el error de clasificación y la distancia de los prototipos con los patrones de la ME. En la figura 5.2 se muestran los resultados de la función de aptitud para el AG de una muestra de prototipos. Los resultados se dividen en aptitud máxima, mínima y promedio. En total del AG global realizó 1050 generaciones para cada Submuestra de Prototipos (SMP) generados.

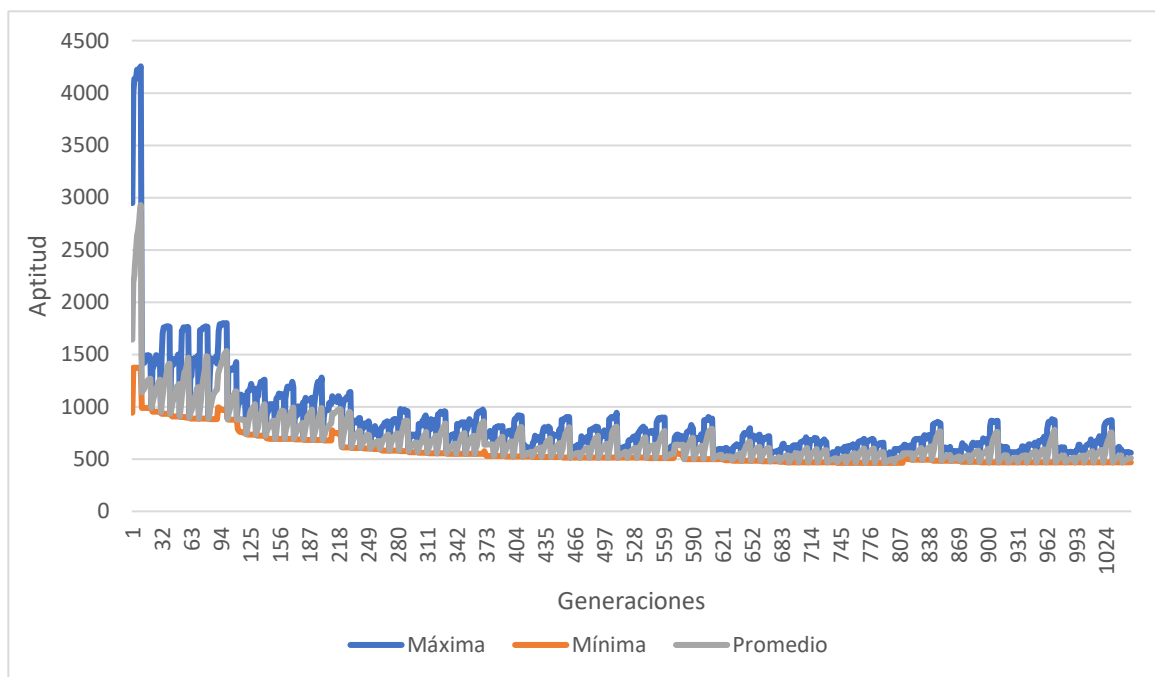


Figura 5.2 Aptitud del AG global con los parámetros del método propuestos.

5.3.2 Resultados obtenidos del método propuesto

La evaluación de todos los métodos de GP se realizó con la métrica de exactitud, como en el trabajo de Triguero (Triguero et al., 2012). Los resultados que se obtuvieron de los experimentos con el método propuesto se muestran en la tabla 5.3. En la tabla se muestra la exactitud de la ME, MP y la Reducción (Re) alcanzada por el AG-MfP. Además, en la tabla se muestran los resultados que obtiene la regla NN al evaluar la MP. La evaluación de la ME con la regla NN no se realizó debido a que clasificaría correctamente toda la ME.

El método propuesto generó como mínimo para una muestra de datos 4 prototipos, y como máximo generó 26 prototipos. El número de prototipos que puede generar el método propuesto depende del número de clases y la complejidad de la muestra de datos. Como ejemplo, para una ME con dos clases y los parámetros propuesto, el número máximo de prototipos que se pueden generar son 20 prototipos.

Tabla 5.3 Resultados obtenidos por el método propuesto y la métrica MfP.

No.	Nombre	AG-MfP				KNN	
		ME	MP	Re	Prototipos	MP	Patrones
1	Appendicitis	0.935	0.874	92.76	6.9	0.801	95
2	Australian	0.88	0.713	98.66	8.3	0.815	621
3	Balance	0.92	0.836	96.84	17.8	0.784	562
4	Banana	0.897	0.885	99.73	13	0.869	4770
5	Bupa	0.803	0.631	95.85	12.9	0.602	310
6	BCW	0.977	0.946	98.52	7.6	0.951	512
7	Iris	0.991	0.925	95.48	6.1	0.946	135
8	Phoneme	0.819	0.803	99.78	10.9	0.899	4863
9	Pima	0.84	0.683	98.08	13.3	0.654	691
10	Ring	0.898	0.868	99.8	13.2	0.736	6660
11	Titanic	0.763	0.764	99.8	4	0.607	1980
12	Vehicle	0.725	0.64	97.11	22	0.691	761
13	Wine	0.999	0.978	95.38	7.4	0.954	160
14	Wisconsin	0.978	0.961	99.22	4.9	0.951	629
	Promedio	0.883	0.822	97.64		0.804	

5.3.3 Evaluación de los resultados obtenidos

En la tabla 5.4 se muestran los resultados de obtenidos por los métodos de GP del estado del arte. Los métodos de GP se encuentran organizados alfabéticamente. Para cada método de GP se muestra el promedio de los resultados en ME, MP y Re para las 14 muestras de datos de la tabla 5.1. Todos los métodos usaron las mismas muestras de datos normalizadas y particionadas. Además, se usó la herramienta *Keel Suite 3.0* para generar los prototipos de todos los métodos.

Tabla 5.4 Resultados de los métodos de GP del estado del arte.

No.	Nombre	ME	MP	Re
1	AMPSO	0.883	0.822	97.64
2	AVQ	0.776	0.76	93.21
3	BTS3	0.754	0.745	98.63
4	DSM	0.814	0.808	5.04
5	ENPC	0.831	0.81	5.04
6	GENN	0.901	0.801	79.33
7	HYB	0.857	0.831	14.26
8	IPADE	0.877	0.785	45.37
9	LVQ3	0.85	0.827	99.15
10	LVQPRU	0.789	0.786	95.13
11	LVQTC	0.796	0.779	95.26
12	MixGauss	0.819	0.804	97.28
13	MSE	0.807	0.799	95.32
14	POC	0.832	0.805	94.24
15	PSO	0.758	0.735	71.01
16	RSP3	0.874	0.838	95.13
17	SGP	0.87	0.826	80.04
18	VQ	0.689	0.682	97.01

a) Evaluación de la exactitud en la ME

En la tabla 5.5 se muestran los resultados ordenados de mayor a menor en la ME. El primer lugar lo ocupa la regla NN. Mientras, el segundo lugar es para ENPC, el tercer lugar lo ocupa el método HYB, mientras, el método propuesto ocupa el 4 lugar. Sin embargo, ocupar los primeros lugares en la ME indica que los métodos pueden tener un sobreajuste que afecte la

Experimentación

clasificación de nuevos patrones. La mayoría de los métodos del estado del arte que hacen uso de modelos evolutivos se encuentran dentro de los 9 primeros lugares. Los métodos son ENPC, PSO, IPADE y MSE.

Tabla 5.5 Resultados de exactitud para las ME ordenados de mayor a menor.

No.	Nombre	ME	MP	R
1	KNN	0.985	0.804	0
2	ENPC	0.901	0.801	79.33
3	HYB	0.888	0.798	45.37
4	AG-MfP	0.883	0.822	97.64
5	PSO	0.881	0.842	95.99
6	RSP3	0.87	0.826	80.04
7	GENN	0.857	0.831	14.26
8	IPADE	0.85	0.827	99.15
9	MSE	0.832	0.805	94.24
10	LVQTC	0.822	0.807	97.27
11	BTS3	0.814	0.808	5.04
12	MixGauss	0.807	0.799	95.32
13	LVQPRU	0.795	0.778	96.11
14	LVQ3	0.786	0.783	95.99
15	DSM	0.784	0.781	95.99
16	AMPSO	0.776	0.76	94.18
17	VQ	0.769	0.77	89.53
18	POC	0.758	0.735	71.01
19	AVQ	0.754	0.745	98.63
20	SGP	0.689	0.682	97.01

b) Evaluación de la exactitud en la MP

La evaluación de la generalización para todos los métodos de GP se evalúa con la MP, en la tabla 5.6 se muestran los resultados ordenados de mayor a menor en la MP. El método de GP que obtiene el mejor resultado en generalización es PSO. El método propuesto AG-MfP obtiene el 5 lugar. Mientras, el método ENPC que ocupó el segundo lugar en la evaluación de la ME fue desplazado al lugar 10 en los resultados de la MP. Lo que indica que ENPC tiene un sobreajuste y afecta negativamente para clasificar nuevos patrones. En cambio, el método propuesto AG-MfP se mantiene cercano a RSP3 y IPADE.

Tabla 5.6 Resultados de exactitud para las MP ordenados de mayor a menor.

No.	Nombre	ME	MP	Reducción
1	PSO	0.881	0.842	95.99
2	GENN	0.857	0.831	14.26
3	IPADE	0.85	0.827	99.15
4	RSP3	0.87	0.826	80.04
5	AG-MfP	0.883	0.822	97.64
6	BTS3	0.814	0.808	5.04
7	LVQTC	0.822	0.807	97.27
8	MSE	0.832	0.805	94.24
9	KNN	0.985	0.804	0
10	ENPC	0.901	0.801	79.33
11	MixGauss	0.807	0.799	95.32
12	HYB	0.888	0.798	45.37
13	LVQ3	0.786	0.783	95.99
14	DSM	0.784	0.781	95.99
15	LVQPRU	0.795	0.778	96.11
16	VQ	0.769	0.77	89.53
17	AMPSO	0.776	0.76	94.18
18	AVQ	0.754	0.745	98.63
19	POC	0.758	0.735	71.01
20	SGP	0.689	0.682	97.01

c) Reducción obtenida

La última evaluación que se realiza a todos los métodos de GP es el porcentaje de reducción obtenido en la muestra de prototipos generados. En la tabla 5.7 se ordenan los resultados de la reducción de mayor a menor. La mejor reducción la obtuvo el método IPADE con un promedio de 99.15 % y el método propuesto AG-MfP se encuentra en el 3 lugar. Sin embargo, los métodos que superaron al método propuesto en MP como PSO ocupa el 7 lugar, GENN ocupa el lugar 18 y RSP3 ocupa el lugar 14.

Tabla 5.7 Porcentaje de reducción de la ME ordenados de mayor a menor.

No.	Nombre	ME	MP	Reducción
1	IPADE	0.85	0.827	99.15
2	AVQ	0.754	0.745	98.63
3	AG-MfP	0.883	0.822	97.64
4	LVQTC	0.822	0.807	97.27
5	SGP	0.689	0.682	97.01
6	LVQPRU	0.795	0.778	96.11
7	PSO	0.881	0.842	95.99
8	LVQ3	0.786	0.783	95.99
9	DSM	0.784	0.781	95.99
10	MixGauss	0.807	0.799	95.32
11	MSE	0.832	0.805	94.24
12	AMPSO	0.776	0.76	94.18
13	VQ	0.769	0.77	89.53
14	RSP3	0.87	0.826	80.04
15	ENPC	0.901	0.801	79.33
16	POC	0.758	0.735	71.01
17	HYB	0.888	0.798	45.37
18	GENN	0.857	0.831	14.26
19	BTS3	0.814	0.808	5.04
20	KNN	0.985	0.804	0

5.3.4 Evaluación de los resultados obtenidos

Los resultados que se obtuvieron demuestran que el método propuesto obtiene resultados aceptables en la reducción y la clasificación en la ME. Sin embargo, en la clasificación de la MP ocupa el 5 lugar total de 18 métodos de GP y la regla NN. Para hacer una evaluación que considere los resultados obtenidos en la MP, ME y la RE de cada método de GP se usa el ranking global usado en (Vázquez et al., 2018). La fórmula para calcular el ranking global se muestra a continuación.

$$rank(\text{metodo}) = \sum_{s=1}^m \frac{m - s + 1}{m} r_s$$

Donde r_s indica el número de veces que el método aparece en un rango s , y m indica el total de métodos evaluados en el ranking global. Los resultados que se obtuvieron se muestran en la tabla 5.8 ordenados de mayor a menor. El ranking global se calculó usando el promedio de

Experimentación

MP, ME y reducción. En la tabla 5.8 se muestran los resultados del ranking global. El método propuesto AG-MfP y IPADE obtienen el mejor resultado de la evaluación, seguido de PSO en tercer lugar.

Tabla 5.8 Ranking de los métodos de GP.

No.	Nombre	Ranking
1	AG-MfP	2.55
2	IPADE	2.55
3	PSO	2.4
4	LVQTC	2.1
5	RSP3	1.95
6	ENPC	1.8
7	GENN	1.8
8	MSE	1.75
9	KNN	1.65
10	HYB	1.55
11	MixGauss	1.5
12	LVQPRU	1.45
13	LVQ3	1.45
14	BTS3	1.35
15	DSM	1.3
16	AVQ	1.2
17	AMPSO	0.9
18	SGP	0.9
19	VQ	0.85
20	POC	0.5

Con esto se demuestra que el método propuesto que usó el AG con la métrica de MfP tiene la capacidad de generar nuevos prototipos para clasificar nuevos patrones.

5.3.5 Resultados de ME con menos de 500 patrones

Uno de los problemas que se detectaron en el método propuesto fue el sobre ajuste en ME con menos de 500 patrones. Los prototipos generados para estas de ME obtienen una clasificación casi perfecta para todos los patrones. Sin embargo, al clasificar los patrones de la MP se obtiene una peor clasificación. Para mostrar el sobreajuste que presentó el método propuesto se hace uso de la muestra de datos de apendicitis y el método de GP IPADE. La

Experimentación

muestra de datos de "apendicitis" contiene 106 patrones y dos clases. En la tabla 5.9 se muestran los resultados obtenidos en la ME, MP y los prototipos generados en cada método.

Tabla 5.9 Resultados obtenidos en la muestra de datos de apendicitis.

No.	AG-MfP			IPADE		
	ME	MP	P.	ME	MP	P.
1	0.985	0.933	5	0.993	1	3
2	0.993	0.861	5	0.993	1	4
3	0.978	1	4	0.985	1	3
4	0.993	1	8	0.993	0.933	4
5	0.985	1	4	0.985	1	3
6	0.993	0.867	9	0.985	1	3
7	0.985	0.933	6	0.985	0.933	3
8	0.985	1	7	0.993	0.867	3
9	0.993	0.933	8	1	0.933	3
10	0.993	0.861	5	0.985	0.933	3
Promedio	0.988	0.939	6.1	0.99	0.96	3.2

La métrica usada para evaluar la clasificación es exactitud. Los resultados obtenidos por el método propuesto AG-MfP y IPADE en la ME son altos. Sin embargo, los resultados que se obtuvieron en la MP para IPADE se obtiene mejor exactitud. El método propuesto tiende a generar un número mayor de prototipos y sobreajustarse en ME pequeñas. En cambio, el método IPADE demuestra que con un menor número de prototipos el sobreajuste es menor y puede obtener mejores resultados en la MP.

5.3.6 Resultados de ME con más de 2,000 patrones

En muestras de datos con más de 2,000 patrones los resultados que se obtienen son mejores en la generalización. En la tabla 5.10 se muestran los resultados obtenidos de la muestra de datos de banana. La métrica usada para evaluar los resultados de la muestra de datos es exactitud. La muestra de datos es "banana" con que contiene 5,300 patrones con dos características y dos clases. La muestra de datos "banana" tiene límites de clase claros, no obstante existe una gran superposición entre ambas clases (Triguero et al., 2012). El método propuesto AG-MfP obtiene la misma generalización que PSO. El método PSO obtiene la mejor generalización en esta muestra de datos de "banana". Sin embargo, el número de prototipos obtenido por el método propuesto AG-MfP es considerablemente menor.

Tabla 5.10 Resultados obtenidos en la muestra de datos de banana.

No.	AG-MfP			PSO		
	ME	MP	Proto.	ME	MP	Proto.
1	0.895	0.902	13	0,892	0,866	95
2	0.892	0.871	14	0,908	0,887	95
3	0.895	0.873	14	0,901	0,891	95
4	0.903	0.894	13	0,9	0,887	95
5	0.898	0.888	14	0,894	0,872	95
6	0.901	0.873	12	0,893	0,891	95
7	0.895	0.893	12	0,896	0,896	95
8	0.89	0.872	13	0,906	0,901	95
9	0.89	0.905	12	0,896	0,881	95
10	0.902	0.896	13	0,9	0,903	95
Promedio	0.896	0.887	13	0,899	0,887	95

Con los resultados obtenidos se demuestra que el método propuesto es capaz de obtener una de las mejores reducciones y una generalización aceptable de las submuestras de prototipos generadas. Lo que demuestra que al aplicar el AG para solo optimizar una parte de los prototipos como se hizo en los pasos del método propuesto mejora los resultados del AG. Los pasos del método propuesto trataron de simplificar el problema de generación de prototipos desde la inicialización, generación y validación. El solo iniciar con la cantidad mínima de prototipos cada iteración de inicialización, generación y validación reduce la complejidad de las muestras de datos. No obstante, al combinar cada parte mínima de los prototipos optimizados se trata de cubrir el problema completo de generación de prototipos. Sin embargo, los parámetros usados por el método propuesto para muestras de datos grandes limitan la generalización que puede alcanzar.

5.4 Resumen

En este capítulo se presentaron los diferentes experimentos del método propuesto del capítulo 4 de esta investigación. El método propuesto se aplicó en diferentes tareas de clasificación y generación de prototipos. La generación de prototipos se aplicó en 14 muestras de datos de diferentes. Todas las muestras de datos tuvieron un preprocesamiento para normalizar los valores de sus características. Además, se aplicó la validación cruzada con $n=10$ para generar 10 ME y MP. Las ME se usaron para generar los prototipos y con las MP se evaluó la generalización del método propuesto.

Experimentación

Para comparar los resultados del método propuesto se usaron 18 métodos de GP y el clasificador de la regla NN. Todos los métodos de GP usaron las mismas muestras de datos. La evaluación del método propuesto y los métodos de GP se realizó con la métrica de exactitud. La exactitud evaluó la clasificación de los prototipos con las ME y MP. También, se evaluó la reducción de los prototipos y la ME de cada método. Con los resultados obtenidos en las ME, MP y la reducción se hizo un ranking global para posicionar los métodos de GP. Lo que demostró la capacidad del método propuesto para generar prototipos representativos usando la ME para clasificar nuevos patrones.



Capítulo 6.

Conclusiones

Los métodos de GP buscan generar una submuestra de prototipos que tenga la capacidad de clasificar la MP con una precisión similar o mejor a la ME. La submuestra de prototipos debe tener un tamaño menor a la ME. Los métodos de GP solo usan la ME para la generar la submuestra de prototipos, y solo se mejorarán los resultados para la regla NN. Algunos de los métodos de GP usan diferentes métodos de selección de prototipos que les ayudan a obtener mejores resultados. Sin embargo, la mayoría de los métodos de GP no usan métricas diferentes en la validación de los prototipos. Lo que limita la adaptación de los prototipos a un problema de clasificación específico. Además, se encontró que todos los métodos de GP comparten

Conclusiones

tres procesos generales para poder generar los prototipos. Los tres procesos generales son la inicialización, generación y validación, y al tomar como base estos tres procesos para tratar de simplificarlos se propuso un método de GP que permite construir y buscar los mejores prototipos posibles.

Para generar los primeros prototipos se usó el algoritmo genético tradicional y un clasificador KNN, que requería el número de prototipos a generar y no obtenía buenos resultados en las diferentes muestras de datos. Para mejorar los resultados se investigó, diseñó y se propuso un algoritmo basado en los tres pasos generales que son la inicialización, generación y validación de la submuestra de prototipos. Estos tres pasos generales GP se definen con nombres diferentes en todos los métodos de GP. No obstante, al compararlos todos aplican una inicialización, generación y validación. Además, se encontró que todos los métodos de GP de la herramienta *Keels Suite 3.0* realizan una normalización de datos para reducir el espacio de características.

La investigación que se realizó definió los parámetros y pasos generales que debía seguir el método propuesto. No obstante, para mejorar los resultados se dividió la generación de los prototipos en diferentes etapas que simplificó la generación de prototipos. Posteriormente se definió el número máximo de generaciones, mutación, operadores de selección, tamaño de la población del AG y la métrica de evaluación que se aplicó a todas las muestras de datos.

6.1 Aportaciones

- Se demostró que usar un AG para obtener prototipos representativos para clasificar nuevos patrones con una precisión similar a la ME completa.
- Se demostró que los prototipos obtenidos tienen una buena generalización en las diferentes MP.
- Se demostró que el método propuesto obtuvo una de las mejores reducciones en comparación con las ME.
- Se demostró que el método propuesto genera prototipos representativos para ME con diferente número de patrones, número de características y número de clases, sin importar el desbalance entre las clases.

Conclusiones

- Se describieron y definieron los tres elementos que comparte en forma general los métodos de GP que son inicialización, generación y validación.

6.2 Trabajo Futuro

El trabajo futuro para el método propuesto de GP es el uso de operadores diferentes de selección, cruza y mutación. La mejora o adaptación de estos operadores para buscar dentro del espacio de características global y local supondrá una mejora de los prototipos generados. El uso de métricas de evaluación y clasificadores NN diferentes que no se abarcaron en la investigación. El utilizar diferentes inicializaciones en el método propuesto, como un solo prototipo representativo por clase, centroides o una submuestra más grande de prototipos. El uso de otro método de optimización como, enjambre de partículas (PSO) o evolución diferencial (DE). El hacer uso de una clase definida y comparar los resultados con los prototipos sin una clase definida. La mayoría del trabajo futuro propuesto se espera que mejore la clasificación y tiempo de ejecución del método propuesto.

Referencias Bibliográficas

- Adeli, H., & Sarma, K. C. (2006). Cost Optimization of Structures: Fuzzy Logic, Genetic Algorithms, and Parallel Computing. In *Cost Optimization of Structures: Fuzzy Logic, Genetic Algorithms, and Parallel Computing*. <https://doi.org/10.1002/0470867353>
- Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1), 37–66. <https://doi.org/10.1007/BF00153759>
- Ahn, C. W. (2006). Practical genetic algorithms. In *Studies in Computational Intelligence* (Vol. 18). https://doi.org/10.1007/11543138_2
- Alpaydin, E. (2010). Introduction to Machine Learning Second Edition. In *Introduction to Machine Learning*. https://doi.org/10.1007/978-1-62703-748-8_7
- Berrar, D. (2018). *Cross-validation*. 1.
- Bezdek, J. C., & Kuncheva, L. I. (2001). Nearest prototype classifier designs: An experimental study. *International Journal of Intelligent Systems*, 16(12), 1445–1473. <https://doi.org/10.1002/int.1068>
- Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.679.1104&rep=rep1&type=pdf>
- Bishop, C. M. (2006). Pattern Recognition and Machine Learning. In *Oxidation Communications* (Vol. 27, Issue 1).
- Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. *Proceedings - International Conference on Pattern Recognition*, 3121–3124. <https://doi.org/10.1109/ICPR.2010.764>
- Cervantes, A., Galván, I., & Isasi, P. (2007). An adaptive michigan approach PSO for nearest prototype classification. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4528 LNCS(PART 2), 287–296. https://doi.org/10.1007/978-3-540-73055-2_31
- Cervantes, A., Galván, I. M., & Isasi, P. (2009). AMPSO: A new particle swarm method for nearest neighborhood classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(5), 1082–1091. <https://doi.org/10.1109/TSMCB.2008.2011816>
- Chang, C. L. (1974). Finding Prototypes for Nearest Neighbor Classifiers. *IEEE Transactions on Computers*, C-23(11), 1179–1184. <https://doi.org/10.1109/T-C.1974.223827>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1), 321–357. <https://doi.org/10.1613/jair.953>
- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), 1–13. <https://doi.org/10.1186/s12864-019-6413-7>
- Conway, D., & White, J. M. (2012). *Machine Learning for Hacker*. <https://doi.org/10.1017/CBO9781107415324.004>
- Decaestecker, C. (1997). Finding prototypes for nearest neighbour classification by means of gradient descent and deterministic annealing. *Pattern Recognition*, 30(2), 281–288. [https://doi.org/10.1016/S0031-3203\(96\)00072-6](https://doi.org/10.1016/S0031-3203(96)00072-6)
- Dougherty, G. (2013). Pattern Recognition and Classification. In *Journal of Chemical Information and Modeling*. <https://doi.org/10.1007/978-1-4614-5323-9>
- Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification* (p. 738). <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471056693.html>

- Eiben, A. E., & Smith, J. E. (2015). *Introduction to Evolutionary Computing*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-44874-8>
- Ertekin, S., Huang, J., Bottou, L., & Giles, L. (2007). *Learning on the Border: Active Learning in Imbalanced Data Classification*. 127. <https://doi.org/10.1145/1321440.1321461>
- Fayed, H. A., Hashem, S. R., & Atiya, A. F. (2007). Self-generating prototypes for pattern classification. *Pattern Recognition*, 40(5), 1498–1509. <https://doi.org/10.1016/j.patcog.2006.10.018>
- Fernández, F., & Isasi, P. (2004). Evolutionary design of nearest prototype classifiers. *Journal of Heuristics*, 10(4), 431–454. <https://doi.org/10.1023/B:HEUR.0000034715.70386.5b>
- Fernández Hernández, Y. B., Bello, R., Filiberto, Y., Frías, M., Blanco, L. C., & Caballero, Y. (2015). An approach for prototype generation based on similarity relations for problems of classification. *Computacion y Sistemas*, 19(1), 109–118. <https://doi.org/10.13053/CyS-19-1-2053>
- Francone, W. B. P. N. R. E. K. F. D. (1998). Genetic Programming. In *Handbook of Metaheuristics* (Vol. 146). https://doi.org/doi:10.1007/978-1-4419-1665-5_7
- Garain, U. (2008). Prototype reduction using an artificial immune model. *Pattern Analysis and Applications*, 11(3–4), 353–363. <https://doi.org/10.1007/s10044-008-0106-1>
- García, S., Derrac, J., Cano, J. R., & Herrera, F. (2012). Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3), 417–435. <https://doi.org/10.1109/TPAMI.2011.142>
- Gasca, E., & Barandela, R. (2000). Influencia del Preprocesamiento de la Muestra de Entrenamiento en el Poder de Generalización del Perceptron Multicapa. *Proceedings of the 6th Brazilian Symposium on Neural Networks*.
- Gen, M., Cheng, R., & Lin, L. (2008). *Network Models and Optimization*. Springer London. <https://doi.org/10.1007/978-1-84800-181-7>
- Geva, S., & Sitte, J. (1991a). Adaptive Nearest Neighbor Pattern Classification. *IEEE Transactions on Neural Networks*, 2(2), 318–322. <https://doi.org/10.1109/72.80344>
- Geva, S., & Sitte, J. (1991b). Adaptive Nearest Neighbor Pattern Classification. *IEEE Transactions on Neural Networks*, 2(2), 318–322. <https://doi.org/10.1109/72.80344>
- Hackeling, G. (2014). *Mastering Machine Learning with scikit-learn*. <http://books.google.com/books?id=fZQeBQAAQBAJ&pgis=1>
- Hamamoto, Y., Uchimura, S., & Tomita, S. (1997). A Bootstrap Technique for Nearest Neighbor Classifier Design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1), 73–79. <https://doi.org/10.1109/34.566814>
- Holland, J. H. (1992). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. In *Computing and Processing*.
- Huang, H., Xu, H., Wang, X., & Silamu, W. (2015). Maximum F1-score discriminative training criterion for automatic mispronunciation detection. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 23(4), 787–797. <https://doi.org/10.1109/TASLP.2015.2409733>
- Hummel, J. E. (2013). Oxford Handbooks Online. *Choice Reviews Online*, 50(11), 50-5912-50-5912. <https://doi.org/10.5860/choice.50-5912>
- Kasabov, N. K. (2013). Foundations of neural networks, fuzzy systems, and knowledge engineering. In *Choice Reviews Online* (Vol. 35, Issue 01). <https://doi.org/10.5860/choice.35-0330>
- Kendal, S., & Creen, M. (2007). An Introduction to Knowledge Engineering. In *An Introduction to Knowledge Engineering*. <https://doi.org/10.1007/978-1-84628-667-4>
- Khoshgoftaar, T. M., Zhong, S., & Joshi, V. (2005). Enhancing software quality estimation using ensemble-classifier based noise filtering. *Intelligent Data Analysis*, 9(1), 3–27.

- <https://doi.org/10.3233/ida-2005-9102>
- Kim, S. W., & Oommen, B. J. (2003a). A brief taxonomy and ranking of creative prototype reduction schemes. *Pattern Analysis and Applications*, 6(3), 232–244. <https://doi.org/10.1007/s10044-003-0191-0>
- Kim, S. W., & Oommen, B. J. (2003b). Enhancing prototype reduction schemes with LVQ3-type algorithms. *Pattern Recognition*, 36(5), 1083–1093. [https://doi.org/10.1016/S0031-3203\(02\)00115-2](https://doi.org/10.1016/S0031-3203(02)00115-2)
- Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence, March 2001*.
- Kohonen, T. (1990). The Self-Organizing Map. *Proceedings of the IEEE*, 78(9), 1464–1480. <https://doi.org/10.1109/5.58325>
- Koplowitz, J., & Brown, T. A. (1981). On the relation of performance to editing in nearest neighbor rules. *Pattern Recognition*, 13(3), 251–255. [https://doi.org/10.1016/0031-3203\(81\)90102-3](https://doi.org/10.1016/0031-3203(81)90102-3)
- Kramer, O. (2017). *Genetic Algorithm Essentials* (Vol. 679). <https://doi.org/10.1007/978-3-319-52156-5>
- Lam, W., Keung, C. K., & Liu, D. (2002). Discovering useful concept prototypes for classification based on filtering and abstraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8), 1075–1090. <https://doi.org/10.1109/TPAMI.2002.1023804>
- Lantz, B. (2015). Machine Learning with R - Second Edition. In *Packt Publishing*. http://books.google.com/books?id=ZQu8AQAAQBAJ&printsec=frontcover&dq=intitle:Machine+Learning+with+R&hl=&cd=1&source=gbs_api%5Cnpapers2://publication/uuid/46164A51-A282-4F67-8397-9FA79F39B5B7%5Cnhttp://ebooks.cambridge.org/ref/id/CBO9781107415324A009%5Cnh
- Li, J., Manry, M. T., Yu, C., & Wilson, D. R. (2005). Prototype classifier design with pruning. *International Journal on Artificial Intelligence Tools*, 14(1–2), 261–280. <https://doi.org/10.1142/S0218213005002090>
- Logothetis, N. K. (1996). Visual Object Recognition. *Annual Review of Neuroscience*, 19(1), 577–621. <https://doi.org/10.1146/annurev.neuro.19.1.577>
- Lozano, M., Sotoca, J. M., Sánchez, J. S., Pla, F., Pekalska, E., & Duin, R. P. W. (2006). Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. *Pattern Recognition*, 39(10), 1827–1838. <https://doi.org/10.1016/j.patcog.2006.04.005>
- Marques De Sa, J. P. (2001). *Pattern recognition concepts methods and application*. <https://doi.org/10.1007/978-3-642-56651-6>
- Mehryar Mohri, Afshin Rostamizadeh, and A. T. (2012). Foundations of Machine Learning. In *Czasopismo stomatologiczne* (Vol. 20, Issue 4). https://doi.org/10.1007/978-3-642-34106-9_15
- Michalewicz, Z. (1996). Genetic Algorithms + Data Structures = Evolution Programs. In *Angewandte Chemie International Edition* (Vol. 6, Issue 11). Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-03315-9>
- Mitchell, M. (1998). *An Introduction to Genetic Algorithms* (Vol. 2).
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. https://doi.org/10.1007/978-94-011-3532-0_2
- Nanni, L., & Lumini, A. (2009). Particle swarm optimization for prototype reduction. *Neurocomputing*, 72(4–6), 1092–1097. <https://doi.org/10.1016/j.neucom.2008.03.008>
- Odorico, R. (1997). Learning vector quantization with training count (LVQTC). *Neural Networks*, 10(6), 1083–1088. [https://doi.org/10.1016/S0893-6080\(97\)00012-9](https://doi.org/10.1016/S0893-6080(97)00012-9)
- Raicharen, T., & Lursinsap, C. (2005). A divide-and-conquer approach to the pairwise opposite class-nearest neighbor (POC-NN) algorithm. *Pattern Recognition Letters*, 26(10), 1554–1567.

- <https://doi.org/10.1016/j.patrec.2005.01.003>
- Ribeiro Filho, J., Treleavan, P., & Alippi, C. (1994). *Genetic Algorithm Programming Environments*.
- Samanta, S. (2014). Genetic Algorithm: An Approach for Optimization (Using MATLAB). *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, 3(3), 261–267. <https://www.ijltet.org/wp-content/uploads/2014/01/431.pdf>
- Sánchez, J. S. (2004). High training set size reduction by space partitioning and prototype abstraction. *Pattern Recognition*, 37(7), 1561–1564. <https://doi.org/10.1016/j.patcog.2003.12.012>
- Sánchez, J. S., Barandela, R., Marqués, A. I., Alejo, R., & Badenas, J. (2003). Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters*, 24(7), 1015–1022. [https://doi.org/10.1016/S0167-8655\(02\)00225-8](https://doi.org/10.1016/S0167-8655(02)00225-8)
- Scikit learn, D. (2021). *Sklearn Metrics*. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
- Singh, A., Yadav, A., & Rana, A. (2013). K-means with Three different Distance Metrics. *International Journal of Computer Applications*, 67(10), 13–17. <https://doi.org/10.5120/11430-6785>
- Sotoca, J. M., Pla, F., Sec, C. R., & Lozano, M. (2004). An Adaptive Condensing Algorithm Based on Mixtures of Gaussians. In *Setè Congrés Català d'Intel. Ligencia Artificial, Barcelona (Spain), Frontiers in Artificial Intelligence and Applications*, 113, 225–232.
- Suthaharan, S. (2016). Machine Learning Models and Algorithms for Big Data Classification. In *Integrated Series in Information Systems* (Vol. 36). <https://doi.org/10.1007/978-1-4899-7641-3>
- The Royal Society. (2017). Machine learning: the power and promise of computers that learn by example. In *Report by the Royal Society* (Vol. 66, Issue January). <https://doi.org/10.1126/scitranslmed.3002564>
- Theodoridis, S., & Koutroumbas, K. (2003). *Pattern Recognition*.
- Triguero, I., Derrac, J., García, S., & Herrera, F. (2012). A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 42(1), 86–100. <https://doi.org/10.1109/TSMCC.2010.2103939>
- Triguero, I., García, S., & Herrera, F. (2010). IPADE: Iterative prototype adjustment for nearest neighbor classification. *IEEE Transactions on Neural Networks*, 21(12), 1984–1990. <https://doi.org/10.1109/TNN.2010.2087415>
- Triguero, I., & Herrera, F. (2012). *Prototype Generation for Nearest Neighbor Classification: Survey of Methods*. 1–19.
- Vázquez, E., Arnulfo García-Hernández, R., & Ledeneva, Y. (2018). Sentence features relevance for extractive text summarization using genetic algorithms. *Journal of Intelligent & Fuzzy Systems*, 35(1), 353–365. <https://doi.org/10.3233/JIFS-169594>
- Whitley, D. (1994). A genetic algorithm tutorial. In *Statistics and computing* (Vol. 4, Issue 2).
- Xie, Q., Laszlo, C. A., & Ward, R. K. (1993). Vector Quantization Technique for Nonparametric Classifier Design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12), 1326–1330. <https://doi.org/10.1109/34.250849>
- Yen, C. W., Young, C. N., & Nagurka, M. L. (2004). A vector quantization method for nearest neighbor classifier design. *Pattern Recognition Letters*, 25(6), 725–731. <https://doi.org/10.1016/j.patrec.2004.01.012>