

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN
Laboratorio de Inteligencia Artificial

**Memorias asociativas geométricas
para la clasificación de patrones**

T E S I S

que para obtener el grado de

Doctor en Ciencias de la Computación

Presenta:

Benjamín Cruz Torres

Directores de tesis:

Dr. Ricardo Barrón Fernández

Dr. Juan Humberto Sossa Azuela



México, D.F. Noviembre del 2009

Dedicado a:

*A mi mamá y a mi papá por darme
la vida y por darme los principios
con los que se rige mi vida.*

*A mi abuelita por cuidarme durante
el tiempo que pasé en mi doctorado.*

*Don Nacho Zafra y Eduardo Gil Roble,
que Dios los tenga en su gloria.*

Agradecimientos

A Dios.

Realmente si quisiera agregar a todas las personas que me ayudaron directa o indirectamente en la realización de este trabajo, tendría que usar más hojas de las que consta este escrito, así que intentaré abreviar.

Agradecimiento especial a mi gran amiga Jeanine, por su compañía, cariño, afecto y por sus consejos, pero principalmente por darme su amistad. A Roberto Razo y Laura Gil por su amistad y cariño. A mis amigas Lina, Ludim, Carmen, Gaby y a mi amigos Gabriel, Uriel, Fernando, Christian, por brindarme su amistad y consejos cuando más lo necesitaba.

A mis amigos del CIC: Bryan, Víctor (*el Mayor Oso*), Miri, Sam, Gaby, Erandi, Fernando, Jesús, Cynthia, Gil, Juan Villegas, Juan Evencio, Félix, Jymmi, Cecy, Hermes, Roberto, Bety, Miguel; por hacerme más placentera mi estancia en el CIC. A todos mis compañeros del laboratorio de IA y de RP.

A mis amigos de *The Eva Village* y del *Movimiento Iglesia Joven*, por complementar mi vida estudiantil con diversión, ratos de ocio y aprendizaje espiritual. A mis hermanos, a mis tíos, a mis primos y a toda mi familia por su apoyo y consejos.

A mis directores de tesis, Dr. Ricardo Barrón Fernández y al Dr. Juan Humberto Sossa Azuela y a mis sinodales por sus enseñanzas y comentarios para mejorar este trabajo. También al Dr. Salvador Godoy, a la Dr. Nareli Cruz, al Dr. Hiram Calvo, al Dr. Marco Moreno, a la Dra. Elsa Rubio, al Dr. Edgardo Manuel y al Dr. Domingo.

Al personal del Centro de Investigación en Computación y al Instituto Politécnico Nacional por el espacio y los recursos necesarios para completar mis estudios. A CONACyT, COMECyT y al CIC por el apoyo económico prestado para terminar mis estudios. Al personal de la UTE, al personal de la biblioteca y la USTO; ya que sin ellos no podría haber terminado mis estudios.

Este documento se ha realizado con la ayuda financiera de la Comunidad Europea, la Unión Europea, la Comisión Europea y el CONACYT bajo el proyecto FONCICYT 93829. El contenido de este documento es responsabilidad exclusiva del Instituto Politécnico Nacional y en modo alguno debe considerarse re refleja la posición de la Unión Europea.

En la literatura, existen muchos modelos para el reconocimiento de patrones cuya finalidad puede enmarcarse en la clasificación o la restauración de patrones. Las memorias asociativas son, precisamente, uno de estos modelos. Desde el siglo pasado han emergido muchos e interesantes modelos de memorias asociativas que resuelven estos problemas.

En el presente trabajo se describe un nuevo modelo de memoria asociativa que utiliza operaciones y operadores de la llamada *álgebra geométrica conforme* para su funcionamiento, las llamadas memorias asociativas geométricas. El álgebra geométrica conforme proporciona, a bajo nivel, un novedoso marco de trabajo, libre de coordenadas, para resolver problemas numéricos.

Se presentan tres modos de operación para las memorias asociativas geométricas; los cuales permiten resolver, precisamente, tres problemas en el área de reconocimiento de patrones: la clasificación supervisada, clasificación no supervisada y restauración de patrones.

Además, se incluye una amplia investigación sobre el estado del arte, junto con una introducción sobre los conceptos y definiciones más importantes de reconocimiento y clasificación de patrones, memorias asociativas y álgebra geométrica (en particular el álgebra geométrica conforme).

Dentro del trabajo, se incluyen condiciones formales de recuperación, clasificación y restauración perfecta y robusta, bajo las cuales el modelo puede funcionar. Además, se dan varios ejemplos con datos numéricos y con patrones reales, para mostrar la funcionalidad de las nuevas memorias. Las nuevas propuestas son, también, comparadas con otras propuestas reportadas en la literatura existente. Al final, se describen las conclusiones del trabajo junto con algunas ideas para trabajos futuros.

ÍNDICE

1. INTRODUCCIÓN	1
1.1 OBJETIVO GENERAL	3
1.1.1 OBJETIVOS ESPECÍFICOS	3
1.2 JUSTIFICACIÓN.....	4
1.3 APORTACIONES	4
1.4 ESTRUCTURA DE LA TESIS	5
2. ESTADO DEL ARTE	7
2.1 REDES NEURONALES CON ÁLGEBRA GEOMÉTRICA.....	7
2.2 MÁQUINAS DE SOPORTE VECTORIAL EN ÁLGEBRA GEOMÉTRICA.....	9
2.3 LA NEURONA HÍPER-ESFÉRICA.....	9
2.4 AGRUPANDO FORMAS IRREGULARES CON NEURONAS DE ALTO ORDEN.....	10
2.5 SUPERFICIES DE DECISIÓN ESFÉRICAS. USANDO EL MODELO CONFORME.....	11
2.6 CÁLCULO NEURONAL CON ALGEBRA GEOMÉTRICA	12
3. MARCO TEÓRICO	15
3.1 RECONOCIMIENTO Y CLASIFICACIÓN DE PATRONES	15
3.1.1 EL CONCEPTO DE PATRÓN.....	15
3.1.2 CLASIFICACIÓN DE PATRONES	16
3.1.3 SEPARABILIDAD LINEAL	19
3.1.4 REGIONES DE SOPORTE	20
3.1.5 RESTAURACIÓN DE PATRONES	21
3.1.6 CLASIFICACIÓN NO SUPERVISADA.....	24
3.2 MEMORIAS ASOCIATIVAS	26

3.2.1 REDES NEURONALES	26
3.2.2 CONCEPTOS SOBRE MEMORIAS ASOCIATIVAS.....	27
3.2.3 MODELOS DE MEMORIAS ASOCIATIVAS	30
3.3 ÁLGEBRA GEOMÉTRICA	32
3.3.1 PROPIEDADES DEL ÁLGEBRA GEOMÉTRICA	32
3.3.2 EL PRODUCTO EXTERIOR.....	34
3.3.3 EL PRODUCTO INTERIOR	37
3.3.4 EL PRODUCTO GEOMÉTRICO	38
3.4 ÁLGEBRA GEOMÉTRICA CONFORME.....	40
3.4.1 GEOMETRÍA CONFORME Y EL ESPACIO EUCLIDIANO	40
3.4.2 REPRESENTACIÓN DE OBJETOS.....	42
3.4.3 OBJETOS DEL ÁLGEBRA GEOMÉTRICA CONFORME.....	44
3.4.4 LA DISTANCIA Y EL PRODUCTO INTERIOR.....	46
4. MEMORIAS ASOCIATIVAS GEOMÉTRICAS.....	49
4.1 MODO PARA CLASIFICACIÓN DE PATRONES	49
4.1.1 CREACIÓN DE LAS VECINDADES HÍPER-ESFÉRICAS	50
4.1.2 NUEVO MÉTODO PARA CREAR VECINDADES ESFÉRICAS	51
4.1.3 FASE DE APRENDIZAJE	57
4.1.4 FASE DE CLASIFICACIÓN	57
4.1.5 CONDICIÓN PARA CLASIFICACIÓN PERFECTA	59
4.1.6 CONDICIÓN PARA CLASIFICACIÓN ROBUSTA	60
4.2 MODO PARA RESTAURACIÓN DE PATRONES	61
4.2.1 CREACIÓN DE LAS VECINDADES ESFÉRICAS	62
4.2.2 FASE DE APRENDIZAJE	65
4.2.3 FASE DE RESTAURACIÓN.....	65
4.2.4 CONDICIÓN PARA RESTAURACIÓN PERFECTA.....	66
4.2.5 CONDICIÓN PARA RESTAURACIÓN ROBUSTA	66
4.3 MODO PARA CLASIFICACIÓN NO SUPERVISADA	67
4.3.1 CLASIFICACIÓN NO SUPERVISADA CON ÁLGEBRA GEOMÉTRICA	68
4.3.2 NUEVO ALGORITMO DE AGRUPAMIENTO.....	68
5. EXPERIMENTACIÓN	71

5.1 EXPERIMENTACIÓN PARA EL MODELO DE CLASIFICACIÓN	
.....	71
5.1.1 EJEMPLOS NUMÉRICOS.....	71
5.1.2 EJEMPLOS CON PATRONES REALES.....	77
5.2 EXPERIMENTACIÓN PARA EL MODELO DE RESTAURACIÓN	
.....	80
5.2.1 EJEMPLOS NUMÉRICOS.....	80
5.2.2 EJEMPLOS CON PATRONES REALES.....	85
5.2.3 COMPARACIÓN CON OTROS MODELOS	92
5.3 EXPERIMENTACIÓN PARA CLASIFICACIÓN NO SUPERVISADA	
.....	97
5.4 DISCUSIÓN.....	102
6. CONCLUSIONES Y TRABAJO FUTURO	105
6.1 CONCLUSIONES	105
6.2 APORTACIONES	108
6.3 CONSIDERACIONES PARA TRABAJOS FUTUROS.....	109
6.4 PUBLICACIONES GENERADAS	110
APÉNDICE A. CÓDIGOS.....	121
CLASIFICACIÓN SUPERVISADA.....	121
RESTAURACIÓN DE PATRONES	127
CLASIFICACIÓN NO SUPERVISADA	130

ÍNDICE DE FIGURAS

FIGURA 1. EJEMPLOS DE OBJETOS DE UNA POBLACIÓN ESPECÍFICA.....	17
FIGURA 2. EJEMPLO DE RUIDO	18
FIGURA 3. EJEMPLOS DE CLASES LINEALMENTE Y NO LINEALMENTE SEPARABLES	20
FIGURA 4. RESTAURACIÓN DE UNA IMAGEN ALTERADA CON RUIDO	22
FIGURA 5. IMAGEN MUY AFECTADA CON RUIDO	23
FIGURA 6. MODELO DEL FILTRO PARA RECONOCIMIENTO ASOCIATIVO...	28
FIGURA 7. EJEMPLOS DE MULTI-VECTORES.....	35
FIGURA 8. TRES POSIBLES POSICIONES DE UN PUNTO CON RESPECTO A UNA HÍPER-ESFERA.	47
FIGURA 9. TRES SOLUCIONES PARA ENCONTRAR EL CENTRO Y EL RADIO DE UNA HÍPER-ESFERA.	51
FIGURA 10. MEMORIA ASOCIATIVA GEOMÉTRICA DESPUÉS DE LA FASE DE APRENDIZAJE.	58
FIGURA 11. EJEMPLO DE DOS PATRONES EN DENTRO DE UNA HÍPER-ESFERA	60
FIGURA 12. REPRESENTACIÓN GRÁFICA DE LAS CLASES DEL EJEMPLO 1	72
FIGURA 13. CÍRCULOS ÓPTIMOS DE LAS CLASES DEL EJEMPLO 1	73
FIGURA 14. REPRESENTACIÓN GRÁFICA DE LAS CLASES DEL EJEMPLO 2	75
FIGURA 15. ESFERAS ÓPTIMAS DE LAS CLASES DEL EJEMPLO 2	76
FIGURA 16. GRÁFICA CON LOS RESULTADOS DE LA CLASIFICACIÓN PARA EL EJEMPLO 3.....	79
FIGURA 17. GRÁFICA CON LOS RESULTADOS DE LA CLASIFICACIÓN PARA EL EJEMPLO 4.....	80
FIGURA 18. REPRESENTACIÓN GRÁFICA DE LOS PUNTOS DEL EJEMPLO 5.	80
FIGURA 19. SOLUCIÓN GRÁFICA AL PROBLEMA DE MINIMIZACIÓN.	81

FIGURA 20. HÍPER-ESFERAS CORRESPONDIENTES PARA LOS PATRONES DEL EJEMPLO 5.....	82
FIGURA 21. HÍPER-ESFERAS CORRESPONDIENTES PARA LOS PATRONES DEL EJEMPLO 6.....	83
FIGURA 22. SOLUCIÓN GRÁFICA PARA LAS HÍPER-ESFERAS 1 Y 2.	84
FIGURA 23. SOLUCIÓN GRÁFICA PARA LA HÍPER-ESFERA 3.....	84
FIGURA 24. HÍPER-ESFERAS CORRESPONDIENTES A LOS PATRONES DEL EJEMPLO 6.....	85
FIGURA 25. MUESTRAS DE ALGUNAS DE LAS IMÁGENES UTILIZADAS PARA EL EJEMPLO 7.....	86
FIGURA 26. MUESTRAS DE ALGUNAS DE LAS IMÁGENES UTILIZADAS PARA EL EJEMPLO 8.....	87
FIGURA 27. MUESTRAS DE IMÁGENES AFECTADAS CON RUIDO PARA EL EJEMPLO 7.....	87
FIGURA 28. GRÁFICA CON LA CLASIFICACIÓN PARA LOS CONJUNTOS DE ANIMALES Y CIENTÍFICOS.	88
FIGURA 29. MUESTRAS DE LAS IMÁGENES UTILIZADAS EN EL EJEMPLO 9.....	89
FIGURA 30. MUESTRAS DE IMÁGENES DESENFOCADAS.....	90
FIGURA 31. DOS IMÁGENES DESENFOCADAS PARA DISTINTOS VALORES DE K.	91
FIGURA 32. RESULTADO DE LA RECUPERACIÓN EN IMÁGENES DESENFOCADAS.	92
FIGURA 33. MUESTRAS DE IMÁGENES CON INFORMACIÓN PARCIALMENTE ELIMINADA	97
FIGURA 34. CONJUNTO DE PUNTOS EN A AGRUPAR.	99
FIGURA 35. DIFERENTES SOLUCIONES QUE RESUELVEN EL PROBLEMA DE LA FIGURA 34.....	99
FIGURA 36. OTRAS SOLUCIONES QUE RESUELVEN EL PROBLEMA DE LA FIGURA 34.....	99
FIGURA 37. MEJOR SOLUCIÓN QUE RESUELVE EL PROBLEMA DE LA FIGURA 34.	100

FIGURA 38. CONJUNTO DE PUNTOS PARA EL EJEMPLO 12.....	100
FIGURA 39. DIFERENTES SOLUCIONES QUE RESUELVEN EL PROBLEMA DE LA FIGURA 38.....	101
FIGURA 40. OTRAS SOLUCIONES QUE RESUELVEN EL PROBLEMA DE LA FIGURA 38.....	102
FIGURA 41. MEJOR SOLUCIÓN QUE RESUELVE EL PROBLEMA DE LA FIGURA 38.	102

ÍNDICE DE TABLAS

TABLA 1. SIGNIFICADO DEL VECTOR CONFORME S	43
TABLA 2. LISTA DE ENTIDADES DEL ÁLGEBRA GEOMÉTRICA CONFORME.	46
TABLA 3. RESULTADOS DE LA CLASIFICACIÓN PARA EL EJEMPLO 3	78
TABLA 4. RESULTADOS DE LA CLASIFICACIÓN PARA EL EJEMPLO 4	79
TABLA 5. RESULTADOS DE LA RESTAURACIÓN PARA LOS EJEMPLO 7 Y EL EJEMPLO 8.....	88
TABLA 6. COMPARACIÓN PARA LA RESTAURACIÓN DE UNA IMAGEN	93
TABLA 7. COMPARACIÓN PARA LA RESTAURACIÓN DE UNA IMAGEN	94
TABLA 8. ERROR CUADRÁTICO MEDIO PARA LAS IMÁGENES DE LA TABLA 6	95
TABLA 9. ERROR CUADRÁTICO MEDIO PARA LAS IMÁGENES DE LA TABLA 7	95

Resumen

En la literatura, existen muchos modelos para el reconocimiento de patrones cuya finalidad puede enmarcarse en la clasificación o la restauración de patrones. Las memorias asociativas son, precisamente, uno de estos modelos. Desde el siglo pasado han emergido muchos e interesantes modelos de memorias asociativas que resuelven estos problemas.

En el presente trabajo se describe un nuevo modelo de memoria asociativa que utiliza operaciones y operadores de la llamada *álgebra geométrica conforme* para su funcionamiento, las llamadas memorias asociativas geométricas. El álgebra geométrica conforme proporciona, a bajo nivel, un novedoso marco de trabajo, libre de coordenadas, para resolver problemas numéricos.

Se presentan tres modos de operación para las memorias asociativas geométricas; los cuales permiten resolver, precisamente, tres problemas en el área de reconocimiento de patrones: la clasificación supervisada, clasificación no supervisada y restauración de patrones.

Además, se incluye una amplia investigación sobre el estado del arte, junto con una introducción sobre los conceptos y definiciones más importantes de reconocimiento y clasificación de patrones, memorias asociativas y álgebra geométrica (en particular el álgebra geométrica conforme).

Dentro del trabajo, se incluyen condiciones formales de recuperación, clasificación y restauración perfecta y robusta, bajo las cuales el modelo puede funcionar. Además, se dan varios ejemplos con datos numéricos y con patrones reales, para mostrar la funcionalidad de las nuevas memorias. Las nuevas propuestas son, también, comparadas con otras propuestas reportadas en la literatura existente. Al final, se describen las conclusiones del trabajo junto con algunas ideas para trabajos futuros.

Abstract

In the literature several pattern recognition models whose main objective can be remarked in the classification or restoration of patterns, have been reported. Associative memories are, precisely one of these models. Many interesting models have emerged in the last years with this aim.

In this work, a novel associative memory model based on conformal geometric algebra principles is described. It uses operations and operators of the so-called *conformal geometric algebra*, the so-called geometric associative memories. At a low level, conformal geometric algebra provides a new coordinate-free framework for numeric processing in problem solving.

Three operation modes for the geometric associative memories are presented. With them one can solve three important pattern recognition problems: supervised classification, unsupervised classification and pattern restoration.

A widely research about the state of the art and an introduction of the significant notions and definitions about the pattern recognition and classification, associative memories and geometric algebra (particularly conformal geometric algebra) are included.

Formal conditions that ensure the correct functioning of the geometric associative memories are also provided. In particular those for perfect and robust classification (and restoration) are presented. Numerical and real examples to test the proposal are also given. A comparison with other reported state of the art proposals is also given. Conclusions and future work ideas are described too.

GLOSARIO DE TÉRMINOS

- **Álgebra geométrica.**

Enfoque libre de coordenadas de la geometría que posee un producto entre vectores asociativo e invertible.

- **Álgebra geométrica conforme.**

Esquema geométrico libre de coordenadas de 5 dimensiones construido a partir de puntos e hiper-esferas.

- **Agrupamiento.**

Asignación de un conjunto de patrones en sub-conjuntos llamados *grupos*; tal que, los patrones en el mismo *grupo* tengan características similares en algún sentido.

- **Clasificación o clasificación supervisada.**

Proceso de decidir si un patrón dado pertenece o no a una clase de equivalencia dada.

- **Clasificación no supervisada.**

Agrupar automáticamente un conjunto de patrones *sin etiquetar*.

- **Envoltura convexa.**

Frontera o borde del conjunto convexo mínimo que contiene a un conjunto no vacío y finito de puntos dados.

- **Memoria asociativa.**

Dispositivo que asocia un patrón de entrada con su correspondiente patrón de salida.

- **Multi-vector.**

En álgebra geométrica es el producto exterior de varios vectores.

- **Patrón.**

Representación numérica de un objeto dado; a partir de sus características que lo distinguen de otros objetos.

- **Producto exterior.**

Uno de los tres productos básicos del álgebra geométrica, generalmente usado para aumentar el grado de un multi-vector.

- **Producto geométrico.**

Uno de los tres productos básicos del álgebra geométrica, combina en una sola operación el producto interior y el exterior.

- **Producto interior.**

Uno de los tres productos básicos del álgebra geométrica, generalmente usado para disminuir el grado de un multi-vector.

- **Región o vecindad de soporte.**

En clasificación de patrones es un área geométrica donde los patrones que están dentro pertenecen a una clase en particular y los que están afuera no pertenecen a dicha clase.

- **Restauración.**

Eliminar, en algún sentido, el ruido presente en un objeto.

- **Separabilidad lineal.**

Dos clases son linealmente separables cuando se pueden separar por un hiper-plano.

- **Separabilidad esférica.**

Dos clases son esféricamente separables cuando se pueden separar por una hiper-esfera.

NOMENCLATURA

Para facilitar la lectura de este trabajo, los vectores euclidianos se denotarán con letras minúsculas en itálica y negrita (\mathbf{p} , \mathbf{q} , \mathbf{s}); las letras: i , j , k , l y m serán usadas para referirse a los índices de los objetos y variables. Los correspondientes puntos conformes (multi-vectores) serán denotados por letras mayúsculas en itálica (C , P , S). Una matriz euclidiana se denotará por una letra mayúscula en negrita (\mathbf{M} , \mathbf{H}). Finalmente, para denotar escalares se utilizarán letras griegas (α , λ , γ , δ).

Ahora bien para diferenciar un elemento que pertenece a un objeto (vector) determinado se utilizará un subíndice. Por su parte, para referirse a un objeto que pertenece a un conjunto de objetos del mismo tipo se utilizará un superíndice. Por ejemplo, sea S una hiper-esfera; S_k es, por tanto, el k -ésimo componente de dicha hiper-esfera; mientras que S^k será la k -ésima hiper-esfera en un conjunto de hiper-esferas.

CAPÍTULO

1. INTRODUCCIÓN

En 1844 el profesor de preparatoria Hermann Grassmann publicó un trabajo ambicioso titulado *Die Lineale Ausdehnungslehre, ein neuer Zweig der Mathematik* (La Teoría de la Extensión, una nueva rama de las matemáticas) más conocido como *Ausdehnungslehre* (Grassmann 1844). Según Grassmann ésta era, en efecto, *La rama* de las matemáticas que según sus propias palabras *superaba* a las demás.

Usando las ideas de su padre, Grassmann definió también *el producto exterior* o *producto combinatorio* (*äußeres Produkt* o *kombinatorisches Produkt*), el cual es la operación básica de lo que hoy se conoce como *álgebra exterior*. Su trabajo contiene las leyes de los espacios vectoriales pero, como contenía una multiplicación definida, sus estructuras satisfacen las propiedades de lo que ahora se conoce como álgebras. Los conjuntos de elementos linealmente independientes y dependientes están claramente contenidos en su trabajo como la idea de la dimensión.

En 1846 Grassmann ganó un premio de 45 ducados de oro por resolver uno de los problemas planteados por Leibniz: idear un cálculo geométrico desprovisto de coordenadas y propiedades métricas (lo que Leibniz llamó *analysis situ*). Sin embargo, sus métodos formales fueron duramente criticados y objetados por muchos matemáticos. Grassmann se justificó diciendo que estaba creando una teoría axiomática, lo que demostró que estaba muy adelantado para su tiempo.

Años más tarde, influenciado por los trabajos de Grassmann, el matemático inglés William K. Clifford presentó lo que ahora se conoce como *álgebra geométrica* o *álgebra*

de Clifford, en un artículo titulado: “Aplicaciones del álgebra exterior de Grassmann” (Clifford 1878). Él pensaba, al igual que Grassmann, que las álgebras de Grassmann y los cuaterniones de Hamilton se podían unir en un mismo sistema algebraico, haciendo un ligero cambio en el producto exterior.

Clifford admiraba el *Ausdehnungslehre* de Grassmann y estaba convencido que sus principios ejercerían una gran influencia en el futuro de las ciencias matemáticas. Pero debido a su temprana muerte, el análisis vectorial de Gibbs y Heaviside predominó en casi todo el siglo XIX y no el álgebra geométrica.

Al comienzo del siglo XX algunos investigadores, pensaban que si se combinan los trabajos de Clifford y Grassmann, casi todas las matemáticas podrían ser formuladas en un cálculo geométrico sencillo y universal, con fundamentos concretamente geométricos (Hestenes 2003).

Uno de ellos fue David Hestenes. Él, en particular, quería encontrar un lenguaje unificado para las matemáticas y la física (Hestenes 1986). Hestenes pensaba que con el álgebra geométrica se podrían resolver muchos de los problemas. En otras palabras, quería encontrar un sistema algebraico *universal* para las matemáticas y la física (Hestenes 1988). Mostró las ventajas que se podrían obtener al usar el álgebra geométrica en muchas áreas de la física y la geometría (Hestenes y Sobczyk 1984).

A partir de las publicaciones de Hestenes, muchos investigadores han obtenido buenos resultados al aplicar el álgebra geométrica en varias otras áreas. En ese sentido surge la pregunta de ¿por qué no usarla también en las ciencias de la computación?

Ya se han aplicado en algunas áreas, como las gráficas computacionales (Doran 2003), donde se han obtenido muy buenos resultados. Inclusive, hay algunos trabajos en el área del reconocimiento de patrones, como los que se presentarán en el capítulo **¡Error! No se encuentra el origen de la referencia..** En el presente trabajo se explorará una rama muy importante del reconocimiento de patrones: las memorias asociativas. En este caso, en el marco de trabajo del álgebra geométrica.

1.1 Objetivo general

El objetivo de este trabajo es desarrollar un nuevo modelo de memorias asociativas, las *memorias asociativas geométricas*. Estas memorias utilizarán operadores y operaciones del álgebra geométrica para su funcionamiento.

Se desarrollarán, a su vez, tres modos de operación para estas memorias: clasificación supervisada, clasificación no supervisada y restauración de patrones.

1.1.1 Objetivos específicos

- Estudiar el estado del arte para modelos de reconocimiento de patrones que utilicen álgebra geométrica.
- Desarrollar un nuevo modelo de memorias asociativas utilizando operadores del álgebra geométrica.
 - Adecuar el modelo para que opere en el modo de operación para clasificación supervisada.
 - Plantear las fases de entrenamiento y clasificación usando operaciones y operadores del álgebra geométrica.
 - Desarrollar las condiciones bajo las cuales este modelo funcionará correctamente.
 - Adecuar el modelo para que opere en el modo de operación para restauración de patrones.
 - Plantear las fases de entrenamiento y restauración usando operaciones y operadores del álgebra geométrica.
 - Desarrollar las condiciones bajo las cuales este modelo funcionará correctamente.
 - Adecuar el modelo para que opere en el modo de operación para clasificación no supervisada.
 - Modificar el algoritmo de k -medias para que funcione en el marco del álgebra geométrica.
 - Mejorar la complejidad del algoritmo original de k -medias.
 - Aplicar los modelos desarrollados en ejemplos numéricos y con patrones reales para mostrar su funcionamiento.
- Comparar los modelos desarrollados con otros modelos ya existentes.

- Presentar los resultados obtenidos en revistas y/o congresos internacionales de reconocimiento de patrones.
- Integrar a cada uno de los modelos sus respectivas condiciones para recuperación perfecta y robusta.

1.2 Justificación

Con este trabajo se pretenden resolver algunos de los problemas más importantes en los campos del tratamiento digital de patrones y el reconocimiento de patrones, usando, para ello, el álgebra geométrica.

De esta forma se estará dando un paso importante a la universalidad del álgebra geométrica, no solo para resolver problemas en matemáticas o física, sino también en el área de reconocimiento de patrones.

También se pretende abrir una nueva línea de investigación al combinar técnicas de la inteligencia artificial junto con el álgebra geométrica.

1.3 Aportaciones

El trabajo de tesis aporta las siguientes contribuciones a la ciencia, en particular en el área de la inteligencia artificial.

- Un nuevo modelo de memorias asociativas.
- Un algoritmo de clasificación que utiliza superficies de separación esféricas.
- Un nuevo modelo de clasificación supervisada de patrones en el marco del álgebra geométrica.
- Un nuevo modelo de clasificación no supervisada de patrones en el marco del álgebra geométrica.
- Un nuevo modelo de restauración de patrones en el marco del álgebra geométrica.
- Un modelo de memorias asociativas que funcionan ante ruido mezclado.
- Una mejora significativa al algoritmo de k-medias.

Un algoritmo para resolver un problema de optimización por mínimos cuadrados con álgebra geométrica en el álgebra convencional.

1.4 Estructura de la tesis

Este trabajo está estructurado de la siguiente manera:

El presente capítulo: Introducción, presenta el objetivo general y los objetivos específicos del trabajo. También se presenta la justificación de la tesis y las aportaciones a la ciencia. Además se resume la estructura general de este escrito.

El segundo capítulo: Estado del arte, se presentan algunos trabajos que tienen relación con el tema de este trabajo. En particular, clasificadores como redes neuronales o máquinas de soporte vectorial modelados con álgebra geométrica.

En el capítulo 3: Marco teórico, se presentan los conceptos teóricos que se utilizan en este trabajo, una breve introducción al reconocimiento de patrones y algunos conceptos sobre memorias asociativas. También se introducen conceptos sobre álgebra geométrica y el modelo conforme.

El capítulo 4: Memorias asociativas geométricas, presenta el nuevo modelo desarrollado, se divide a su vez en tres secciones, cada una describiendo un modo de operación de las memorias, se describen las fases que componen cada modelo así como las condiciones para que dichos modos de operación funcionen correctamente.

El siguiente capítulo: Experimentación, como su nombre lo indica muestra algunos ejemplos de operación, con valores numéricos y con patrones reales, de los modos de operación de las memorias asociativas geométricas descritas en el capítulo 4.

En el capítulo final: Conclusiones y trabajo futuro, se presentan las conclusiones finales de este trabajo, dando algunas ventajas de las memorias asociativas presentadas. También se describen, brevemente, algunas ideas sobre posibles trabajos futuros sobre el tema. Al final de este capítulo se enlistan las publicaciones que este trabajo de tesis generó.

CAPÍTULO

2. ESTADO DEL ARTE

En este capítulo se presenta un panorama de los trabajos de investigación más relacionados con el trabajo desarrollado en esta tesis y que, de alguna forma dieron pie a la presente investigación.

2.1 Redes Neuronales con Álgebra Geométrica

(Bayro-Corrochano y Vallejo 2001)

En este trabajo se presenta un nuevo conjunto de redes neuronales que utilizan operadores del álgebra geométrica. Las redes neuronales tradicionales, las complejas y las cuaterniónicas se pueden generalizar en el marco del álgebra geométrica. Para estos autores, los pesos, las funciones de activación y las salidas se representarán como multi-vectores¹. El producto geométrico se utiliza para realizar las operaciones de la red, en lugar del producto escalar.

Se propone una función de activación que afecta cada elemento base de los multi-vectores, la función de activación $f \cdot$ para un multi-vector A de dimensión n está dada por:

¹ Un multi-vector es el producto exterior de dos o más vectores, véase sección. **¡Error! No se encuentra el origen de la referencia.**

$$\begin{aligned}
f(A) &= f(A_0 + A_i\sigma_i + A_j\sigma_j + A_k\sigma_k + \dots + A_{ij}\sigma_i \wedge \sigma_j + \\
&\quad \dots + A_{ijk}\sigma_i \wedge \sigma_j \wedge \sigma_k + \dots + A_n\sigma_i \wedge \sigma_j \wedge \dots \wedge \sigma_n) \\
&= A_0 + f A_i \sigma_i + f A_j \sigma_j + f A_k \sigma_k + \dots + f A_{ij} \sigma_i \wedge \sigma_j + \\
&\quad \dots + f A_{ijk} \sigma_i \wedge \sigma_j \wedge \sigma_k + \dots + f A_n \sigma_i \wedge \sigma_j \wedge \dots \wedge \sigma_n
\end{aligned}$$

donde \wedge es el producto exterior², los valores de $f \cdot$ pueden ser de tipo gaussiano o sigmoidal.

Mientras que las neuronas de (McCulloch y Pitts 1943) usan el producto escalar del vector de entrada y su vector de peso, en ese modelo se propone una *neurona geométrica*. En dicha neurona se sustituye el producto escalar tradicional por el producto geométrico, esto es:

$$W^t X + \theta \Rightarrow wx + \theta = w \cdot x + w \wedge x + \theta$$

Una neurona geométrica puede ser vista como un tipo de operador de correlación geométrico que, a diferencia de las neuronas de McCulloch-Pitts, ofrecen interpolación no solo entre puntos sino entre multi-vectores de alto orden como planos, volúmenes, hiper-volúmenes, etc.

Una *red neuronal geométrica tipo feed-forward* se construye al cambiar el producto interior por el producto geométrico y las funciones de activación como se mencionó anteriormente.

En ese trabajo, se presenta una *generalización multidimensional* de regla de aprendizaje del gradiente descendiente en álgebra geométrica. La cual se utiliza para el perceptrón geométrico multicapa y para ajustar los pesos de las funciones geométricas de base radial.

² Para más información sobre el producto exterior véase la sección **¡Error! No se encuentra el origen de la referencia.**

2.2 Máquinas de Soporte Vectorial en Álgebra Geométrica

(Bayro-Corrochano y Vallejo 2001)

Las máquinas de soporte vectorial aplican métodos de optimización para el aprendizaje. Con estas máquinas se pueden generar redes de dos capas, redes de base radial y redes con otros *núcleos*. En ese trabajo se generan, con álgebra geométrica, redes neuronales usando máquinas de soporte vectorial. Las llamadas *máquinas de soporte multi-vectorial*.

Una máquina de soporte multi-vectorial mapea un espacio de entrada multi-vectorial en un espacio de características de alto orden. Estas máquinas construyen un hiper-plano de separación óptimo en el espacio multi-vectorial de características.

Usar álgebra geométrica con máquinas de soporte vectorial ofrece nuevas herramientas y un nuevo conocimiento sobre las mismas para aprendizaje multidimensional. Los autores realizan varios experimentos como estimación de movimiento rígido tridimensional y separación de dos conjuntos de puntos en tres dimensiones.

2.3 La neurona hiper-esférica

(Banarar, Perwass y Sommer 2003)

En este trabajo se propone una neurona de *alto orden* que, para separar clases, usa una superficie no lineal. Se presenta una extensión simple de un perceptrón, cuya superficie de decisión no es un hiper-plano sino una hiper-esfera. El perceptrón hiper-esférico multi-capas separa el espacio de entrada en regiones donde la clasificación es invariante a la orientación.

Esta hiper-esfera se representa en el espacio conforme del álgebra de Clifford y para decidir si un vector de entrada se encuentra dentro o fuera de ella se utiliza un producto escalar estándar. Para clasificar un vector de entrada X en una hiper-esfera normalizada S se utiliza el signo del producto interior $X \cdot S$.

Para construir un perceptrón cuya superficie de decisión es una hiper-esfera se utilizan las componentes de S como los pesos correspondientes.

La función de propagación de una neurona hiper-esférica se puede implementar como un producto escalar estándar. Durante la fase de entrenamiento, las componentes de S se consideran independientes. La neurona hiper-esférica puede verse como un perceptrón estándar con un segundo sesgo. La complejidad se puede comparar con añadir un perceptrón adicional a cada capa en un perceptrón multi-capas convencional.

El resultado del producto escalar es la entrada de la función de activación, los pesos en una neurona hiper-esférica se tratan como componentes independientes y representan una hiper-esfera no normalizada.

Por ejemplo; sea X la representación conforme de un vector de entrada x y sea S la representación de una neurona hiper-esférica con centro c y radio γ , además sea φ un escalar. Se considera la función sigmoideal $\sigma(\lambda, z) = \frac{1}{1 + e^{-\lambda z}}$ como la función de activación de la neurona. El entrenamiento de la neurona hiper-esférica, para clasificar a x como perteneciente a la clase que representa esa neurona consiste en variar c , γ y φ tal que $\sigma(\lambda, X \cdot S) > 1 - \varepsilon$ donde ε es el umbral de decisión.

Para un ε , c , y φ fijos; el radio de la hiper-esfera depende del parámetro λ de la función sigmoideal. Mientras más pequeño sea λ , más grande será el radio.

2.4 Agrupando formas irregulares con neuronas de alto orden

(Lipson y Siegelmann 2000)

En este trabajo se presenta un método de *clusterización* de arreglos de datos con formas irregulares con neuronas de alto orden. Las formas analíticamente complejas se modelan al reemplazar los pesos sinápticos clásicos de la neurona con tensores de alto orden en coordenadas homogéneas. Las formas de alto orden se formulan siguiendo el principio de activación de máxima correlación y la regla de aprendizaje de Hebb.

En una neurona tradicional, para seleccionar la neurona que exhiba la máxima correlación con la entrada, se utiliza un criterio de distancia mínima euclidiana. Este criterio conlleva varias dificultades, al utilizar el criterio de distancia mínima implica que las características del dominio de entrada deben ser esféricas.

El trabajo presenta un intento de generalizar el esquema esférico/elíptico a métricas más generales. En general, la restricción de forma de las neuronas clásicas se puede evitar al reemplazar el vector de pesos o la matriz de covariancia de una neurona clásica con un tensor de alto orden, capaz de formar correlaciones multilineales entre las señales asociadas a las neuronas. Esto permite crear formas con hoyos y/o áreas separadas.

Se utilizan, además las coordenadas homogéneas para combinar las correlaciones de diferente orden en un tensor sencillo. Estas neuronas de alto orden presentan buena estabilidad y buen desempeño al entrenarlas con aprendizaje Hebbiano simple.

2.5 Superficies de decisión esféricas usando el modelo conforme

(Perwass, Banarier y Sommer 2003)

Al incrustar el espacio euclidiano en un espacio conforme, las hiper-esferas se pueden expresar como vectores. El producto escalar entre puntos y hiper-esferas en el espacio conforme da una medida de qué tan lejos está un punto dentro o fuera de una hiper-esfera. Los autores muestran que una neurona hiper-esférica se puede implementar como un perceptrón con dos *sesgos*. También argumentan que un perceptrón multicapa basado en estas neuronas es similar a una red de base radial.

La idea básica de este trabajo es separar el espacio de entrada en dos clases utilizando hiper-esferas. La separación lineal, en general, no es suficiente para fines prácticos, ya que los datos están separados en varias clases y cada clase cubre una región en particular del espacio de entrada. Dependiendo de la estructura de los datos puede ser más útil usar neuronas que utilicen una superficie de separación no lineal en lugar de usar perceptrones. Dichas neuronas se conocen como *neuronas de alto orden*.

Los autores esperan que superficies de separación más complejas resuelvan una tarea con pocas neuronas, sin que el costo computacional se vea afectado. Para esto proponen una extensión simple del perceptrón, donde las superficies de separación son hiper-esferas en lugar de hiper-planos. La ventaja de esta representación es que, un producto escalar estándar es suficiente para decidir si un vector de entrada está dentro o fuera de la hiper-esfera.

En términos de complejidad concluyen que la complejidad de una red multi-capas hiper-esférica es la misma que la de un perceptrón multi-capas pero agregando un perceptrón a cada capa. Además, se puede interpretar como una extensión de una red de base radial con igual número de neuronas.

2.6 Cálculo neuronal con álgebra geométrica

(Buchholz 2005)

En esa tesis se presenta al álgebra geométrica como un marco de trabajo para el cálculo neuronal, o dicho de otra manera, el autor desarrolla un modelo de redes neuronales basado en el álgebra geométrica. Su principal objetivo es demostrar la utilidad del álgebra geométrica para el cálculo neuronal, debido a la naturaleza geométrica del mismo. En particular el autor desarrolla las llamadas *neuronas de Clifford*; parecidas a las neuronas clásicas, pero con las diferencias que los pesos y el sesgo se reemplazan por multi-vectores y se utiliza el producto de Clifford en lugar del producto tradicional.

Una neurona de Clifford básica se define de la siguiente manera:

$$Y = \sum_{i=1}^n W_i \otimes X_i + B,$$

donde Y , X , B y W_i son multi-vectores y \otimes es el producto de Clifford. Estas neuronas son parecidas a las neuronas de alto orden, ya que desde un punto de vista formal, ambas procesan funciones polinómicas a las entradas.

Una neurona de Clifford básica se puede ver como un asociador lineal; si se está trabajando con un álgebra geométrica de dimensión 1, éstas se comportan como neuronas tradicionales. Sin embargo, se desarrollaron neuronas de Clifford de orden superior, como las neuronas de Clifford básicas complejas, las hiperbólicas o las duales; las tres de dimensión 2. Con estas neuronas se pueden calcular transformaciones de varias entidades geométricas.

Algo interesante es que por medio de estas neuronas básicas se puede desarrollar un *asociador de Clifford*, que es simplemente una red neuronal con neuronas de Clifford

en lugar de las neuronas tradicionales y utiliza, como se mencionó anteriormente, el producto de Clifford, en lugar del producto tradicional.

Además de las neuronas básicas, se presentan las llamadas neuronas de Clifford *espinoriales*. A diferencia de su contraparte, la neurona básica, éstas se definen mediante la siguiente función:

$$Y = \sum_{i=1}^n W_i \otimes X_i \otimes f W + B,$$

donde $f W$ es la función reversión, conjugación o inversión. Las neuronas espinoriales son una arquitectura neuronal basada en dos productos geométricos. De ésta se derivan las *neuronas de Clifford espinoriales cuaterniónicas* y las *isomórficas*.

Ambas neuronas, las básicas y las espinoriales, pueden considerarse como un punto de partida para futuros trabajos sobre redes de perceptrones multicapa de segundo orden (Buchholz, Tachibana y Hitzer 2007).

CAPÍTULO

3. MARCO TEÓRICO

En este capítulo se presentan los conceptos principales que se abordan en este trabajo. La mayoría de estos conceptos y técnicas fueron obtenidos de libros y artículos cuyos autores son pioneros en el tema.

3.1 Reconocimiento y clasificación de patrones

3.1.1 El concepto de patrón

El concepto de *patrón* es algo universal dentro del área del reconocimiento de patrones. Por ejemplo, los patrones en datos biológicos contienen conocimiento pero sólo si se pueden reconocer. La discriminación de patrones en señales permite una identificación personal por voz, escritura, huellas dactilares, imágenes de rostros y más. Esto mismo se aplica al reconocimiento del habla, caracteres escritos y escenas en imágenes.

Los seres humanos somos capaces de reconocer, sin prácticamente ningún problema, el rostro de nuestros amigos, la forma de los caracteres y el texto escrito por otros, nuestras melodías favoritas, el olor de frutas frescas, las texturas del tronco de un árbol, los patrones ondulares en la ropa, la forma de las hojas, el significado contextual en algunas frases, y mucho más. Todo esto gracias a los respectivos patrones. Otros mamíferos son también muy hábiles en lo que al reconocimiento de patrones se refiere. Los sentidos pre-procesan señales tales como sonidos u ondas luminosas moduladas.

3.1.2 Clasificación de patrones

Cuando se determina que un objeto de una población dada pertenece a una única clase simple (de un solo elemento), se dice que se ha realizado el *reconocimiento de un objeto*. La *clasificación* es el proceso de agrupar objetos en una clase de acuerdo a sus diferencias o similitudes percibidas. Por ejemplo en la Figura 1, la imagen O^{10} , de esa población, de acuerdo a sus características es “la princesa Naussicaä”¹ (reconocimiento) y, también por sus características, pertenece a la clase “chicas” (clasificación).

El aprendizaje se realiza por un sistema cuando éste *graba* su experiencia dentro de cambios sistemáticos internos que causan que su comportamiento se modifique. Es un proceso que acumula y concentra experiencia dentro de las modificaciones internas del sistema. Los seres humanos aprenden de la experiencia acumulando reglas en varias formas. Por otra parte, el razonamiento es un proceso de aplicar ciertas reglas, ecuaciones, relaciones y demás, a una colección inicial de datos, hechos, etc. para deducir un resultado o una decisión. Mientras que el reconocimiento es una forma de razonamiento, la clasificación es una forma de aprendizaje.

Definición 1. *El problema de la clasificación consiste en decidir si un patrón pertenece o no a una clase de equivalencia dada.*

Definición 2. *El problema del reconocimiento consiste en decidir si algún patrón es equivalente a un patrón prototipo, o a un conjunto de esos patrones prototipos que podrían representar una clase.*

Cuando se aplica el reconocimiento a un patrón, éste podría estar alterado con ruido. En este caso, es más preciso hablar de una *restauración del patrón*. La restauración es importante para recuperar patrones originales a partir de patrones alterados o a partir de partes del patrón original.

¹ “*Nausicaä of the valley of the wind*”, escrita por Hayao Miyazaki; editorial Tokuma Shoten, 1982 (traducción al inglés por Viz Media Ed.).



Figura 1. Ejemplos de objetos de una población específica.

Sea a un patrón, \tilde{a} una versión alterada de a ; tal que $\tilde{a} = a + c$, donde c es el ruido. Dependiendo de la naturaleza de c , el ruido puede ser de tres tipos (para los fines de este trabajo):

- a) *Ruido aditivo*, en este caso todas las componentes de c son números positivos.
- b) *Ruido sustractivo*, cuando todas las componentes de c son números negativos.
- c) *Ruido mezclado*, en este caso, las componentes de c son números tanto positivos como negativos.

Por ejemplo, en la imagen de la Figura 2a se observa una imagen sin ruido. La imagen de la Figura 2b es la misma imagen pero afectada con un porcentaje de ruido aditivo, nótese que se ha agregado información a la imagen original. La imagen de la Figura 2c, por su parte, presenta ruido sustractivo y se observa que se ha quitado información a la imagen original. Por otro lado, en la imagen de la Figura 2d se ha eliminado y agregado información de la imagen original, por lo tanto presenta ruido mixto.



Figura 2. Ejemplo de ruido. a) Imagen original, b) imagen afectada con ruido aditivo, c) imagen afectada con ruido sustractivo, d) imagen afectada con ruido mezclado o mixto.

Un *sistema de reconocimiento o clasificación de patrones* es un sistema operacional que consta al menos de dos fases: a) el entrenamiento, aquí el sistema es alimentado por patrones muestra, generalmente en forma de vectores; y b) la decisión, donde el sistema reconoce o decide a que clase pertenece un patrón de entrada.

Existen dos modos de entrenamiento, el entrenamiento supervisado y el no supervisado. Cuando el sistema se alimenta con cada uno de los ejemplares de los patrones de entrada junto a su identificador de clase como la salida deseada y el sistema aprende a mapear los patrones de entrada con el identificador de clase deseado, entonces se está hablando del aprendizaje supervisado.

En el entrenamiento no supervisado, los ejemplares de entrada alimentan al sistema en alguna secuencia, y en ningún momento una entrada es mapeada a un identificador de clase, sino que el sistema mismo establece una nueva clase, asigna un identificador único a esa clase y ajusta sus parámetros para mapear el patrón de entrada con ese identificador.

3.1.3 Separabilidad lineal

La mayoría de sistemas de clasificación, como el clasificador lineal, requieren que los patrones a clasificar sean linealmente separables.

Definición 3. *Si existe un hiper-plano H en un espacio de rasgos tal que, todos los patrones (en forma de vectores) de una clase están a un lado de H , mientras que todos los patrones de la otra clase están del otro lado; entonces se dice que ambas clases son linealmente separables por H (Figura 3a).*

En tal caso sólo es necesario evaluar la ecuación del hiper-plano en un vector de entrada para determinar en qué parte del hiper-plano se encuentra dicho vector y asignarlo a la clase correspondiente.

Definición 4. *Si varias clases pueden ser separadas de dos en dos por hiper-planos, entonces son llamadas linealmente separables por pares, y la decisión se toma al determinar en qué lado de cada hiper-plano de separación se encuentra cada vector de entrada (Figura 3c).*

Definición 5. *Cuando dos clases no pueden ser separadas por un hiper-plano, se dice que dichas clases son no linealmente separables (Figura 3b y Figura 3d).*

En este caso se requiere de una superficie de separación de orden mayor para lograr tal objetivo. Este es el caso, por ejemplo, de las clases de la Figura 3b.

Sea Z un conjunto de clases en \mathbb{R}^n , y sea $Z^i \subset Z$ un subconjunto de patrones similares que pertenecen a la clase i , donde $i = 1, \dots, q$ y q es el número total de clases. Dado un patrón $x \in Z$, el problema de la clasificación consiste en decidir a cuál clase pertenece. La clase asignada será la que tenga patrones similares a x .

Existen muchas técnicas que resuelven este problema, una de ellas es la regla de decisión por el vecino más cercano (Cover y Hart 1967). Esta regla asigna un punto muestra no clasificado al conjunto más cercano de puntos clasificados. La principal desventaja de este método es la cantidad de tiempo que toma calcular todas las distancias.

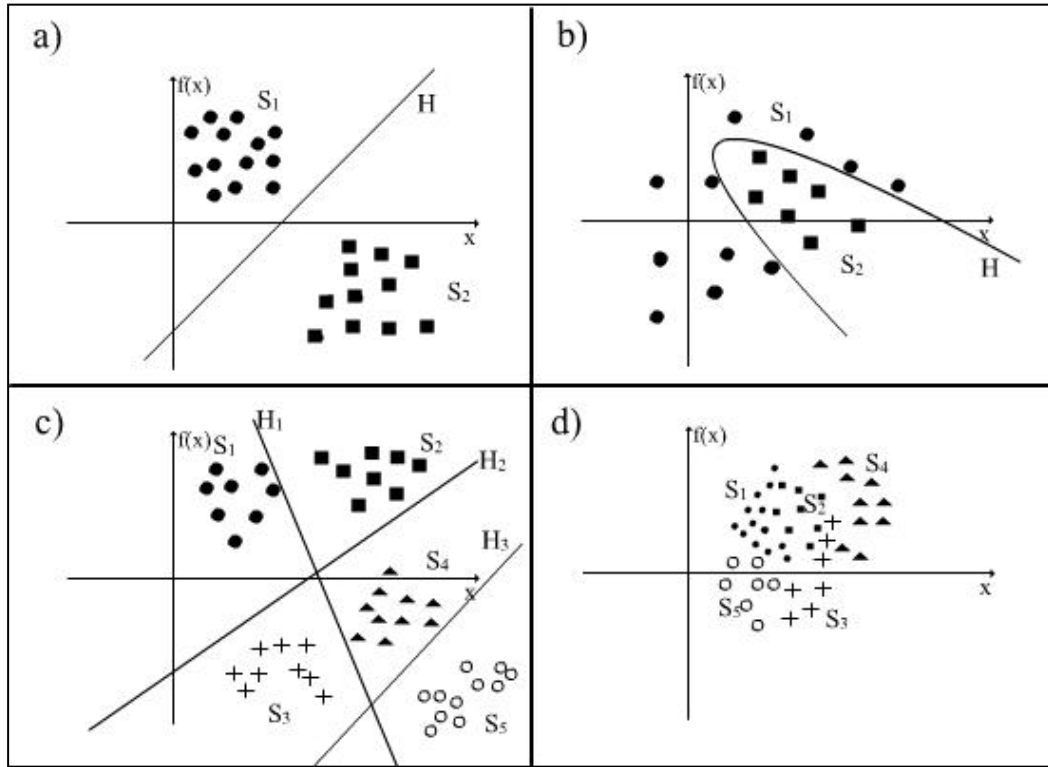


Figura 3. Ejemplos de clases linealmente y no linealmente separables: a) son linealmente separables, c) son linealmente separables por pares, b) y d) son no linealmente separables.

3.1.4 Regiones de soporte

Partiendo del hecho de que los patrones de una misma clase se encuentran agrupados, generalmente, cercanos unos a otros dentro del espacio de rasgos se puede utilizar otra aproximación: agrupar (*clusterizar*) el espacio de trabajo con las llamadas *regiones* o *vecindades de soporte*. En cada región hay solamente patrones de la misma clase. La clasificación se realiza cuando un patrón no entrenado cae dentro de una de las regiones de soporte, el patrón se asigna a la clase asociada a esa región.

Una primera solución es por medio del conocido *perceptrón morfológico* (Ritter 2004) donde las regiones de soporte son hiper-rectángulos. La finalidad es clasificar un patrón dentro de la llamada *región de encendido*. Durante la fase de entrenamiento se construyen las regiones de soporte de la memoria como hiper-cubos de tamaño α , donde α es la distancia mínima entre todos los patrones representantes. Una

desventaja de las regiones construidas de esta manera es que si dos de ellas están muy cerca, el tamaño de todas las demás vecindades se reduce drásticamente.

En (Barrón 2006) se presentó una mejora a este algoritmo. Se desarrolla un algoritmo con vecindades de soporte máximas, donde se calcula el promedio de variabilidad por patrón para obtener un umbral, de esta manera se supera la desventaja del modelo original, pero el método permite oclusiones entre las vecindades

Las regiones de soporte en ambos algoritmos son hiper-rectángulos, las memorias asociativas geométricas utilizan hiper-esferas en lugar de hiper-rectángulos. La discusión sobre si las vecindades hiper-esféricas son mejores que las hiper-rectangulares es un tema para futuras investigaciones.

Generalmente, las vecindades hiper-esféricas son difíciles de manejar, pero en el contexto del álgebra geométrica (más específicamente en el contexto del álgebra geométrica conforme), esto no ocurre. Un método para construir neuronas hiper-esféricas de forma iterativa se describe de forma detallada en (Banarier, Perwass y Sommer 2003).

3.1.5 Restauración de patrones

El problema de la clasificación, según la Definición 1, consiste en decidir si un patrón dado pertenece o no a una clase específica. Por su parte, el problema de la restauración de patrones es, básicamente, similar al problema de la clasificación.

Definición 6. El problema de la restauración de patrones consiste en decidir si un patrón, afectado por algún tipo de ruido, es, en algún sentido, igual a un patrón prototipo.

En otro sentido, la restauración de patrones consiste en compensar o deshacer, de alguna manera, los defectos que podrían degradar un patrón.

La restauración de patrones es un parte esencial de muchas aplicaciones en procesamiento digital de señales e imágenes. Actualmente, existe una gran necesidad de desarrollar algoritmo prácticos que permitan restaurar patrones afectados por ruido.

Durante los últimos años se han propuesto muchos modelos cuya finalidad sea resolver este problema. En la mayoría de los casos (sobre todo al utilizar imágenes, sonido o vídeo afectados por ruido), el problema se puede resolver aplicando ciertos filtros al objeto de entrada, con el fin de eliminar el ruido que podría contener. Los filtros pueden resolver, en cierto sentido, el problema de la restauración de patrones; los objetos restaurados con este tipo de métodos son, en la mayoría de los casos, diferentes de sus respectivos objetos originales. Esto se debe a que los filtros operan sobre el objeto en sí.

Por ejemplo, la imagen de la Figura 4B es una versión afectada con ruido de la imagen de la Figura 4A. En la Figura 4C y la Figura 4D se observa una imagen restaurada por medio de un filtro de promedio y un filtro morfológico (Gonzalez and Woods 2008), respectivamente.



Figura 4. Restauración de una imagen alterada con ruido. A) Imagen original, B) Imagen afectada con ruido, C) e D) Imágenes recuperadas.

Como se puede observar, la restauración no fue perfecta pero se pudo recuperar a cierta calidad. Sin embargo, en un objeto muy afectado con ruido, como el de la Figura 5, no es posible recuperar la información perdida utilizando los métodos tradicionales. Para ello es necesario contar con la información del objeto original codificada de

alguna manera. En estos casos surge la necesidad de recurrir a otros modelos de reconocimiento de patrones.

En particular, las redes neuronales se pueden usar para resolver este problema, por ejemplo los trabajos de (Chinarov y Menzinger 2003) y (Fukushima 2005). En dichos trabajos, se utiliza una red neuronal que converge, precisamente, en el patrón original.



Figura 5. Imagen muy afectada con ruido. No se puede recuperar con los filtros tradicionales.

Junto a las redes neuronales, las memorias asociativas, de las cuales se hablará en la sección 3.2, proveen una solución al problema de la restauración de patrones a partir de versiones distorsionadas de los mismos, refiérase por ejemplo a (Barrón 2006), (Sossa y Barrón 2003), (Ritter, Li y Wilson 1989), o (Anderson and Rosenfeld 1988).

Es importante mencionar que, a diferencia de los filtros, en las memorias asociativas utilizadas en la restauración de patrones, en general, el patrón restaurado resulta ser el patrón original. Ya que el patrón original se encuentra, en alguna manera, codificado dentro de la memoria, lo cual no ocurre con los filtros que toman el objeto original como entrada y tratan de reducir o eliminar, en algún sentido, el ruido añadido al objeto.

Las memorias asociativas geométricas, presentadas en este trabajo, no son la excepción.

3.1.6 Clasificación no supervisada

Generalmente, se tiene una clara distinción entre problemas de reconocimiento de patrones que son supervisados (clasificación) o no supervisados (agrupamiento), mientras que en los primeros solo se involucran datos con una etiqueta (clase), en los segundos solamente se utilizan datos no etiquetados (Duda, Hart y Stork 2001).

En general, los problemas de agrupamiento son más difíciles y presentan más retos que los problemas de clasificación (Jain 2008). El objetivo principal de la clasificación no supervisada es encontrar el agrupamiento natural en un conjunto de patrones, puntos u objetos sin conocer su respectiva etiqueta de clase. En otras palabras, consiste en desarrollar un algoritmo automático que descubra el agrupamiento natural en un conjunto de datos no etiquetados.

En la literatura existen muchos algoritmos que resuelven este problema. Uno de los más simples, pero muy efectivos, que desde su propuesta se sigue usando para resolver problemas de agrupamiento es el algoritmo *k*-medias (*k-means*) desarrollado por McQueen (McQueen 1967). La idea básica es definir *k* centros aleatorios, uno por cada agrupamiento. Después, tomar cada punto del conjunto de entrada y asociarlo al centro más cercano usando una distancia euclidiana. Se re-calculan los nuevos centros como el promedio de los agrupamientos formados. Entonces, con estos nuevos centros se repite el proceso hasta que el sistema se estabilice, o sea, hasta que los centros ya no cambien.

El algoritmo ISODATA es una variación del algoritmo *k*-medias, donde se separan y mezclan las agrupaciones (Jensen 1996). Las agrupaciones se mezclan si el número de patrones en una agrupación es menor que un cierto umbral o, si los centro de dos agrupaciones están lo suficientemente cerca de acuerdo a otro umbral establecido. Por otro lado, las agrupaciones se van a separar en dos agrupaciones diferentes si la desviación estándar excede un valor predefinido y el número de patrones es el doble de un umbral para el número mínimo de patrones en una agrupación.

Ambos algoritmos son muy sensibles a los valores de inicio generados. Para resolver este problema se podría escoger la clasificación con el menor error cuadrático medio; sin embargo, para dos valores de inicio diferentes, las diferencias con respecto al error cuadrático medio son, usualmente, muy pequeñas mientras que las clasificaciones son muy diferentes (Center for Earth Observation 2007). En general, visualmente no está muy claro que la clasificación con el menor error cuadrático medio es la mejor clasificación.

Entre los modelos de redes neuronales, que funcionan para clasificación no supervisada, los más comúnmente utilizados son los llamados *mapas auto-organizados* (Kohonen 2001) y *la teoría de resonancia adaptativa* (Carpenter y Grosberg 2003). Los mapas auto-organizados se basan en una organización topográfica en la cual las localidades más cercanas en el mapa representan entradas con propiedades o atributos similares. Por otra parte, la teoría de resonancia adaptativa permite variar el número de agrupaciones de acuerdo al tamaño del problema y, también, permite al usuario controlar el grado de similitud entre los miembros de la misma agrupación.

Existen otros métodos que resuelven el problema de la clasificación no supervisada que son similares a los anteriores, como los que utilizan clasificación difusa en lugar de una clasificación dura (Kaymak y Setnes 2000). En este caso, los patrones poseen un grado de pertenencia a las agrupaciones, en lugar de una pertenencia completamente a una agrupación. Esto es, los patrones en la frontera de una agrupación en particular, podrían pertenecer a esa agrupación en menor grado que los patrones ubicados en el centro de la agrupación.

En la literatura se pueden encontrar, también, los métodos basados en la teoría de grafos (Jardine y Sibson 1968). También hay que mencionar los métodos basados en análisis conceptual (Biswas, Weinberg y Fisher 1998). Y, por último, los métodos de agrupación espectral que hacen uso del espectro de la matriz de similitud de los patrones para reducir un poco la dimensión de las agrupaciones como el algoritmo de Shi-Malik (Shi y Malik 2000).

3.2 Memorias asociativas

3.2.1 Redes neuronales

Recientemente se ha demostrado que existen los ensambles neuronales, postulados como el rastro de la memoria en el cerebro (Carrillo-Reid, y otros 2008). Un rastro de memoria estaría dado por los pesos o fuerzas relativas que las conexiones van adquiriendo en el cerebro con el paso del tiempo. Las *redes neuronales artificiales* son un modelo matemático para representar información, basado en algunas características de las redes neuronales biológicas (Kishan, Chilukuri y Sanjay 1997). Consisten de un conjunto de elementos de procesamiento simples conectados entre sí y que envían la información a través de las conexiones.

En 1943 Warren McCulloch y Walter Pitts desarrollaban el primer modelo matemático de elementos binarios, conocido como *neurona* (McCulloch y Pitts 1943). Años más tarde, Donald Hebb postula un principio para un proceso de aprendizaje a nivel celular, *el aprendizaje Hebbiano* (Hebb 1949), indispensable para el desarrollo de los modelos de redes neuronales artificiales actuales. Casi una década después Frank Rosenblatt define el *perceptrón* como una máquina capaz de aprender con ejemplos (Rosenblatt 1958).

El no poder entrenar y clasificar la función lógica XOR por una capa de perceptrones, problema propuesto por Minsky y Papert en 1969 (Minsky y Papert 1988), disminuyó el interés sobre los perceptrones y otras neuronas artificiales durante casi dos décadas. La adición de otra capa de perceptrones llamada capa oculta forma una red que funciona como un poderoso modelo de clasificación no-lineal. Este perceptrón multicapa es también llamado red neuronal artificial hacia adelante (*feed-forward*).

El algoritmo más usado para entrenar una red neuronal es, sin duda, el de *retro-propagación* (*backpropagation*), basado en el método del gradiente descendente. Fue propuesto por primera vez en la tesis doctoral de Paul Werbos (Werbos 1974), pero fue hasta 1986 (Rumelhart, Hinton y Williams 1986) que el algoritmo fue reconocido e hizo renacer el área de las redes neuronales. En este algoritmo, el aprendizaje es de alguna forma independiente del orden en el cual los ejemplares son presentados;

además, su arquitectura puede ser manipulada para obtener mejores resultados y el modo de operación puede mejorarse con procesamiento paralelo. Sin embargo, también tiene algunas desventajas, por ejemplo, el entrenamiento puede converger a un mínimo local, lo cual podría dar resultados menos robustos; además, la tasa de aprendizaje no puede predecir el avance y podría ser o muy pequeña, en cuyo caso se necesitan muchos pasos para converger, o muy grande y entonces oscilaría en lugar de converger, entre otras.

En la práctica, la convergencia generalmente ocurre, particularmente con pequeños a moderados tamaños de perceptrones multicapa. Cuando no ocurre, podría converger en otra corrida ajustando los pesos de inicio, o requeriría modificar la arquitectura.

A finales de la década de los noventa, el área de las redes neuronales artificiales dio un cambio repentino debido a los trabajos de (Ritter, Li y Wilson 1989), quienes presentan un nuevo tipo de redes neuronales llamadas *redes neuronales morfológicas*. En ellas las operaciones clásicas de multiplicación y adición son reemplazadas por sumas y máximos (o mínimos), respectivamente. Este nuevo tipo de redes neuronales dieron lugar a una nueva herramienta en el área de reconocimiento de patrones, las memorias asociativas morfológicas.

3.2.2 Conceptos sobre memorias asociativas

El concepto de memoria puede ser entendido de diferentes maneras. Generalmente, involucra un *mecanismo de almacenamiento* el cual utiliza un *medio* de almacenamiento. Esta operación es llamada la *función memoria*, la cual coopera con las demás funciones del organismo. Gracias a estas operaciones, la memoria puede generar fenómenos muy complejos, desde simples ideas mentales hasta complejas secuencias de pensamiento.

Las operaciones de procesamiento de información dentro del cerebro pueden ser expresadas en términos de *funciones de filtros adaptativos*. La idea general es que el cerebro está organizado en un número de *unidades funcionales*. Una de estas unidades funcionales, que es una parte de las tantas redes neuronales en el cerebro, es un

sistema bien definido de *entradas* y *salidas* conocido como filtro para reconocimiento o recuperación asociativo (Figura 6).

La operación básica de dicho filtro es transformar un conjunto de valores de señales de entrada, denotado como x , en otro conjunto de señales de salida denotado como y .

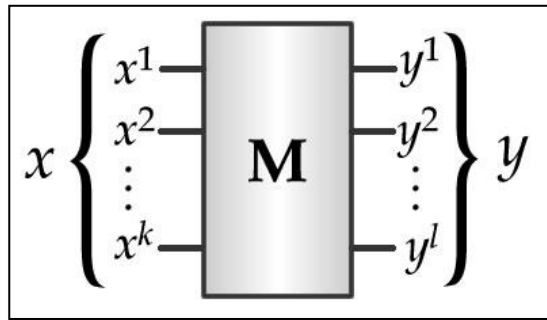


Figura 6. Modelo del filtro para reconocimiento asociativo

En las redes neuronales artificiales que juegan el papel de memorias asociativas, un dato de entrada es almacenado conjuntamente en forma de una matriz de pesos, la cual es usada para generar la salida que correspondería con la entrada correspondiente.

Asociación es el proceso de mapear patrones de entrada con patrones de salida, incluso si los patrones de entrada fueran alterados. Por ejemplo, una palabra y su significado deben ser asociados correctamente al leer una oración a pesar de que la palabra haya sido escrita incorrectamente.

Definición 7. Una memoria asociativa \mathbf{M} es un dispositivo que asocia patrones de entrada con patrones de salida.

Esto es, al presentarle a \mathbf{M} un patrón de entrada x (en forma de vector), ésta responde con el correspondiente patrón (vector) de salida y . En este sentido, podría decirse que una memoria asociativa es una red neuronal de una sola capa.

La notación para una asociación entre dos vectores x e y puede verse como la de un par ordenado (x, y) . El conjunto de todas las asociaciones que servirá para formar la memoria asociativa es conocido como *conjunto fundamental* y se representa como:

$$x^k, y^k \quad | k = 1, 2, \dots, m, \quad (3.1)$$

donde m es el número total de asociaciones. A los patrones que forman las asociaciones del conjunto fundamental se les conoce como *patrones fundamentales*.

Las asociaciones son almacenadas conjuntamente en una matriz de pesos, la cual es usada para generar el patrón de salida que corresponde con el patrón de entrada asociado. Esta matriz de pesos es precisamente la memoria asociativa.

El proceso de construir dicha matriz es llamado *aprendizaje* o *entrenamiento*, mientras que el obtener un patrón de salida cuando un patrón de entrada es presentado a la memoria es llamado *recuperación* o *clasificación* (dependiendo del uso que se le dé a la memoria).

En la mayoría de los modelos de memorias asociativas, las conexiones entre los patrones de entrada y los de salida se ajustan únicamente en la fase de entrenamiento, después de esta fase no se pueden volver a ajustar. En (Vázquez y Sossa 2008) se presenta un nuevo modelo dinámico, en el cual la sinapsis de una memoria asociativa se puede ajustar aún después de la fase de entrenamiento como respuesta a un estímulo de entrada.

Definición 8. *Cuando al presentar a una memoria \mathbf{M} un patrón fundamental como entrada, si ésta responde con su correspondiente patrón de salida, se dice que la recuperación para ese patrón fue perfecta. Una memoria que haga recuperaciones perfectas para todos los patrones fundamentales se dice que es una memoria con recuperación perfecta.*

Definición 9. *Por otra parte, si una memoria asociativa \mathbf{M} puede recuperar o clasificar patrones del conjunto fundamental pero afectados con ruido, se dice que \mathbf{M} es una memoria con recuperación robusta.*

La robustez de una memoria asociativa depende de dos factores: del tipo de ruido al que se sometió el patrón fundamental y, en mayor medida, a la cantidad de ruido con que se afectó dicho patrón.

3.2.3 Modelos de memorias asociativas

En las últimas cinco décadas han aparecido varios modelos de memorias asociativas, comenzando con la *Lernmatrix* de Steinbuch (Steinbuch 1961) el primer modelo de memoria asociativa. Años después el *Linear Associator* tuvo su origen en los trabajos de Anderson (Anderson 1972) y Kohonen (Kohonen 1972), en el mismo año Kaoru Nakano presenta su *Associatron* (Nakano 1972). La siguiente contribución importante fue realizada por Hopfield, con la *Memoria Hopfield* (Hopfield 1982) y posteriormente la *Memoria Asociativa Bidireccional* de Kosko (Kosko 1988), una mejora de este modelo se puede encontrar en (Vázquez y Sossa 2008).

En la actualidad, por medio de la llamada *programación genética* se han desarrollado otras memorias asociativas que ofrecen resultados, si no mejores muy similares a los modelos desarrollados hasta la fecha (Villegas-Cortez, y otros 2009).

Casi 20 años después del gran éxito que tuvo la memoria Hopfield, hicieron su aparición un nuevo tipo de memorias asociativas basadas en el álgebra de rejillas (*lattices*), las *Memorias Asociativas Morfológicas* (Ritter, Li y Wilson 1989). En estas memorias se usan máximos o mínimos de sumas para sus operaciones, en contraste con la suma de productos usada en los modelos clásicos. Una variante de estas memorias apareció algunos años después, las *memorias asociativas $\alpha\beta$ (alfa-beta)* (Yáñez y Díaz de León 2003) cuyas operaciones se basan en dos nuevos operadores: α (*alfa*) y β (*beta*).

Existen dos tipos de memorias asociativas morfológicas (y alfa-beta): las memorias *min*, que son capaces de responder ante patrones alterados con ruido sustractivo y las memorias *max* que funcionan muy bien frente al ruido aditivo. Sin embargo, contra lo que se podría pensar, su funcionamiento se ve frenado frente al ruido mezclado, el cual es el más común.

Intuitivamente, este problema podría resolverse realizando una combinación de una memoria asociativa *min* con una *max*; pero, como se muestra en (Sussner 2000), esto no es posible. Hay tres formas de resolver el problema del ruido mezclado. La primera es por medio de los llamados *núcleos (kernels)* propuestos en (Sussner 2000).

Las memorias tipo *mediana* (Sossa y Barrón 2003) que utilizan para su operación el operador mediana (*med*) son otra alternativa al problema del ruido mezclado, una ventaja significativa que poseen es que a diferencia de los otros tipos de memorias, éstas pueden ser usadas con patrones reales (Sossa, Barrón y Vázquez 2004), lo cual es sumamente útil.

En (Cruz, Sossa y Barrón 2007) se describe un nuevo método para atacar el problema del ruido mezclado sin usar núcleos o el operador mediana, basado en la localidad del ruido. Mediante los llamados sub-patrones. La idea básica es dividir al objeto original en partes, a cada parte hay que obtener su patrón correspondiente (sub-patrón); con cada uno de los sub-patrones del objeto se entrena una memoria asociativa. Debido al carácter local del ruido, algunos sub-patrones no se verán afectados por él.

Claro está, estas tres alternativas tienen sus desventajas. Encontrar un buen *núcleo* es difícil, además de que su uso está muy restringido. Si al conjunto fundamental se le agrega dinámicamente un nuevo patrón parecido a uno existente, hay que re-calcular todos los núcleos de los demás patrones para evitar repeticiones. El operador mediana en la práctica es muy costoso de implementar y es difícil cumplir las condiciones para recuperación perfecta. Por otro lado, el uso de los sub-patrones puede consumir mucho tiempo de cómputo a menos que se use procesamiento paralelo.

Se puede observar que, los primeros modelos de memorias asociativas utilizaban, para sus operaciones, el álgebra convencional; es decir, sumas y productos entre vectores y/o matrices. Los modelos morfológicos utilizaban por su parte, operadores de la morfología matemática; es decir, máximos, mínimos y mediana de sumas y restas. En la presente tesis se presenta un nuevo modelo de memorias asociativas que, para su funcionamiento utilizan operadores y operaciones del álgebra geométrica.

3.3 Álgebra Geométrica

A finales del siglo XIX el matemático inglés William K. Clifford presentó lo que ahora se conoce como *Álgebra geométrica* o *Álgebra de Clifford*, en un artículo titulado Aplicaciones del álgebra exterior de Grassmann (Clifford 1878). Él pensaba (al igual que Grassmann) que las álgebras de Grassmann y los cuaterniones de Hamilton se

podían unir en un mismo sistema algebraico, haciendo un ligero cambio en el producto exterior.

Pero fue hasta los años sesenta cuando álgebra geométrica tuvo un gran impulso en muchas áreas de la ciencia, gracias a los trabajos de David Hestenes. Hestenes mostró las ventajas que se podrían obtener al usar el álgebra geométrica en muchas áreas de la física y la geometría (Hestenes y Sobczyk 1984). Posteriormente muchos otros investigadores siguieron sus pasos y mostraron las ventajas de utilizar el álgebra geométrica en sus respectivas áreas de investigación.

3.3.1 Propiedades del álgebra geométrica

El álgebra geométrica es un enfoque libre de coordenadas de la geometría que, a diferencia del álgebra vectorial convencional, tiene un producto entre vectores asociativo e invertible. Además permite la adición de objetos de diferente tipo, como escalares y vectores, muy parecido a la forma en que se construye un número complejo a partir de una parte real y una parte imaginaria.

En el álgebra geométrica los objetos geométricos y los operadores de estos objetos son tratados en una sola álgebra. En otras palabras, es sencilla, compacta, es libre de coordenadas y permite formulaciones dimensionalmente fluidas. Algunas propiedades del álgebra geométrica se enuncian a continuación.

- **Geoméricamente intuitiva**

Una característica muy especial del álgebra geométrica es que es geoméricamente su intuitiva. Por ejemplo las esferas y los círculos son objetos algebraicos con significado geométrico. Se puede representar un círculo tomando dos esferas y realizando la intersección entre ellas.

- **Unificación**

El álgebra geométrica incluye también muchos otros sistemas matemáticos como el álgebra vectorial, el álgebra de Grassmann, los números complejos y los cuaterniones, entre otros.

- **Poca complejidad simbólica**

Las expresiones en álgebra geométrica normalmente tienen poca complejidad.

En el álgebra geométrica es posible la división entre elementos (no nulos), algo que el análisis vectorial de Gibbs y Heaviside no puede hacer. Lo cual es muy importante en el cálculo geométrico al calcular las derivadas inversas.

Al manejar los números complejos como cierta sub-álgebra del álgebra geométrica, la unidad imaginaria recibe interpretaciones geométricas concretas. Además, el producto exterior de Grassmann se define para cualquier dimensión, a diferencia del análisis vectorial convencional que no se puede extender tan fácilmente a otras dimensiones. La *electrodinámica de Maxwell* se puede formular en una ecuación usando el álgebra geométrica. La mecánica cuántica relativista se ha reformulado en términos del álgebra geométrica reemplazando las matrices de Dirac por vectores base reales. Finalmente, existen muchas implementaciones de la *relatividad general* usando el álgebra geométrica (Hitzer 2003).

El *cálculo geométrico* (Hestenes y Sobczyk 1984) (junto con el análisis de Clifford) es un lenguaje para expresar y analizar todos los conceptos geométricos en matemáticas. El álgebra de Clifford provee la gramática (Hitzer 2003). Los números complejos, cuaterniones, el álgebra de matrices, los vectores, los tensores, el cálculo de *spinors* y las formas diferenciales se pueden integrar en un sistema sencillo y entendible.

Algunas áreas en las que se ha trabajado con álgebra geométrica son: visión por computadora, gráficos y reconstrucción, robótica, procesamiento de imágenes y señales, investigación aeroespacial, teoría de control, computación cuántica, bioingeniería y diseño molecular, electromagnetismo y propagación de ondas, álgebra simbólica y algoritmos numéricos.

Un álgebra geométrica se construye sobre un espacio vectorial usando un producto específico. Los principales productos del álgebra geométrica son: *el producto geométrico* o *producto de Clifford*, *el producto exterior* y *el producto interior*. Los tres se explican brevemente a continuación.

3.3.2 El producto exterior

En el álgebra vectorial convencional existen dos productos entre vectores: el producto *escalar* (producto *interno* o producto *punto*) que produce un escalar y el producto *vectorial* (producto *cruz*) que produce otro vector.

Sea un espacio vectorial euclidiano n -dimensional. El producto escalar $\mathbf{a} \cdot \mathbf{b}$ de dos vectores \mathbf{a} y \mathbf{b} es un escalar con magnitud $|\mathbf{a}||\mathbf{b}|\cos\theta$. El producto vectorial ($\mathbf{a} \times \mathbf{b}$) es un tercer vector con magnitud $|\mathbf{a}||\mathbf{b}|\sin\theta$ en la dirección perpendicular al plano que contiene a los vectores \mathbf{a} y \mathbf{b} . Donde $|\mathbf{a}|$ y $|\mathbf{b}|$ son las longitudes de los vectores \mathbf{a} y \mathbf{b} respectivamente y θ es el ángulo entre ellos.

En el álgebra geométrica el producto interior es particularmente útil y puede coincidir con el producto escalar tradicional, pero en general representa una operación algebraica que no resulta en un escalar, ya que no depende de una métrica.

Sin embargo, el producto vectorial es particularmente deficiente. Por ejemplo, en dos dimensiones un vector perpendicular a los vectores \mathbf{a} y \mathbf{b} no tiene significado y en cuatro o más dimensiones no está definido (Perwass and Hildebrand 2004). Evidentemente, es necesario un producto más general que contenga información direccional y sea generalizable a todas las dimensiones. En este caso se está hablando de un nuevo producto: el *producto exterior* (o producto *externo*). Éste se presentó por primera vez en 1844 en los trabajos de Grassmann (Grassmann 1844).

Sin explicar a grandes rasgos lo que es, se definirá un álgebra geométrica sobre los reales \mathbb{R}^n (\mathbb{C} \mathbb{R}^n ó \mathbb{C}_n); el producto exterior de dos vectores está definido en esta álgebra (Perwass y Hildebrand 2004) y se denota por el símbolo \wedge . De acuerdo a su definición, éste permite obtener todas las propiedades geométricas del plano (Dorst, Fontijne y Mann 2007).

Sean \mathbf{a} , y \mathbf{b} vectores en un espacio euclidiano n -dimensional. El producto exterior se escribe como en la expresión (3.2) y tiene magnitud $|\mathbf{a}||\mathbf{b}|\sin\theta$. El resultado no es un escalar o un vector, sino más bien un *bivector*.

$$a \wedge b. \tag{3.2}$$

El resultado de aplicar el producto exterior entre ellos es un área dirigida en el segmento del plano que contiene a los dos vectores a y b , esta cantidad es el bivector propiamente dicho (ver la Figura 7-2).

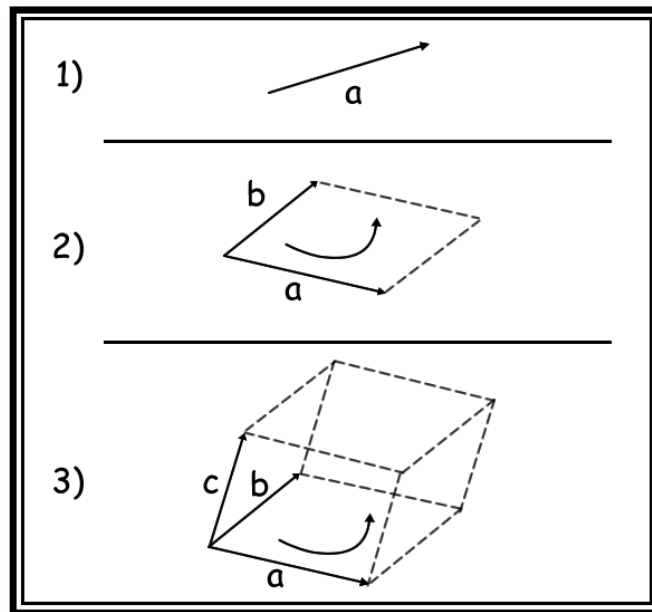


Figura 7. Ejemplos de multi-vectores. (1) Multi-vector de grado 1. (2) Multi-vector de grado 2. (3) Multi-vector de grado 3.

Una característica de la interpretación geométrica del producto exterior es que es generalizable a dimensiones superiores. Si c es, también un vector, la interpretación de $a \wedge b \wedge c$ se puede ver como un volumen tridimensional dirigido, al cual se le llama *trivector*, como se observa en la Figura 7-3.

Sean a y b los dos vectores mostrados en la Figura 7-2, la orientación de a y b está definida por el sentido contrario a las manecillas del reloj y la orientación de $b \wedge a$ es la dirección contraria, es decir en el sentido de las manecillas del reloj. Por lo tanto el producto exterior es *anti simétrico* como se observa en la expresión (3.3).

$$a \wedge b = - b \wedge a . \tag{3.3}$$

Cuando \mathbf{b} es igual al vector \mathbf{a} se obtiene un resultado algebraico inusual, éste se muestra en (3.4), sugiere que el cuadrado de \mathbf{a} con este producto debe ser cero (Dorst, Fontijne y Mann 2007). Razonablemente suena correcto, pues no se puede codificar un segmento del plano con el mismo vector.

$$\mathbf{a} \wedge \mathbf{a} = -\mathbf{a} \wedge \mathbf{a} = 0. \quad (3.4)$$

El producto exterior es asociativo, distributivo con respecto a la suma y saca escalares. Sean e_1, e_2 las bases en el plano bidimensional y sean \mathbf{a} y \mathbf{b} dos vectores en este plano, ambos se pueden expresar como en la expresión (3.5).

$$\begin{aligned} \mathbf{a} &= a_1 e_1 + a_2 e_2 \\ \mathbf{b} &= b_1 e_1 + b_2 e_2. \end{aligned} \quad (3.5)$$

Con base en esto se puede reducir el producto exterior entre \mathbf{a} y \mathbf{b} a:

$$\mathbf{a} \wedge \mathbf{b} = a_1 b_2 - a_2 b_1 e_1 \wedge e_2. \quad (3.6)$$

De la misma forma, en el espacio tridimensional con bases e_1, e_2, e_3 el producto exterior se reduce a:

$$\begin{aligned} \mathbf{a} \wedge \mathbf{b} &= a_1 b_2 - a_2 b_1 e_1 \wedge e_2 + a_2 b_3 - a_3 b_2 e_2 \wedge e_3 \\ &+ a_3 b_1 - a_1 b_3 e_3 \wedge e_1. \end{aligned} \quad (3.7)$$

El producto exterior de dos vectores en el plano o en el espacio se puede escribir como la suma escalada de la combinación de sus elementos bases. El valor de la suma se puede calcular usando los determinantes de los elementos que conforman cada vector (Dorst, Fontijne y Mann 2007).

Sea \mathcal{C}_n un álgebra geométrica, ésta contiene las bases mostradas en (3.8), al elemento base de más alto grado se le conoce como el *pseudoescalar* y se representa por el símbolo I .

$$1 \quad e_1 \quad e_1 \wedge e_2 \quad e_1 \wedge e_2 \wedge e_3 \quad \dots \quad e_1 \wedge e_2 \wedge e_3 \wedge e_4 \quad (3.8)$$

Definición 10. En cualquier dimensión n , el producto exterior de k vectores o $A_{\langle k \rangle}$ se define como en (3.9), para $k \geq n$. A $A_{\langle k \rangle}$ se le conoce como k -vector o multi-vector. Los multi-vectores son los elementos básicos del álgebra geométrica.

$$A_{\langle k \rangle} = \mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \dots \wedge \mathbf{a}_k = \bigwedge_{i=1}^k \mathbf{a}_i \quad (3.9)$$

Cualquier multi-vector de \mathcal{C}_n se puede expresar en términos de las bases mostradas en (3.8).

A la dimensión de cualquier multi-vector también se lo conoce como su *grado*; es decir, un escalar tiene grado 0, un vector tiene grado 1, un bivector tiene grado 2 y así sucesivamente (ver Figura 7). Dentro del álgebra geométrica es posible manipular expresiones que contengan multi-vectores de cualquier grado; esto es, la suma de un escalar y un bivector está perfectamente definida.

3.3.3 El producto interior

Otra operación importante en el álgebra geométrica es el *producto interior* (producto interno), se denota por el símbolo \cdot . Si A , B y C son multi-vectores, el producto interior se representa:

$$B \cdot A. \quad (3.10)$$

El producto interior reduce el grado del vector B por λ , y el producto exterior aumenta el grado de B por λ , donde λ es el grado del vector A . Si A y B son vectores de grado uno, el producto interior es el mismo que el producto escalar tradicional, cuyo resultado es un escalar.

El producto interior es conmutativo y es distributivo bajo la suma, es decir se cumplen las expresiones mostradas en (3.11).

$$\begin{aligned}
A \cdot B &= B \cdot A \\
(A + B) \cdot C &= A \cdot C + B \cdot C .
\end{aligned}
\tag{3.11}$$

Para dos multi-vectores cualesquiera, la definición de producto interno y externo está dada por (3.12) y (3.13) respectivamente.

$$A \wedge B = \sum_r A_r \wedge B = \sum_s A \wedge B_s = \sum_{r,s} A_r \wedge B_s \tag{3.12}$$

$$A \cdot B = \sum_r A_r \cdot B = \sum_s A \cdot B_s = \sum_{r,s} A_r \cdot B_s. \tag{3.13}$$

3.3.4 El producto geométrico

Los productos interior y exterior, proporcionan información sobre la magnitud y dirección de los vectores. Sería deseable combinar la información proporcionada por ambos productos en un simple producto. Esto es lo que hace, precisamente, el *producto geométrico* (o *producto de Clifford*) de dos multi-vectores A y B ; éste se define de la siguiente manera:

$$AB = A \cdot B + A \wedge B. \tag{3.14}$$

Si A y B son paralelos, entonces su producto exterior es cero y el producto geométrico AB es igual al producto interior. Si A y B son perpendiculares entonces su producto interno es cero y el producto geométrico AB es igual al producto exterior. Es decir se cumplen las expresiones (3.15) y (3.16).

$$AB = A \cdot B \leftrightarrow A \parallel B, \tag{3.15}$$

$$AB = A \wedge B \leftrightarrow A \perp B. \tag{3.16}$$

De acuerdo con estas expresiones, con el producto geométrico los vectores ortogonales anti conmutan y los vectores paralelos conmutan. Por lo tanto, este producto codifica las relaciones geométricas básicas entre vectores (Doran 2003).

El producto geométrico es asociativo y distributivo con respecto a la suma, como se observa en las expresiones (3.17). Además la multiplicación por un escalar λ está definida, como en la expresión (3.18).

$$\begin{aligned} A \cdot BC &= AB \cdot C, \\ A \cdot (B+C) &= AB + AC, \\ (B+C) \cdot A &= BA + CA. \end{aligned} \tag{3.17}$$

$$\lambda A = A\lambda. \tag{3.18}$$

Una propiedad que hay que tener en cuenta es que el producto geométrico no es conmutativo, como se observa en las expresiones (3.19). Mientras que el producto interior es simétrico, el producto exterior no lo es, como se mostró en (3.3). Por lo tanto, el producto geométrico tampoco lo es; es decir se cumple la expresión (3.20).

$$\begin{aligned} AB &= A \cdot B + A \wedge B \\ BA &= A \cdot B - A \wedge B \end{aligned} \tag{3.19}$$

$$AB \neq BA \tag{3.20}$$

A partir de las expresiones (3.19) se definen los productos interior y exterior en función de las partes simétricas y anti simétricas del producto geométrico, como se observa en (3.21).

$$\begin{aligned} A \cdot B &= \frac{1}{2} (AB + BA) \\ A \wedge B &= \frac{1}{2} (AB - BA) \end{aligned} \tag{3.21}$$

Una de las propiedades del álgebra geométrica, en particular de sus dos productos, es que pueden elevar (producto externo) o bajar (producto interno) de grado según las necesidades, con esto es posible moverse entre sub-espacios de diferente dimensión.

3.4 Álgebra Geométrica Conforme

3.4.1 Geometría Conforme y el espacio euclidiano

Sea un espacio vectorial euclidiano n -dimensional. El grupo conforme es el conjunto de todas las transformaciones en \mathbb{R}^n que dejan los ángulos invariantes pero pueden alterar las distancias. Para especificar un punto en este espacio de dos dimensiones más, son necesarias dos restricciones (Doran 2003).

La primera es que la representación es homogénea, por lo tanto X y λX representan el mismo punto en el espacio euclidiano, donde λ es un escalar. La segunda restricción es que el vector X es nulo, es decir se cumple (3.22).

$$X^2 = 0 \quad (3.22)$$

Sean e_1, e_2, e_3 tres vectores bases en el espacio euclidiano tridimensional, en el espacio conforme se agregan dos vectores bases más e_0, e_4 que satisfacen las expresiones mostradas en (3.23). En este caso, los cinco vectores son ortogonales.

$$e_0^2 = -1, \quad e_4^2 = +1 \quad (3.23)$$

A partir de los dos nuevos vectores se definen los vectores nulos v y \bar{v} :

$$v = e_4 + e_0, \quad \bar{v} = e_4 - e_0 \quad (3.24)$$

En este punto es necesario escoger un vector para representar el origen. Convencionalmente se toma $-\frac{1}{2}\bar{v}$. Entonces, el vector X se puede escribir como en (3.25), donde x es la representación del punto en el espacio euclidiano y α se elige tal que X sea nulo, lo cual se alcanza cuando $\alpha = x^2$.

$$X = 2x - \bar{v} + \alpha v \quad (3.25)$$

La representación de un punto en el espacio conforme se observa en (3.26).

$$X = 2x + x^2v - \bar{v} \quad (3.26)$$

En esta ecuación v representa el punto al infinito. El poder de las representaciones en un espacio conforme se observa más claramente cuando se considera el producto interior de los puntos. El producto interior en el espacio conforme codifica la distancia entre puntos en el espacio euclidiano (Dorst y Fontijne 2005). Cualquier transformación de dos puntos que deje invariante este producto debe representar una simetría de un espacio euclidiano. Debido a esto, una transformación que deja invariante el producto interior, debe ser una transformación ortogonal en el espacio conforme.

Las transformaciones conformes dejan invariantes los ángulos pero pueden alterar distancias (Doran 2003). Las transformaciones euclidianas son un subconjunto de las transformaciones conformes. En esta geometría, las líneas rectas son círculos de radio infinito. Y en tres dimensiones, los planos y las esferas se tratan como un mismo objeto.

3.4.2 Representación de objetos

Además de ser un modelo muy conveniente para representar la geometría euclidiana (Dorst y Fontijne 2005), el Álgebra Geométrica Conforme es un esquema geométrico libre de coordenadas de 5 dimensiones. Se construye a partir de una representación de puntos e híper-esferas (internamente estos objetos se representan como vectores). Además, las operaciones no dependen de un origen y no necesitan ser especificadas en términos de coordenadas relativas a un origen, de ahí que sea libre de coordenadas.

En esta tesis, el álgebra trabaja en el dominio conforme y la semántica geométrica lo hace en el dominio euclidiano.

El álgebra geométrica conforme es una extensión de la geometría proyectiva de cuatro dimensiones. Los puntos, las esferas y los planos se pueden representar fácilmente como objetos algebraicos (vectores). Por ejemplo, un círculo se puede

representar con la intersección de dos esferas (o una esfera y un plano), la cual se realiza con una operación algebraica básica. Por otro lado, al combinar tres puntos (usando otro producto del álgebra) se obtiene el círculo que pasa por ellos.

Esta álgebra trata a las líneas y a los círculos de una manera unificada como tri-vectores. Las mismas consideraciones se aplican al multi-vector S que se define usando cuatro puntos X^1, X^2, X^3, X^4 . El objeto definido por S es una hiper-esfera única que pasa por los cuatro puntos.

Las intersecciones entre líneas, círculos, hiper-planos e hiper-esferas se generan directamente. En la geometría básica se utilizan los puntos y los vectores mientras que en esta álgebra se tienen una gran variedad de objetos geométricos básicos con los cuales se puede trabajar (Hildenbrand 2005).

Existen dos formas de representar objetos en el álgebra geométrica conforme. La representación *IPNS* (*Inner Product Null Space*) y la representación *OPNS* (*Outer Product Null Space*), ambas representaciones son *duales* una a la otra. Dependiendo de la aplicación se puede utilizar una u otra representación.

Dadas las siguientes bases del Álgebra Geométrica Conforme:

$$e_0, e_1, \dots, e_n, e_\infty, \quad (3.27)$$

donde e_0 representa el origen euclidiano y e_∞ es el punto al infinito. Estas bases cumplen las propiedades (3.28), (3.29), (3.30) y (3.31); donde $e_i^2 = e_i \cdot e_i$ y \cdot es el producto interior.

$$e_i^2 = 1, \forall_{i=1, \dots, n}. \quad (3.28)$$

$$e_\infty^2 = e_0^2 = 0. \quad (3.29)$$

$$e_\infty \cdot e_0 = -1. \quad (3.30)$$

$$e_i \cdot e_\infty = e_i \cdot e_0 = 0, \forall_{i=1, \dots, n}. \quad (3.31)$$

Un vector en álgebra geométrica conforme se denota como una combinación lineal entre las bases:

$$S = S'_1 e_1 + \dots + S'_n e_n + S'_{n+1} e_\infty + S'_{n+2} e_0. \quad (3.32)$$

Si la ecuación (3.32) se multiplica por una constante $\lambda \neq 0$, siempre se obtiene el mismo objeto geométrico. El significado de los valores en las dos nuevas coordenadas e_0 y e_∞ se observa en la Tabla 1.

Tabla 1. Significado del vector conforme S de acuerdo a los valores de las componentes e_0 y e_∞

	$S'_{n+2} = 0$	$S'_{n+2} \neq 0$
$S'_{n+1} = 0$	Plano que pasa por el origen	Híper-esfera/punto que pasa por el origen
$S'_{n+1} \neq 0$	Plano	Híper-esfera/punto

3.4.3 Objetos del Álgebra Geométrica Conforme

En general, dado un punto euclidiano en $p \in \mathbb{R}^n$, este se puede representar en un punto P del álgebra geométrica conforme como (Li, Hestenes y Rockwood 2001):

$$P = p + \frac{1}{2} p^2 e_\infty + e_0, \quad (3.33)$$

donde p^2 es la norma cuadrada o $p^2 = p \cdot p$.

La ecuación (3.33) expresa una relación homogénea entre ambos dominios, el euclidiano y el conforme. Cuando el coeficiente de e_0 es igual a 1, entonces se dice que P tiene una representación canónica.

De la misma forma, una híper-esfera en la representación IPNS toma la siguiente forma canónica (Hitzer 2004):

$$S = C - \frac{1}{2} \gamma^2 e_\infty = c - \frac{1}{2} c^2 - \gamma^2 e_\infty + e_0, \quad (3.34)$$

donde C es el punto central de la hiper-esfera en su representación conforme como en (3.33), y γ es su radio. También se puede representar en OPNS utilizando cuatro puntos que están situados sobre ella (Hitzer 2004):

$$S = X^1 \wedge X^2 \wedge X^3 \wedge X^4. \quad (3.35)$$

Por otra parte, un hiper-plano se define (en notación IPNS) por:

$$T = t + \delta e_\infty, \quad (3.36)$$

donde t es el vector euclidiano normal al hiper-plano T , y δ es la distancia de éste al origen. También se puede definir (en notación OPNS) utilizando los tres puntos que están dentro del hiper-plano y el punto al infinito (Hitzer 2004):

$$T = X^1 \wedge X^2 \wedge X^3 \wedge e_\infty. \quad (3.37)$$

De (3.35) y (3.37) se puede observar que un hiper-plano es una hiper-esfera con radio infinito (Hestenes 2001).

Un círculo se define, al usar la representación IPNS, por la intersección de dos hiper-esferas:

$$Z = S^1 \wedge S^2, \quad (3.38)$$

o, al usar la representación OPNS, con los tres puntos que caen sobre su circunferencia:

$$Z = X^1 \wedge X^2 \wedge X^3. \quad (3.39)$$

De la misma forma, una línea se define, con la representación IPNS, por la intersección de dos hiper-planos:

$$L = T^1 \wedge T^2. \quad (3.40)$$

En la representación OPNS se define usando dos puntos que pasan por ella y el punto al infinito:

$$L = X^1 \wedge X^2 \wedge e_\infty. \quad (3.41)$$

Al igual que con la hiper-esfera y el hiper-plano, se puede observar que una línea es un círculo con radio infinito (Hestenes 2001).

En la Tabla 2 se observan todos los objetos que se pueden utilizar en el álgebra geométrica conforme.

Tabla 2. Lista de entidades del álgebra geométrica conforme

Entidad	Representación IPNS	Representación OPNS
Híper-esfera	$S = P - \frac{1}{2} \gamma^2 e_\infty$	$S = P^1 \wedge P^2 \wedge P^3 \wedge P^4$
Híper-plano	$T = t + \delta e_\infty$	$T = P^1 \wedge P^2 \wedge P^3 \wedge e_\infty$
Círculo	$Z = S^1 \wedge S^2$	$Z = P^1 \wedge P^2 \wedge P^3$
Línea	$L = T^1 \wedge T^2$	$L = P^1 \wedge P^2 \wedge e_\infty$
Punto par	$Pp = S^1 \wedge S^2 \wedge S^3$	$Pp = P^1 \wedge P^2$
Punto	$P = S^1 \wedge S^2 \wedge S^3 \wedge S^4$	$P = p + \frac{1}{2} p^2 e_\infty + e_0$

3.4.4 La distancia y el producto interior

El producto interior de dos vectores de grado uno, es similar al ya conocido producto escalar. El cuadrado del producto interior de los vectores base euclidianos es 1, como se vio en (3.28).

Mediante en producto interior, es posible conocer el ángulo θ entre dos objetos A^1 y A^2 (ya sean dos líneas o dos planos):

$$\cos \theta = \frac{A^1 \cdot A^2}{|A^1| |A^2|}. \quad (3.42)$$

El producto interior de dos objetos en el álgebra geométrica conforme se puede usar como una forma para calcular la distancia. Dados dos puntos en el marco del álgebra geométrica conforme P^1 y P^2 , cuando se aplica el producto interior entre ellos se obtiene la siguiente expresión:

$$P^1 \cdot P^2 = \mathbf{p}^1 \cdot \mathbf{p}^2 - \frac{1}{2} \mathbf{p}^1{}^2 - \frac{1}{2} \mathbf{p}^2{}^2. \quad (3.43)$$

O, lo que es igual:

$$P^1 \cdot P^2 = -\frac{1}{2} \mathbf{p}^1 - \mathbf{p}^2{}^2, \quad (3.44)$$

Una medida de distancia entre un punto conforme P y una hiper-esfera S , se obtiene al usar el producto interior (Hildenbrand 2005), como sigue:

$$P \cdot S = \mathbf{p} \cdot \mathbf{c} - \frac{1}{2} \mathbf{c}{}^2 - \gamma^2 - \frac{1}{2} \mathbf{p}{}^2 = \frac{1}{2} \gamma^2 - \mathbf{c} - \mathbf{p}{}^2. \quad (3.45)$$

En forma simplificada:

$$2 P \cdot S = \gamma^2 - \mathbf{c} - \mathbf{p}{}^2. \quad (3.46)$$

De acuerdo a (3.46):

- Si $P \cdot S > 0$, entonces P está dentro de la hiper-esfera (Figura 8A).
- Si $P \cdot S < 0$, entonces P está fuera de la hiper-esfera (Figura 8B).
- Si $P \cdot S = 0$, entonces P está sobre la superficie de la hiper-esfera (Figura 8C).

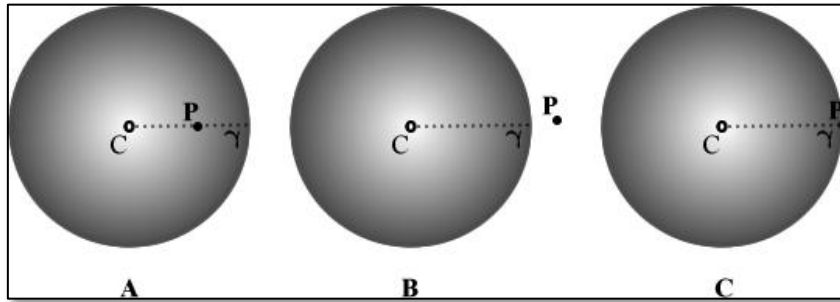


Figura 8. Tres posibles posiciones de un punto con respecto a una hiper-esfera.

Por lo tanto, cuando se utiliza una vecindad esférica, el producto interior hace posible la clasificación de un patrón; ya que permite determinar si el patrón está dentro o fuera de la hiper-esfera.

CAPÍTULO

4. MEMORIAS ASOCIATIVAS GEOMÉTRICAS

En este capítulo se establecen las bases para el desarrollo del nuevo modelo de memorias asociativas, las llamadas memorias asociativas geométricas. Mientras que los primeros modelos de memorias asociativas utilizaban operadores del álgebra convencional para su funcionamiento y las memorias morfológicas basaban su operación en la morfología matemática, el modelo propuesto en esta tesis utiliza, para su operación, operadores del álgebra geométrica; en particular se utiliza el modelo conforme.

Se desarrollaron dos modos de operación de las memorias asociativas geométricas que atacan, precisamente, dos de los problemas más importantes del reconocimiento de patrones: el problema de la clasificación y el problema de la restauración.

Su función, como se verá más adelante, es clasificar un patrón si y solo si el patrón se encuentra dentro de vecindad de soporte (híper-esfera) de esa clase.

4.1 Modo para clasificación de patrones

En las siguientes secciones se explican tres métodos para construir vecindades híper-esféricas de forma no iterativa.

4.1.1 Creación de las vecindades híper-esféricas

La construcción de las vecindades híper-esféricas implica, primero encontrar el centro de cada híper-esfera y después calcular el radio adecuado. En la literatura se han reportado algunos procedimientos para lograr esto con álgebra geométrica conforme. En las siguientes líneas se describen tres de ellos.

El primer método se describe en (Cruz, Barrón y Sossa 2009), donde dado un conjunto de puntos $\mathbf{p} = \mathbf{p}^i$ $_{i=1}^m$ se construye una vecindad híper-esférica. El centro se calcula de una manera muy sencilla:

$$\mathbf{c} = \sum_{i=1}^m \mathbf{p}_i / m. \quad (1.1)$$

En otras palabras, el centro de cada híper-esfera es el promedio de todos los puntos. La expresión (4.2) se utiliza para calcular el radio.

$$-\frac{1}{2} \gamma^2 = \min \left[C \cdot P^i \right], \forall_{i=1, \dots, m}, \quad (1.2)$$

donde C y P^i son la representación conforme de \mathbf{c} y \mathbf{p}^i respectivamente. El procedimiento garantiza que todos los patrones en una clase estarán cubiertos por una híper-esfera. Como el centro y el radio se calculan en diferentes etapas, la híper-esfera resultante podría no estar balanceada, como se observa en la Figura 1A.

Otra desventaja importante de este procedimiento es su alto costo computacional. En (Hildenbrand 2005) se presenta un método para ajustar conjuntos de puntos en híper-planos o híper-esferas usando una aproximación por mínimos cuadrados. En ese trabajo se describe el ajuste de híper-esferas (Figura 1B). Ese algoritmo usa una medida de distancia entre puntos e híper-esferas con ayuda del producto interior. Utiliza una aproximación por mínimos cuadrados para minimizar el cuadrado de la distancia entre un punto y una híper-esfera.

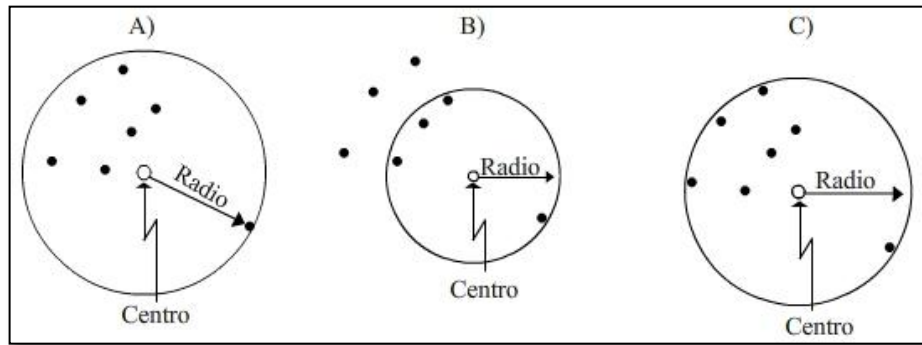


Figura 1. Tres soluciones gráficas para encontrar el centro y el radio de una hiper-esfera.

Con este procedimiento es posible crear vecindades hiper-esféricas que se ajusten a un conjunto de patrones. Un inconveniente que tiene este método es que, algunas veces, algunos puntos pueden aparecer afuera de la resultante; por lo que los patrones usados en el entrenamiento no serán clasificados correctamente.

En (Barrón, Cruz, y otros 2008) se presentó un método para construir de manera óptima una envoltura convexa (del inglés *convex hull*) esférica usando programación cuadrática y álgebra geométrica. Una envoltura convexa es importante porque es una forma más natural de modelar una clase con sus patrones correspondientes y, por lo tanto, clasificarlos de acuerdo a esta clase. El método combina las características de los métodos mencionados en los párrafos previos. El método, además, ajusta una hiper-esfera óptima pero conteniendo todos los puntos (Figura 1C).

Los métodos mencionados se pueden usar para construir una vecindad hiper-esférica para una clase específica usando los puntos (patrones) de esa misma clase. Pero no toman en cuenta los patrones de las otras clases o la separación de las mismas.

En la siguiente sección se presenta un nuevo método para construir vecindades de decisión hiper-esféricas, pero tomando en cuenta los puntos que estarán dentro de la vecindad y los puntos que deben estar afuera de la misma.

4.1.2 Nuevo método para crear vecindades esféricas

Antes de explicar el método se enunciará una nueva definición, partiendo de la **¡Error! No se encuentra el origen de la referencia.** sobre separabilidad lineal:

Definición 1. Cuando dos conjuntos de puntos en \mathbb{R}^n son completamente separados por una hiper-esfera, se dice que estos conjuntos son esféricamente separables.

En este caso la separación consiste en tener un conjunto de puntos *dentro* de la hiper-esfera y el otro conjunto de puntos *afuera* de la hiper-esfera. A partir de esta definición se enunciará el siguiente teorema que tiene que ver con la relación entre la separabilidad esférica y la lineal.

Teorema 1. Si dos conjuntos de puntos en \mathbb{R}^n son linealmente separables, entonces estos conjuntos son también esféricamente separables.

Demostración. Sean dos conjuntos de puntos linealmente separables en \mathbb{R}^n . De la definición 1 existe un hiper-plano T que los separa. Pero de **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** el hiper-plano T es una hiper-esfera que pasa por el punto al infinito. Entonces, también existe una hiper-esfera que los separa, por lo tanto ambos conjuntos son también esféricamente separables ■

Hay que notar que el Teorema 1 no garantiza que dos conjuntos de puntos esféricamente separables sean también linealmente separables.

A continuación se describe una nueva propuesta, inspirada en ideas de (Barrón, Cruz, y otros 2008). Ésta sirve para encontrar una vecindad hiper-esférica que toma en cuenta los patrones de la clase dentro de la hiper-esfera y los patrones de las otras clases.

Sean p^i $_{i=1}^l$ y p^j $_{j=l+1}^m$ dos conjuntos de puntos esféricamente separables en \mathbb{R}^n , tal que p^i son puntos que pertenecen a una clase y p^j son puntos que pertenecen a otra clase. El problema es encontrar una hiper-esfera S óptima (en términos de programación cuadrática) con el menor error cuadrático, donde los puntos p^i estén dentro de S y los de p^j estén afuera. En otras palabras, resolver:

$$\arg \min_S \sum_{i=1}^m P^i \cdot S^2. \quad (1.3)$$

Sujeto a (4.4) para los puntos dentro de la hiper-esfera y a (4.5) para los puntos fuera de ella.

$$P^i \cdot S \geq 0, i = 1, \dots, l. \quad (1.4)$$

$$P^j \cdot S < 0, j = l+1, \dots, m. \quad (1.5)$$

Para poder encontrar una hiper-esfera óptima se debe utilizar un algoritmo de optimización; así que hay que cambiar la expresión (1.3) para ponerla en notación euclidiana y en términos de algún problema clásico de optimización.

Partiendo de (1.3) y considerando que las hiper-esferas están en forma canónica, tales que el término S_{n+2} de las mismas se puede eliminar, se obtienen:

$$\begin{aligned} \sum_{i=1}^m P^i \cdot S^2 &= \sum_{i=1}^m \left(\mathbf{p}^i \cdot \mathbf{c} - S_{n+1} - \frac{1}{2} \mathbf{p}^{i \cdot 2} \right)^2 \\ &= \sum_{i=1}^m \left([\mathbf{p}^i \cdot \mathbf{c} - S_{n+1}] - \frac{1}{2} \mathbf{p}^{i \cdot 2} \right)^2 \\ &= \sum_{i=1}^m \left([\mathbf{p}^i \cdot \mathbf{c} - S_{n+1}]^2 - [\mathbf{p}^i \cdot \mathbf{c} - S_{n+1}] \mathbf{p}^{i \cdot 2} + \frac{1}{4} \mathbf{p}^{i \cdot 4} \right), \end{aligned} \quad (1.6)$$

hay que recordar que $S_{n+1} = \frac{1}{2} \mathbf{p}^{i \cdot 2} - \gamma^2$, siendo γ el radio de la hiper-esfera. Con lo que la expresión (1.3) se puede ver como:

$$\begin{aligned} \sum_{i=1}^m P^i \cdot S^2 &= \sum_{i=1}^m \mathbf{p}^i \cdot \mathbf{c} - S_{n+1}^2 \\ &\quad + \sum_{i=1}^m -\mathbf{p}^i \cdot \mathbf{c} + S_{n+1} \mathbf{p}^{i \cdot 2} + \frac{1}{4} \sum_{i=1}^m \mathbf{p}^{i \cdot 4}. \end{aligned} \quad (1.7)$$

El tercer término de la expresión es irrelevante porque no depende del parámetro S y se puede omitir. Sin pérdida de generalidad esta expresión puede ser escrita en notación matricial euclidiana como en (4.8), donde \mathbf{W} y \mathbf{F} son matrices, cuyas

componentes se observan en (4.9) y en (4.10) respectivamente, y $\mathbf{x} = [S_1, S_2, \dots, S_{n+1}]$. Además, hay que recordar que $S_k = c_k \forall_{k=1, \dots, n}$.

$$\sum_{i=1}^m \left(\sum_{k=1}^{n+1} \mathbf{W}_{i,k} \mathbf{x}_k \right)^2 + \sum_{i=1}^m \left(\sum_{k=1}^{n+1} \mathbf{F}_{i,k} \mathbf{x}_k \right). \quad (1.8)$$

$$\mathbf{W}_{i,k} = \begin{cases} p_k^i & \text{para } k = 1, \dots, n \\ -1 & \text{para } k = n+1 \end{cases} \quad (1.9)$$

$$\mathbf{F}_{i,k} = \begin{cases} -p_k^i & p^{i-2} & \text{para } k = 1, \dots, n \\ p^{i-2} & & \text{para } k = n+1 \end{cases}. \quad (1.10)$$

Sea $\mathbf{w}^i = [\mathbf{W}_{i,1}, \mathbf{W}_{i,2}, \dots, \mathbf{W}_{i,n+1}]$, entonces para el término de la izquierda de la expresión (1.8) y considerando $\mathbf{w}^{i \ t}$ que es el transpuesto del vector \mathbf{w}^i , se tiene:

$$\begin{aligned} \sum_{i=1}^m \left(\sum_{k=1}^{n+1} \mathbf{W}_{i,k} S_k \right)^2 &= \sum_{i=1}^m \mathbf{w}^{i \ t} \mathbf{x}^2 \\ &= \mathbf{w}^{1 \ t} \mathbf{x}^2 + \dots + \mathbf{w}^{m \ t} \mathbf{x}^2 \\ &= \mathbf{w}^{1 \ t} \mathbf{x} \mathbf{w}^{1 \ t} \mathbf{x} + \dots + \mathbf{w}^{m \ t} \mathbf{x} \mathbf{w}^{m \ t} \mathbf{x} \\ &= \mathbf{w}^{1 \ t} \mathbf{x} \mathbf{w}^{1 \ t} + \dots + \mathbf{w}^{m \ t} \mathbf{x} \mathbf{w}^{m \ t} \mathbf{x} \\ &= [\mathbf{x}^t \mathbf{w}^1 + \dots + \mathbf{x}^t \mathbf{w}^m] \mathbf{W} \mathbf{x} \\ &= \mathbf{x}^t \mathbf{W}^t \mathbf{W} \mathbf{x}. \end{aligned} \quad (1.11)$$

Ahora, sea $\mathbf{H} = \mathbf{W}^t \mathbf{W}$ entonces se obtiene la siguiente expresión:

$$\sum_{i=1}^m \left(\sum_{k=1}^{n+1} \mathbf{W}_{i,k} S_k \right)^2 = \mathbf{x}^t \mathbf{H} \mathbf{x}. \quad (1.12)$$

Para el término de la derecha de la expresión (1.8), sea $\mathbf{y}_k = \sum_{i=1}^m \mathbf{F}_{i,k}$ entonces:

$$\begin{aligned} \sum_{i=1}^m \left(\sum_{k=1}^{n+1} \mathbf{F}_{i,k} \mathbf{x}_k \right) &= \sum_{k=1}^{n+1} \left(\sum_{i=1}^m \mathbf{F}_{i,k} \mathbf{x}_k \right) \\ &= \sum_{k=1}^{n+1} \mathbf{y}_k \mathbf{x}_k . \end{aligned} \quad (1.13)$$

Si $\mathbf{y} = [\mathbf{y}_1, \dots, \mathbf{y}_{n+1}]$ se obtiene la siguiente expresión:

$$\sum_{i=1}^m \left(\sum_{k=1}^{n+1} \mathbf{F}_{i,k} \mathbf{S}_k \right) = \mathbf{y}^t \mathbf{x}. \quad (1.14)$$

Al usar las expresiones (1.12) y (1.14), la expresión (1.3) se ha convertido a una notación matricial euclidiana:

$$\mathbf{x}^t \mathbf{H} \mathbf{x} + \mathbf{y}^t \mathbf{x}. \quad (1.15)$$

La restricción (1.4) para puntos dentro de la hiper-esfera se modifica de la siguiente forma:

$$\begin{aligned} P^i \cdot S &\geq 0 \\ \mathbf{p}^i \cdot \mathbf{c} - S_{n+1} - \frac{1}{2} \mathbf{p}^i \cdot \mathbf{p}^i &\geq 0 \\ \mathbf{p}^i \cdot \mathbf{c} - S_{n+1} &\geq \frac{1}{2} \mathbf{p}^i \cdot \mathbf{p}^i \\ \sum_{k=1}^{n+1} -\mathbf{W}_{i,k} S_k &\leq -\frac{1}{2} \mathbf{p}^i \cdot \mathbf{p}^i , \end{aligned} \quad (1.16)$$

donde $\mathbf{W}_{i,k}$ se definió en (1.9), la expresión (1.16) se puede escribir como:

$$-\mathbf{W} \mathbf{x}^t \leq -\frac{1}{2} \mathbf{p}^i \cdot \mathbf{p}^i , \quad (1.17)$$

donde \mathbf{x} se definió anteriormente. La restricción (1.5) para los puntos afuera de la hiper-esfera se modifica a:

$$\begin{aligned}
p^j \cdot S &< 0 \\
p^j \cdot c - S_{n+1} - \frac{1}{2} p^j{}^2 &< 0 \\
p^j \cdot c - S_{n+1} &< \frac{1}{2} p^j{}^2 \\
\sum_{k=1}^{n+1} \mathbf{W}_{j,k} S_k &< \frac{1}{2} p^j{}^2.
\end{aligned} \tag{1.18}$$

Esta expresión puede re-escribirse como:

$$\mathbf{W}x^t < \frac{1}{2} p^j{}^2. \tag{1.19}$$

Sea $\mathbf{A} = -\mathbf{W}, \mathbf{W}$ y sea \mathbf{b} un vector cuya i -ésima componente es $-\frac{1}{2} p^i{}^2$ para $i=1, \dots, l$ y $\frac{1}{2} p^i{}^2 - \varepsilon$ para $i=l+1, \dots, m$, la constante ε es una cantidad positiva muy pequeña que se utiliza para cambiar el “<” de la desigualdad (1.19) a un “≤”. Las restricciones (1.17) y (1.19) son entonces convertidas a una notación matricial euclidiana del tipo:

$$\mathbf{A}x \leq \mathbf{b}. \tag{1.20}$$

Finalmente, la expresión (1.3) se cambia a un problema clásico de optimización con restricciones:

$$\begin{aligned}
\min_x \quad & x^t \mathbf{H}x + y^t x \\
\text{s.a.} \quad & \mathbf{A}x \leq \mathbf{b}.
\end{aligned} \tag{1.21}$$

La hiper-esfera óptima S se encuentra al resolver la expresión (1.21), donde $S_k = x_k$ para $k=1, \dots, n+1$ y $S_{n+2} = 1$. Está claro que al incluir también en las restricciones los puntos que se encuentran fuera de la hiper-esfera, la solución S es una superficie de

separación que permite diferenciar entre dos clases (puntos dentro y fuera de la hiper-esfera).

Este procedimiento funciona muy bien cuando las clases son esféricamente separables. En caso de tener tres o más clases el procedimiento es similar. En este caso, el conjunto de puntos \mathbf{p}^i $_{i=1}^l$ serán todos los patrones de la k -ésima clase, y el conjunto de puntos \mathbf{p}^j $_{j=l+1}^m$ serán todos los patrones de las demás clases. La k -ésima hiper-esfera S^k se encuentra al resolver la expresión (1.21). El mismo procedimiento se debe realizar para las otras clases.

4.1.3 Fase de aprendizaje

La fase de aprendizaje de una memoria asociativa consiste en almacenar las asociaciones entre los patrones de entrada y sus correspondientes patrones de salida. En el caso de las memorias asociativas geométricas, la fase de aprendizaje consiste en crear las vecindades esféricas para cada clase.

Una memoria asociativa geométrica es, por lo tanto, una matriz \mathbf{M} de tamaño $q \times n + 2$, donde q es el número total de clases. El k -ésimo renglón de \mathbf{M} es la k -ésima hiper-esfera. \mathbf{M} se puede observar en la Figura 2, donde C^k y γ^k son, respectivamente, el centro y el radio de la k -ésima hiper-esfera.

Las hiper-esferas se pueden crear usando alguno de los procedimientos mencionados en la sección 4.1, aunque se recomienda utilizar el procedimiento descrito en la sección 4.1.1.

4.1.4 Fase de clasificación

Por otro lado, para la fase de clasificación de una memoria asociativa geométrica se realizará usando la idea principal, presentada originalmente en (Banarer, Perwass y Sommer 2003), pero con ciertos cambios para considerar el caso de patrones que se encuentren dentro de dos hiper-esferas.

$$\mathbf{M} = \begin{bmatrix} S^1 \\ S^2 \\ \vdots \\ S^q \end{bmatrix} = \begin{bmatrix} S^1 = C^1 - \frac{1}{2}(\gamma^1)^2 e_\infty \\ S^2 = C^2 - \frac{1}{2}(\gamma^2)^2 e_\infty \\ \vdots \\ S^q = C^q - \frac{1}{2}(\gamma^q)^2 e_\infty \end{bmatrix}$$

Figura 2. Memoria asociativa geométrica después de la fase de aprendizaje.

Primero se debe aplicar un producto interior entre un vector (fundamental o de prueba) x y la propia memoria asociativa \mathbf{M} para obtener un vector u de tamaño m . Dicho vector contiene todos los productos interiores entre el patrón x y todas las hiper-esferas. La k -ésima componente de u está dada por:

$$u_k = \mathbf{M}_k \cdot X = S^k \cdot X. \quad (1.22)$$

En este caso X es la representación conforme del patrón x . Si x está dentro de una hiper-esfera, la ecuación (1.22) regresa un valor positivo o cero en la posición de esa hiper-esfera y un valor negativo cuando se encuentra fuera de la hiper-esfera. En algunos casos (principalmente cuando se trata de patrones afectados con ruido) x podría estar dentro de dos o más hiper-esferas o podría estar fuera de todas las hiper-esferas. Para decidir a cuál hiper-esfera pertenece un patrón dado, se debe utilizar el siguiente re-mapeo:

$$v_k = \begin{cases} -\infty; & \text{si } u_k < 0 \\ u_k - \gamma^k{}^2; & \text{otro caso} \end{cases}. \quad (1.23)$$

Este re-mapeo se debe realizar para $k = 1, \dots, q$. Finalmente, el identificador de clase se obtiene aplicando una función de argumento máximo:

$$j = \arg \max_k v_k \quad k=1, \dots, q. \quad (1.24)$$

Como se puede observar, cuando X está afuera de la k -ésima hiper-esfera, la expresión (1.23) regresa $-\infty$ y cuando X está dentro de la k -ésima hiper-esfera, regresa la distancia (con signo menos) entre el vector X y el centro de la k -ésima hiper-esfera C^k . Por tanto, el patrón x será clasificado por una hiper-esfera cubriendo su representación conforme y con ayuda de la expresión (1.24) éste será clasificado por la hiper-esfera con centro más cercano a x .

En algunos casos $v_k = -\infty \forall_{k=1, \dots, q}$, esto es, X está afuera de todas las hiper-esferas. Cuando se aplique la expresión (1.24), ésta no puede regresar algún valor. En este punto, se pueden tener dos situaciones. La primera es que x no pertenece a ninguna clase. La segunda opción es usar la expresión (1.24) directamente sobre $u_k - \gamma^k$, aquí la memoria se comportaría como un clasificador de distancia mínima, pero no se tomaría en cuenta el uso de las vecindades esféricas.

Hay que apuntar que la fase de clasificación no depende de la fase de aprendizaje.

4.1.5 Condición para clasificación perfecta

Por la **¡Error! No se encuentra el origen de la referencia.**, cuando una memoria asociativa clasifica correctamente el conjunto fundamental, entonces la memoria presenta clasificación perfecta. Sea \mathbf{M} una memoria asociativa geométrica entrenada con el método descrito en la página 3.

Teorema 2. *Sean q clases esféricamente separables en \mathbb{R}^n y sea \mathbf{M} una memoria asociativa geométrica entrenada para esas clases, entonces \mathbf{M} presenta recuperación perfecta.*

Demostración. *Sea k un índice de clase cuya hiper-esfera S^k es la k -ésima componente de \mathbf{M} y sea \mathbf{p} un patrón fundamental de la clase k , y sea j un índice $j=1, \dots, q$ tal que $j \neq k$. La hiper-esfera S^k se obtuvo por medio de la expresión (1.21). Entonces, de acuerdo a las condiciones (1.4) y (1.5), $\mathbf{p} \cdot S^k \geq 0$ porque \mathbf{p} es un patrón de clase k , y $\mathbf{p} \cdot S^j < 0$. Cuando se aplica la expresión (1.23), el vector \mathbf{u} tiene un número positivo o cero en la posición k y $-\infty$ en las otras posiciones. Por lo tanto la expresión (1.24) regresa k . Lo mismo sucede con todos los demás patrones en todas las demás clases ■.*

4.1.6 Condición para clasificación robusta

De acuerdo a la ~~¡Error! No se encuentra el origen de la referencia.~~, una memoria asociativa presenta clasificación robusta cuando ésta clasifica patrones afectados con ruido.

La robustez de una memoria asociativa geométrica depende en gran medida del tamaño de su radio; la memoria puede clasificar cualquier patrón como perteneciente a su clase cuando el patrón se encuentre dentro de la hiper-esfera. Los patrones localizados fuera de una hiper-esfera específica no serán clasificados como pertenecientes a dicha hiper-esfera.

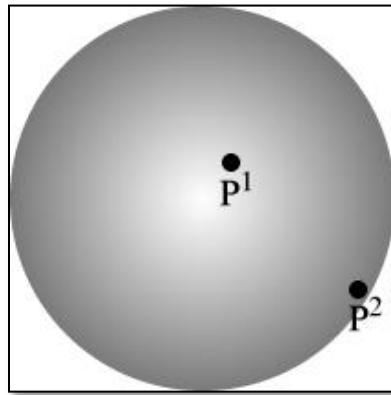


Figura 3. Ejemplo de dos patrones en \mathbb{R}^3 dentro de una hiper-esfera, el patrón P^2 , más cercano al borde, acepta menos cantidad de ruido que P^1 , que está más cercano al centro.

La cantidad de ruido que puede admitir un patrón fundamental depende de la posición del mismo con respecto al centro y al borde de la hiper-esfera. Los patrones que se encuentran más cerca del centro pueden admitir más cantidad de ruido que aquellos localizados más cerca del borde (Figura 3).

Teorema 3. Sea $\tilde{\mathbf{p}}$ un patrón fundamental afectado con ruido, tal que $\tilde{\mathbf{p}} = \mathbf{p} + \mathbf{r}$, donde \mathbf{r} es el ruido y sea \mathbf{M} una memoria asociativa geométrica donde la hiper-esfera S^i es la hiper-esfera de la clase a la cual pertenece \mathbf{p} y m es el número de clases. Si se cumple la expresión (4.25) para algún $i = 1, \dots, m$, entonces $\tilde{\mathbf{p}}$ será clasificado por \mathbf{M} a la clase más cercana a él.

$$S^i \cdot \tilde{P} \geq -\frac{\gamma^i{}^2}{2}. \quad (1.25)$$

Demostración. *Mediante la expresión ¡Error! No se encuentra el origen de la referencia., se obtiene la expresión (4.26), al obtener el producto interior entre una la esfera y el patrón afectado con ruido.*

$$S^i \cdot \tilde{P} = \frac{1}{2} \gamma^i{}^2 - \mathbf{c}^i - \tilde{\mathbf{p}}{}^2 \quad (1.26)$$

Mediante esta expresión y la expresión (1.25) se obtiene el siguiente desarrollo:

$$\begin{aligned} -\frac{1}{2} \mathbf{c}^i - \tilde{\mathbf{p}}{}^2 &\geq -\frac{\gamma^i{}^2}{2} \\ -\mathbf{c}^i - \tilde{\mathbf{p}}{}^2 &\geq -\gamma^i{}^2 \\ \mathbf{c}^i - \tilde{\mathbf{p}}{}^2 &\leq \gamma^i{}^2 \\ 0 &\leq \gamma^i{}^2 - \mathbf{c}^i - \tilde{\mathbf{p}}{}^2 \\ \gamma^i{}^2 - \mathbf{c}^i - \tilde{\mathbf{p}}{}^2 &\geq 0 \\ 2 S^i \cdot \tilde{P} &\geq 0 \\ S^i \cdot \tilde{P} &\geq 0 \end{aligned} \quad (1.27)$$

Por lo tanto, $\tilde{\mathbf{p}}$ estará dentro de alguna esfera S^i y mediante la expresión (1.24), se garantiza que dicho patrón se encuentre dentro de esfera con centro más cercano ■.

Note que el Teorema 3 no depende del tipo de ruido que afecta al patrón.

4.2 Modo para restauración de patrones

A continuación se presenta un nuevo modo de operación de las memorias asociativas geométricas, basado en las ideas mostradas en la sección 4.1. En este caso se pretende resolver el problema de la restauración de patrones. La idea general es asignar a cada patrón del conjunto fundamental una vecindad esférica, esa funciona como una zona

de encendido y se pueden restaurar patrones dentro de esa vecindad. El patrón recuperado será, precisamente, el centro de la hiper-esfera.

La principal ventaja de este modelo es su capacidad de restaurar, de forma directa, patrones afectados con ruido mezclado. Además, la cantidad de ruido que un patrón puede admitir se encuentra dada por el radio de la vecindad esférica.

A continuación se presenta una nueva forma de construir las memorias asociativas geométricas para que puedan restaurar patrones afectados con ruido.

4.2.1 Creación de las vecindades esféricas

Al igual que en el modo de clasificación, la construcción de las vecindades esféricas puede hacerse, de forma sencilla, tomando el patrón mismo como centro de la hiper-esfera y el radio se calcula por medio de la distancia entre este patrón y los demás. De esta forma se garantiza que no habrá solapamiento entre las hiper-esferas. Sin embargo, al igual que con el modo de clasificación, este procedimiento es computacionalmente costoso, así que de la misma que en la sección 4.1.1 se utilizó una aproximación por mínimos cuadrados para encontrar una hiper-esfera óptima.

Sea $\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^m$ un conjunto fundamental de patrones en \mathbb{R}^n , el problema es encontrar una hiper-esfera óptima S con el mínimo error cuadrático, tal que \mathbf{p}^k se encuentra dentro de S y $\mathbf{p}^j \mid j=1, \dots, m, j \neq k$ se encuentran fuera de ella. La solución se puede encontrar aplicando, de nuevo, la expresión (1.3) con sus respectivas restricciones. Podría pensarse que este problema podría resolverse de la misma manera que el modo de clasificación, mediante la expresión (1.21); sin embargo si se utiliza esta expresión como tal, el centro de la hiper-esfera podría no ser el patrón \mathbf{p}^k . La idea, como se mencionó anteriormente, es regresar, como patrón restaurado, el centro de la hiper-esfera y con este procedimiento, esto no sería posible.

La solución es agregar una nueva restricción a la expresión (1.3), en la cual el centro de la hiper-esfera es igual a \mathbf{p}^k :

$$P^k = C \Leftrightarrow \mathbf{p}^k = \mathbf{c}, \quad (1.28)$$

donde C es el centro de la hiper-esfera y P^k es el patrón a entrenar, ambos en su representación conforme. Entonces, partiendo de la expresión (1.7) (omitiendo el tercer término de la expresión sin pérdida de generalidad) y tomando en cuenta la restricción (1.28) se obtiene:

$$\begin{aligned}
\sum_{i=1}^m P^i \cdot S^2 &= \sum_{i=1}^m p^i \cdot p^k - S_{n+1}^2 + \sum_{i=1}^m -p^i \cdot p^k + S_{n+1} p^{i^2} \\
&= \sum_{i=1}^m p^i \cdot p^k^2 - 2 p^i \cdot p^k S_{n+1} + S_{n+1}^2 \\
&\quad - \sum_{i=1}^m p^i \cdot p^k - S_{n+1} p^{i^2} \\
&= \sum_{i=1}^m p^i \cdot p^k^2 - \sum_{i=1}^m 2 p^i \cdot p^k S_{n+1} + \sum_{i=1}^m S_{n+1}^2 \\
&\quad - \sum_{i=1}^m p^i \cdot p^k p^{i^2} + \sum_{i=1}^m S_{n+1} p^{i^2}.
\end{aligned} \tag{1.29}$$

En este caso el primer y el cuarto término de esta expresión se pueden omitir, debido a que no dependen de S . De esta forma se obtiene el siguiente desarrollo:

$$\begin{aligned}
\sum_{i=1}^m P^i \cdot S^2 &= -\sum_{i=1}^m 2 p^i \cdot p^k S_{n+1} + \sum_{i=1}^m S_{n+1}^2 \\
&\quad + \sum_{i=1}^m S_{n+1} p^{i^2}, \\
&= -2S_{n+1} \sum_{i=1}^m p^i \cdot p^k + m S_{n+1}^2 + S_{n+1} \sum_{i=1}^m p^{i^2}, \\
&= S_{n+1} \left(2 \sum_{i=1}^m p^i \cdot p^k + \sum_{i=1}^m p^{i^2} \right) + m S_{n+1}^2, \\
&= m S_{n+1}^2 + S_{n+1} \sum_{i=1}^m 2 p^i \cdot p^k + p^{i^2}.
\end{aligned} \tag{1.30}$$

Finalmente, sean $\mathbf{y} = \left(\sum_{i=1}^m 2 \mathbf{p}^i \cdot \mathbf{p}^k + \mathbf{p}^i \cdot \mathbf{p}^i \right)$ y $x = S_{n+1}$. Al igual que en la sección 4.1.1, se tiene una expresión en notación matricial euclidiana como en la siguiente expresión:

$$\min_x \mathbf{x} \mathbf{m} \mathbf{x} + \mathbf{y} \mathbf{x} . \quad (1.31)$$

A diferencia del modelo de clasificación, en este caso la variable a optimizar es, simplemente S_{n+1} . La condición para puntos dentro de la hiper-esfera es:

$$\begin{aligned} P^i \cdot S &\geq 0 \\ \mathbf{p}^i \cdot \mathbf{c} - S_{n+1} - \frac{1}{2} \mathbf{p}^i \cdot \mathbf{p}^i &\geq 0 \\ \mathbf{p}^i \cdot \mathbf{p}^i - S_{n+1} - \frac{1}{2} \mathbf{p}^i \cdot \mathbf{p}^i &\geq 0 \\ -S_{n+1} + \frac{1}{2} \mathbf{p}^i \cdot \mathbf{p}^i &\geq 0 \\ -S_{n+1} &\geq -\frac{1}{2} \mathbf{p}^i \cdot \mathbf{p}^i \\ S_{n+1} &\leq \frac{1}{2} \mathbf{p}^i \cdot \mathbf{p}^i . \end{aligned} \quad (1.32)$$

Por su parte, la condición para los puntos fuera de la hiper-esfera es:

$$\begin{aligned} P^j \cdot S &< 0 \\ \mathbf{p}^j \cdot \mathbf{c} - S_{n+1} - \frac{1}{2} \mathbf{p}^j \cdot \mathbf{p}^j &< 0 \\ \mathbf{p}^j \cdot \mathbf{p}^i - S_{n+1} &< \frac{1}{2} \mathbf{p}^i \cdot \mathbf{p}^i \\ -S_{n+1} &< \frac{1}{2} \mathbf{p}^j \cdot \mathbf{p}^j - \mathbf{p}^j \cdot \mathbf{p}^i \\ -S_{n+1} &\leq \frac{1}{2} \mathbf{p}^j \cdot \mathbf{p}^j - \mathbf{p}^j \cdot \mathbf{p}^i + \varepsilon . \end{aligned} \quad (1.33)$$

De la misma forma que el modelo de clasificación, las restricciones (1.32) y (1.33) se pueden convertir a una notación matricial euclidiana, como en la expresión (1.20) y

obtener, finalmente, una expresión similar a (1.21); sin embargo, al ser solamente una variable (S_{n+1}) a optimizar, ésta se puede obtener directamente al calcular la primera derivada de (1.31), pero tomando en cuenta el intervalo entre las dos restricciones.

La hiper-esfera óptima S se encuentra al resolver la expresión (1.21), solo hay que despejar el valor del radio (γ) de S_{n+1} . El centro de la hiper-esfera será el propio punto p^i . En este caso, la solución S es una superficie de separación que permite separar un punto en particular, de todos los demás.

4.2.2 Fase de aprendizaje

Como en el modo de clasificación, la fase de aprendizaje de una memoria asociativa consiste en almacenar las asociaciones entre los patrones de entrada y sus correspondientes patrones de salida. En este caso, la fase de aprendizaje consiste en crear las vecindades esféricas para cada patrón.

Al igual que en el modo de clasificación, la memoria asociativa geométrica es una matriz \mathbf{M} de tamaño $q \times n+2$, donde q es el número de patrones. El k -ésimo renglón de \mathbf{M} es la k -ésima hiper-esfera. \mathbf{M} se puede observar en la Figura 2.

Las hiper-esferas se pueden construir al usar el procedimiento descrito en la sección 4.1.2. Este mismo procedimiento debe realizarse para todos los patrones en el conjunto fundamental.

4.2.3 Fase de restauración

La restauración de un patrón se puede realizar con los mismos pasos que en la Fase de clasificación del modo de clasificación, mostrados en la sección 4.1.4, excepto el último paso, mientras que en el modelo de clasificación se regresa un índice, en el modelo de restauración se regresa el centro de la j -ésima hiper-esfera, como el patrón restaurado, donde j es el valor obtenido por medio de la expresión (1.24).

Al igual que en el modelo de clasificación, un patrón afectado con ruido será restaurado por la hiper-esfera que cubra su representación conforme y con centro más cercano a dicho patrón. En este modelo, las hiper-esferas funcionan como “atractores” del patrón a restaurar.

De la misma forma que en el modelo de clasificación, la fase de restauración de este modo no depende de la fase de aprendizaje.

4.2.4 Condición para restauración perfecta

La condición para restauración perfecta (**¡Error! No se encuentra el origen de la referencia.**) de una memoria asociativa geométrica se enuncia en el Teorema 4. Sea

\mathbf{p}^i un conjunto fundamental de patrones en \mathbb{R}^n y sea \mathbf{M} su memoria asociativa geométrica construida como se mostró en las secciones 4.2.1 y 4.2.2.

Teorema 4. *\mathbf{M} presenta restauración perfecta.*

Demostración. *Sea S^i la hiper-esfera correspondiente al i -ésimo patrón (\mathbf{p}^i). S^i se obtuvo usando la expresión (1.3) y siguiendo el procedimiento de la sección 4.2.1, por lo tanto se cumplen las condiciones (1.4) y (1.5). Al aplicar la expresión (1.23) el vector \mathbf{u} tiene un número positivo en la posición i y $-\infty$ en las demás posiciones. Entonces, si se aplica la expresión (1.24), ésta regresa i . Finalmente el patrón restaurado es, por lo tanto es el centro de la hiper-esfera S^i , es decir \mathbf{p}^i ■.*

4.2.5 Condición para restauración robusta

Al igual que en el modelo anterior, la robustez (**¡Error! No se encuentra el origen de la referencia.**) de una memoria asociativa geométrica depende del tamaño de su radio. La cantidad de ruido que esta memoria puede admitir se enuncia en el Teorema

5. Sea \mathbf{p}^i un conjunto fundamental de patrones en \mathbb{R}^n y sea \mathbf{M} su memoria asociativa geométrica construida como se mostró en las secciones 4.2.1 y 4.2.2. Sea, además, $\tilde{\mathbf{p}}^i = \mathbf{p}^i + \mathbf{r}$ un patrón fundamental afectado con ruido.

Teorema 5. *Si se cumple la expresión (4.34), donde γ^i es el radio de la i -ésima hiper-esfera. El patrón $\tilde{\mathbf{p}}^i$ puede ser restaurado por \mathbf{M} .*

$$r^2 \leq \gamma^i{}^2 \tag{1.34}$$

Demostración. Sea S^i la hiper-esfera del patrón i -ésimo, obtenida mediante el método descrito en la sección 4.2.1. El ruido del patrón $\tilde{\mathbf{p}}^i$ se puede expresar de la siguiente manera:

$$\begin{aligned}\tilde{\mathbf{p}}^i &= \mathbf{p}^i + \mathbf{r} \\ \mathbf{r} &= \tilde{\mathbf{p}}^i - \mathbf{p}^i \\ -\mathbf{r} &= \mathbf{p}^i - \tilde{\mathbf{p}}^i \\ r^2 &= \|\mathbf{p}^i - \tilde{\mathbf{p}}^i\|^2.\end{aligned}\tag{1.35}$$

Si se aplica el producto interior entre la hiper-esfera S^i y la representación conforme de $\tilde{\mathbf{p}}^i$, por la ecuación **¡Error! No se encuentra el origen de la referencia.**, se obtienen las expresiones (4.36).

$$\begin{aligned}2 \tilde{\mathbf{P}} \cdot S^i &= \gamma^2 - \|\mathbf{c}^i - \tilde{\mathbf{p}}^i\|^2 \\ &= \gamma^2 - \|\mathbf{p}^i - \tilde{\mathbf{p}}^i\|^2 \\ &= \gamma^2 - r^2.\end{aligned}\tag{1.36}$$

Entonces:

- Si $r^2 > \gamma^2$ entonces $\tilde{\mathbf{P}}^i \cdot S^i < 0$.
- Si $r^2 \leq \gamma^2$ entonces $\tilde{\mathbf{P}}^i \cdot S^i \geq 0$.

Por lo tanto, el patrón $\tilde{\mathbf{p}}^i$ será restaurado por \mathbf{M} si y sólo si $r^2 \leq \gamma^2$ ■.

Hay que recalcar que el Teorema 5 no depende del tipo de ruido utilizado.

4.3 Modo para clasificación no supervisada

Originalmente las memorias asociativas geométricas para clasificación de patrones fueron desarrolladas para funcionar de forma supervisada. Sin embargo, en esta sección se mostrará un algoritmo de clasificación no supervisada para estas memorias.

Se presentará una variante del famoso algoritmo *k-medias* (del inglés *k-means*) (McQueen 1967), pero usando operadores del álgebra geométrica conforme para construir los *clusters* (agrupamientos) respectivos. Además, al igual que en los modos

anteriores de operación, se puede utilizar el producto interior para saber si un patrón dado se encuentra dentro o fuera de alguno de los agrupamientos. También con el producto interior, es muy fácil conocer la distancia entre dos puntos.

4.3.1 Clasificación no supervisada con álgebra geométrica

El algoritmo propuesto se basa en las ideas mencionadas en las líneas anteriores, pero se usa en el marco de trabajo del álgebra geométrica conforme. A diferencia del algoritmo k -medias, los agrupamientos se encontrarán separando los puntos asociados a un centro específico de los otros puntos al resolver el problema de optimización mostrado en la sección 4.1.1; es decir, al usar una memoria asociativa geométrica. La principal ventaja de este procedimiento es que los nuevos centros se calculan automáticamente.

El método de la sección 4.1.1 toma en cuenta los puntos dentro y fuera de una hiper-esfera de clasificación para un mejor rendimiento. Al igual que en el modo de restauración, las hiper-esferas en la clasificación no supervisada funcionan como atractores para los puntos que están dentro y como contractores para los puntos que están afuera.

Otra diferencia es el cálculo de la distancia entre los centros y los patrones correspondientes. Este procedimiento se realiza con la memoria asociativa geométrica, formada por los centros correspondientes, y con un producto interior.

Durante la primera iteración del algoritmo, los centros se generan de manera aleatoria, como en el algoritmo de k -medias; en las siguientes iteraciones los centros se calculan automáticamente al construir las hiper-esferas correspondientes.

4.3.2 Nuevo algoritmo de agrupamiento

Básicamente el algoritmo es como sigue. Dado un conjunto de puntos \mathbf{p}^i en \mathbb{R}^n , y sea k el número de agrupaciones a generar para esos puntos:

1. Cambiar cada punto \mathbf{p}^i su respectiva representación conforme P^i por medio de la expresión **¡Error! No se encuentra el origen de la referencia..**

2. Establecer k puntos aleatorios, en su representación conforme: C^j donde $j = 1, \dots, k$.
3. Construir la memoria $M = [C^1, \dots, C^k]^t$.
4. Generar k agrupaciones asociando cada punto p^i con el centro más cercano. Este procedimiento se puede realizar aplicando:

$$\arg \max \mathbf{M} \cdot P^i \quad (1.37)$$

5. Encontrar k superficies esféricas S^k por cada agrupación, por medio de la expresión (1.3) y sus respectivas restricciones.
6. Los nuevos valores de los k centros serán, precisamente, los centros de cada hiper-esfera.
7. Repetir desde el paso 3 hasta que los centros no cambien o se llegue a un número determinado de iteraciones.

Como se puede observar, en este algoritmo el cálculo de las agrupaciones se realiza por medio de un simple producto interior y una función de máximo; a diferencia del método tradicional que, para encontrar el centro más cercano utiliza un algoritmo del vecino más cercano.

Con este algoritmo se puede presentar el caso, en algunas situaciones, de que algunos puntos se encuentren fuera de todas las hiper-esferas. Esto, puede ser una ventaja como una desventaja, dependiendo de las características del problema. En algunas ocasiones, estos puntos podrían ser considerados como ruido y se pueden eliminar sin perder información esencial sobre la naturaleza del problema. Sin embargo, para algunas otras situaciones de agrupación estos puntos pueden ser esenciales para resolver el problema.

CAPÍTULO

5. EXPERIMENTACIÓN

En este capítulo se presentan los resultados obtenidos al probar el funcionamiento de las memorias asociativas geométricas propuestas en esta tesis. Por cada modelo desarrollado se presentan dos tipos de problemas, uno numérico y otro con patrones reales. Por simplicidad, y para clarificar los resultados, los problemas numéricos se presentan en dos y tres dimensiones.

Para resolver el problema de optimización, en todos los casos se utiliza la función “*quadprog*” del *toolbox* de optimización del programa MatLab. Esta función resuelve problemas de programación cuadrática. Permite encontrar una solución viable al resolver el problema como uno de programación lineal.

También, para clarificar los resultados, se decidió ajustar el valor de ε a 1.0^{-10} para todos los experimentos; sin embargo, los resultados se muestran con dos centésimas de precisión.

5.1 Experimentación para el modelo de clasificación

5.1.1 Ejemplos numéricos

Ejemplo 1. Sean las siguientes clases linealmente separables en \mathbb{R}^2 :

$$\begin{aligned}
\text{Clase 1} \quad & x^1 = 1 \ 1, \quad x^2 = 2 \ 1, \quad x^3 = 3 \ 2, \quad x^4 = 2 \ 2, \\
\text{Clase 2} \quad & x^5 = 2 \ -1, \quad x^6 = 1 \ -3, \quad x^7 = 3 \ -1, \quad x^8 = 2 \ -2, \\
\text{Clase 3} \quad & x^9 = -1 \ 3, \quad x^{10} = -3 \ 2, \quad x^{11} = -2 \ 2, \quad x^{12} = -4 \ 1, \\
\text{Clase 4} \quad & x^{13} = -2 \ -2, \quad x^{14} = -1 \ -3, \quad x^{15} = -1 \ -1, \quad x^{16} = -3 \ -1.
\end{aligned} \tag{0.1}$$

La siguiente figura muestra una representación gráfica de estos patrones en el plano:

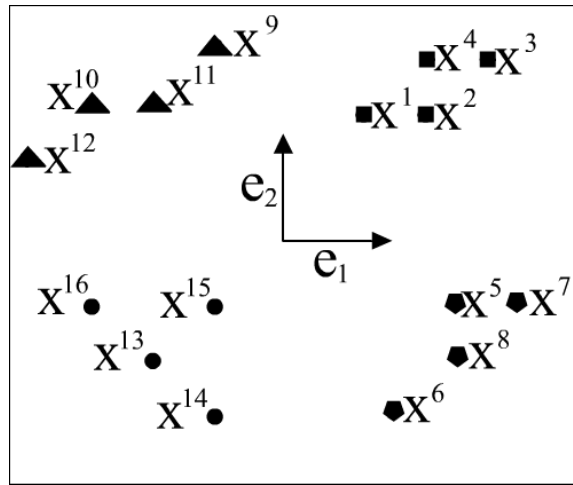


Figura 1. Representación gráfica de las clases del Ejemplo 1. Los cuadrados, pentágonos, círculos y triángulos son puntos pertenecientes a las clases 1, 2, 3 y 4, respectivamente.

Al usar la expresión **¡Error! No se encuentra el origen de la referencia.**, por cada clase, se obtienen las hiper-esferas respectivas. Ya que se está trabajando en un espacio de dos dimensiones las hiper-esferas, en este ejemplo, son, en realidad, círculos. Los centros y radios son, respectivamente:

$$c^1 = 0.65 \ 1.11, \quad \gamma^1 = 2.51.$$

$$c^2 = 1 \ -1, \quad \gamma^2 = 2.$$

$$c^3 = -1.5 \ 1.5, \quad \gamma^3 = 2.55.$$

$$c^4 = -0.8 \ -0.6, \quad \gamma^4 = 2.41.$$

Finalmente, la memoria es:

$$\mathbf{M} = \begin{bmatrix} S^1 = C^1 - \frac{1}{2} \gamma^{1^2} e_\infty \\ \cdot \\ \cdot \\ S^4 = C^4 - \frac{1}{2} \gamma^{4^2} e_\infty \end{bmatrix} = \begin{bmatrix} S^1 = 0.65e_1 + 1.11e_2 - 2.31e_\infty + e_0 \\ S^2 = e_1 - e_2 - e_\infty + e_0 \\ S^3 = -1.5e_1 + 1.5e_2 - e_\infty + e_0 \\ S^4 = -0.8e_1 - 0.6e_2 - 2.4e_\infty + e_0 \end{bmatrix}. \quad (0.2)$$

En la Figura 2 se presentan las hiper-esferas (en este caso círculos) de cada clase. Se puede observar que las hiper-esferas son óptimas, porque si alguna de ellas crece un poco más, algunos puntos de otras clases estarían dentro. Por el contrario, si alguna de ellas decrece un poco, algunos puntos se quedarían fuera.

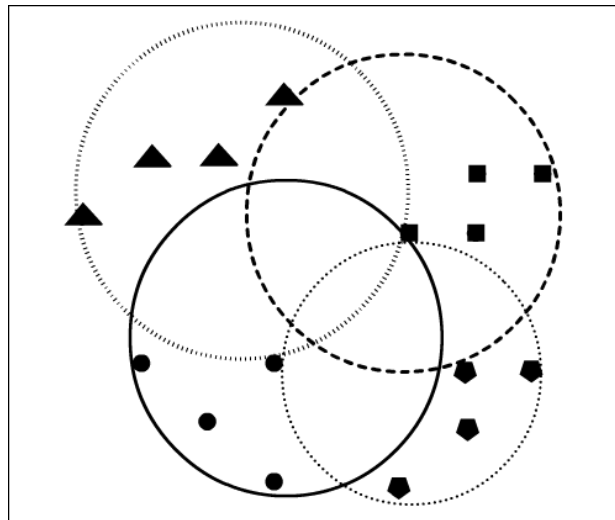


Figura 2. Círculos óptimos de las clases del Ejemplo 1.
Funcionan como superficies de separación.

Al clasificar todos los patrones de (0.1) siempre se obtiene una clasificación perfecta. Ahora, sean los patrones mostrados a continuación, éstos son patrones pertenecientes al conjunto fundamental, pero han sido contaminados con ruido aditivo, sustractivo o mixto.

$$\begin{aligned}
\tilde{x}^1 &= x^1 + 0 \quad 2 &= 1 \quad 3 . \\
\tilde{x}^8 &= x^8 + -2 \quad 0 &= 0 \quad -2 . \\
\tilde{x}^9 &= x^9 + -1 \quad 1 &= -2 \quad 4 . \\
\tilde{x}^{15} &= x^{15} + 0 \quad -1 &= -1 \quad -2 .
\end{aligned}
\tag{0.3}$$

Al aplicar la ecuación **¡Error! No se encuentra el origen de la referencia.**, se obtiene el siguiente resultado para cada patrón:

$$u^1 = \begin{bmatrix} 1.31 \\ -6 \\ -1 \\ -5.2 \end{bmatrix}, u^8 = \begin{bmatrix} -1.92 \\ 1 \\ -4 \\ 1.6 \end{bmatrix}, u^9 = \begin{bmatrix} -4.53 \\ -15 \\ 0 \\ -8.4 \end{bmatrix}, u^{15} = \begin{bmatrix} -3.07 \\ -0.5 \\ -3 \\ 1.9 \end{bmatrix},
\tag{0.4}$$

el siguiente paso es aplicar la expresión **¡Error! No se encuentra el origen de la referencia.** para obtener el vector v , por cada vector:

$$v^1 = \begin{bmatrix} -1.83 \\ -\infty \\ -\infty \\ -\infty \end{bmatrix}, v^8 = \begin{bmatrix} -\infty \\ 1 \\ -\infty \\ -1.3 \end{bmatrix}, v^9 = \begin{bmatrix} -\infty \\ -\infty \\ -3.25 \\ -\infty \end{bmatrix}, v^{15} = \begin{bmatrix} -\infty \\ -\infty \\ -\infty \\ -1 \end{bmatrix}.
\tag{0.5}$$

Finalmente, el índice de cada clase se obtiene por medio de la expresión **¡Error! No se encuentra el origen de la referencia.**, por lo tanto, para \tilde{x}^1 , \tilde{x}^8 , \tilde{x}^9 y \tilde{x}^{15} se obtiene $j=1,2,3,4$ respectivamente, lo cual es correcto. En estos casos la clasificación es correcta aún cuando los patrones han sido afectados con ruido (mixto) y aún cuando \tilde{x}^8 aparece dentro de dos hiper-esferas. Sin embargo, al considerar el siguiente patrón:

$$\tilde{x}^3 = x^3 + 0.05 \quad 0 &= 3.05 \quad 2 ,
\tag{0.6}$$

En este caso el ruido es muy poco, pero al aplicar las expresiones **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** se obtienen los siguientes vectores:

$$u^3 = \begin{bmatrix} -0.12 \\ -4.6 \\ -7.23 \\ -7.89 \end{bmatrix}, v^3 = \begin{bmatrix} -\infty \\ -\infty \\ -\infty \\ -\infty \end{bmatrix}. \quad (0.7)$$

En este caso, la expresión **¡Error! No se encuentra el origen de la referencia.** no puede clasificarlo, ya que se encuentra fuera de todas las hiper-esferas. Como se mencionó en la sección **¡Error! No se encuentra el origen de la referencia.**, si se aplica la expresión **¡Error! No se encuentra el origen de la referencia.** sobre u^3 la clasificación sería correcta; sin embargo, no se tomaría en cuenta la vecindad esférica.

Ejemplo 2. Sean, ahora, las siguientes clases no linealmente separables en \mathbb{R}^3 :

$$\begin{aligned} \text{Clase 1} & \begin{cases} x_1 = 0.5 & 0.5 & 0, & x_3 = 2.5 & 2 & -1, \\ x_2 = -0.5 & 2.5 & 1.5, & x_4 = 1 & 3 & 0. \end{cases} \\ \text{Clase 2} & \begin{cases} x_5 = 2 & 0 & -0.5, & x_7 = 2 & -1.5 & 0.5, \\ x_6 = 0.5 & -2 & 0, & x_8 = 1 & -1 & -0.5 \end{cases} \\ \text{Clase 3} & \begin{cases} x_9 = -0.5 & 0.5 & 0, & x_{11} = -1 & 0 & -0.5, \\ x_{10} = -1 & -1 & 0, & x_{12} = 0 & -0.5 & 0. \end{cases} \end{aligned} \quad (0.8)$$

La Figura 3 muestra una representación gráfica de estos patrones en el espacio.

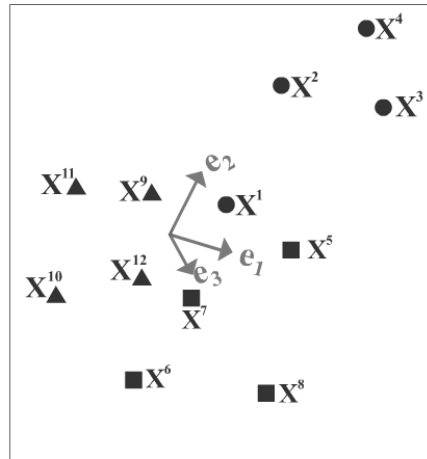


Figura 3. Representación gráfica de las clases del Ejemplo 2. Los círculos, cuadrados y triángulos son puntos pertenecientes a las clases 1, 2, y 3, respectivamente.

Al aplicar **¡Error! No se encuentra el origen de la referencia.**, a cada clase, se obtienen las respectivas híper-esferas, en este caso esferas, cuyos centros y radios son:

$$\begin{aligned}
 c^1 &= 1.19 \quad 1.60 \quad 0.61, & \gamma^1 &= 2.11 \\
 c^2 &= 1.75 \quad -0.75 \quad 1.72, & \gamma^2 &= 2.47 \\
 c^3 &= -0.29 \quad -0.20 \quad 0.05, & \gamma^3 &= 1.06
 \end{aligned}$$

La memoria asociativa geométrica correspondiente se observa a continuación.

$$M = \begin{bmatrix} S^1 = C^1 - \frac{1}{2} \gamma^1{}^2 e_\infty \\ \vdots \\ S^3 = C^3 - \frac{1}{2} \gamma^3{}^2 e_\infty \end{bmatrix} = \begin{bmatrix} S^1 = 1.19e_1 + 1.6e_2 + 0.61e_3 - 0.04e_\infty + e_0 \\ S^2 = 1.75e_1 - 0.75e_2 + 1.72e_3 + 0.25e_\infty + e_0 \\ S^3 = -0.29e_1 - 0.2e_2 + 0.05e_3 - 0.5e_\infty + e_0 \end{bmatrix} \quad (0.9)$$

En la Figura 4 se observan las híper-esferas de cada clase. Al igual que en el ejemplo anterior, éstas son óptimas.

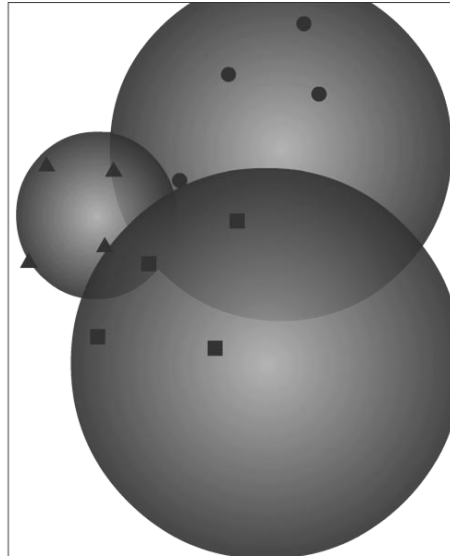


Figura 4. Esferas óptimas de las clases del Ejemplo 2.
Éstas funcionan como superficies de separación.

Como se puede ver, al clasificar los patrones fundamentales se obtiene clasificación perfecta en todos los casos. Sean, ahora, los patrones afectados con ruido mostrados a continuación:

$$\begin{aligned}
 \tilde{x}^1 &= x^1 + 0.5 \quad -0.5 \quad -1 \quad = 0.5 \quad 0.5 \quad 0 . \\
 \tilde{x}^7 &= x^7 + -1 \quad 0.5 \quad -0.5 \quad = 1 \quad -1 \quad 0 . \\
 \tilde{x}^{10} &= x^{10} + 1 \quad 1 \quad 0 \quad = 0 \quad 0 \quad 0 .
 \end{aligned}
 \tag{0.10}$$

Al aplicar la ecuación **¡Error! No se encuentra el origen de la referencia.**, se obtiene el siguiente resultado:

$$u^1 = \begin{bmatrix} 1.25 \\ -0.27 \\ -0.65 \end{bmatrix}, u^7 = \begin{bmatrix} -1.36 \\ 1.25 \\ -0.59 \end{bmatrix}, u^{10} = \begin{bmatrix} 0.04 \\ -0.25 \\ 0.5 \end{bmatrix}.
 \tag{0.11}$$

Al Aplicar la expresión **¡Error! No se encuentra el origen de la referencia.** se obtiene el vector v :

$$v^1 = \begin{bmatrix} -0.97 \\ -\infty \\ -\infty \end{bmatrix}, v^7 = \begin{bmatrix} -\infty \\ -1.79 \\ -\infty \end{bmatrix}, v^{10} = \begin{bmatrix} -2.19 \\ -\infty \\ -0.06 \end{bmatrix} \quad (0.12)$$

El índice de clase se obtiene por medio de la expresión **¡Error! No se encuentra el origen de la referencia.**, por lo tanto, para \tilde{x}^1 , \tilde{x}^7 y \tilde{x}^{10} el valor de j es 1, 2 y 3 respectivamente. Al igual que en el Ejemplo 1, la clasificación es correcta para algunos patrones alterados con ruido.

Como se pudo observar en ambos ejemplos, aunque las memorias asociativas geométricas pueden clasificar perfectamente algunos patrones alterados con ruido, son muy sensibles en ciertos casos; principalmente cuando se trata de patrones localizados cerca del borde de la hiper-esfera.

5.1.2 Ejemplos con patrones reales

Se realizaron dos ejemplos más, pero ahora con datos reales. Las dos bases de datos se obtuvieron de (Asuncion y Newman 2007). Se hicieron tres pruebas, tomando 15, 25 y 30 instancias respectivamente para formar el conjunto fundamental, para entrenar la memoria asociativa geométrica. Las instancias restantes se utilizaron para formar el conjunto de prueba.

Ejemplo 3. Para este ejemplo se utilizó la bien conocida base de datos de la planta del iris. Esta base de datos consiste de tres clases con 50 instancias cada una; cada clase hace referencia a un tipo de planta. Una de las clases es linealmente separable de las otras dos, las otras no son linealmente separables entre sí. Cada instancia tiene cuatro atributos numéricos: longitud y ancho del sépalo, y longitud y ancho del pétalo.

Los resultados de la clasificación se observan en la Tabla 1. La primera columna muestra el número de patrones utilizados para la fase de aprendizaje, la segunda y tercera columnas muestran el porcentaje de patrones correctamente clasificados para el conjunto fundamental y el conjunto de prueba, respectivamente.

Tabla 1. Resultados de la clasificación para el Ejemplo 3

Patrones en el conjunto	Clasificación del conjunto	Clasificación del conjunto
-------------------------	----------------------------	----------------------------

fundamental	fundamental	de prueba
15	100%	90.4%
25	100%	93.3%
30	100%	95.0%

Como se puede apreciar, de esta tabla, la memoria presenta clasificación perfecta debido a que los patrones usados en el conjunto fundamental son esféricamente separables. Además, la clasificación del conjunto de prueba incrementa cuando el número de patrones en el conjunto fundamental también aumenta (ver gráfica de la Figura 5). Esto se debe a que cuando el conjunto fundamental tiene más patrones, el radio de la hiper-esfera aumenta; por lo tanto, una mayor cantidad de patrones del conjunto de prueba se encuentran dentro de la hiper-esfera.

Ejemplo 4. En el segundo ejemplo se utilizó una base de datos de vinos. Estos datos son el resultado de análisis químicos de vinos procesados en la misma región de Italia pero derivados de diferentes cultivos. Los análisis determinaron 13 constituyentes encontrados en cada uno de los tres tipos de vinos. Esta base de datos contiene tres clases de 59, 48 y 71 instancias cada una; donde cada clase se refiere a un tipo de vino. A su vez, cada instancia tiene 13 atributos.

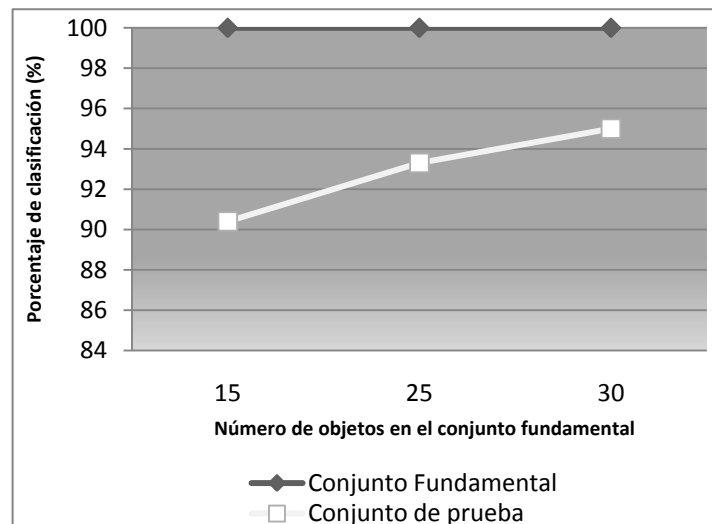


Figura 5. Gráfica con los resultados de la clasificación para el Ejemplo 3.

Los resultados de la clasificación para este ejemplo se observan en la Tabla 2. Las columnas de la Tabla 2 representan lo mismo que las de la Tabla 1.

Tabla 2. Resultados de la clasificación para el Ejemplo 4

Patrones en el conjunto fundamental	Clasificación del conjunto fundamental	Clasificación del conjunto de prueba
15	91.1%	72.1%
25	92.0%	80.5%
30	97.7%	85.2%

Como se puede ver en esta tabla, la memoria no presenta clasificación perfecta. Esto se debe a que los patrones de esta base de datos, a diferencia de la base de datos de la planta del iris, no son esféricamente separables. Y, como en el Ejemplo 3, el porcentaje de clasificación correcto del conjunto de prueba incrementa cuando el número de patrones en el conjunto fundamental también aumenta (ver la gráfica de la Figura 6).

Ambos experimentos se repitieron cien veces, modificando las instancias en el conjunto fundamental. Se obtuvieron, en promedio, los valores que se muestran en la Tabla 1 y en la Tabla 2.

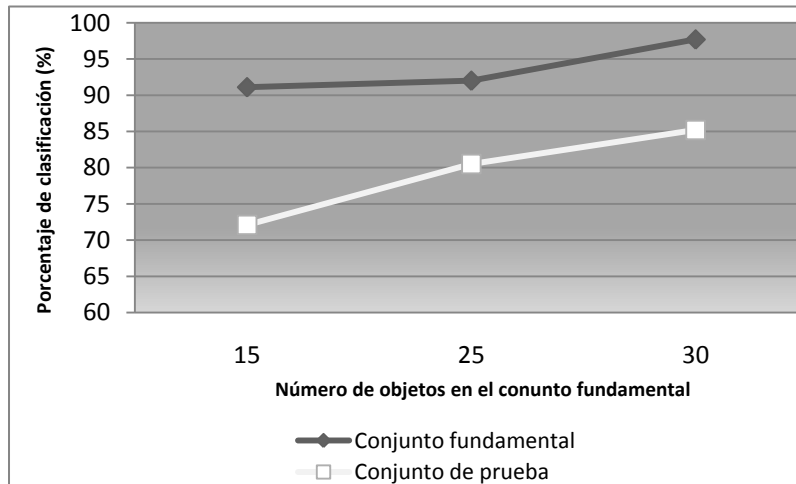


Figura 6. Gráfica con los resultados de la clasificación para el Ejemplo 4.

5.2 Experimentación para el modelo de restauración

5.2.1 Ejemplos numéricos

Ejemplo 5. Sean los siguientes patrones en \mathbb{R}^2 :

$$x^1 = 1 \ 1, x^2 = -1 \ 1, x^3 = 1 \ -1, x^4 = -1 \ -1. \quad (0.13)$$

La Figura 7 muestra una representación gráfica de estos patrones.

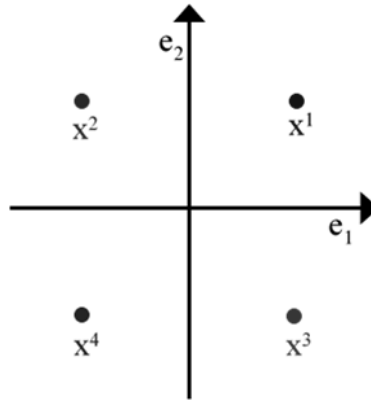


Figura 7. Representación gráfica de los puntos del Ejemplo 5.

Al utilizar el método descrito en las secciones **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** se obtuvieron las hiper-esferas correspondientes a cada patrón. Los centros y radios fueron los esperados:

$$\begin{aligned} c^1 &= 1 \ 1, & \gamma^1 &= 1.99. \\ c^2 &= 1 \ -1, & \gamma^2 &= 1.99. \\ c^3 &= -1 \ 1, & \gamma^3 &= 1.99. \\ c^4 &= -1 \ -1, & \gamma^4 &= 1.99. \end{aligned}$$

Como se puede observar, los centros de las hiper-esferas son sus correspondientes patrones. Otra forma de obtener estos valores, como se explicó al final de la sección **¡Error! No se encuentra el origen de la referencia.**, es encontrando el mínimo de la función **¡Error! No se encuentra el origen de la referencia.**, pero tomando en cuenta las restricciones correspondientes, como se observa en la gráfica de la Figura 8, el círculo y el cuadrado representan las restricciones para los puntos dentro y fuera de la hiper-esfera, respectivamente; el triángulo es el valor mínimo.

Por cualquiera de los dos métodos se obtiene el mismo resultado. La memoria asociativa correspondiente es, por tanto:

$$\mathbf{M} = \begin{bmatrix} S^1 = C^1 - \frac{1}{2} \gamma^{1^2} e_\infty \\ \cdot \\ \cdot \\ S^4 = C^4 - \frac{1}{2} \gamma^{4^2} e_\infty \end{bmatrix} = \begin{bmatrix} S^1 = e_1 + e_2 - 0.98e_\infty + e_0 \\ S^2 = e_1 - e_2 - 0.98e_\infty + e_0 \\ S^3 = -e_1 + e_2 - 0.98e_\infty + e_0 \\ S^4 = -e_1 - e_2 - 0.98e_\infty + e_0 \end{bmatrix}. \quad (0.14)$$

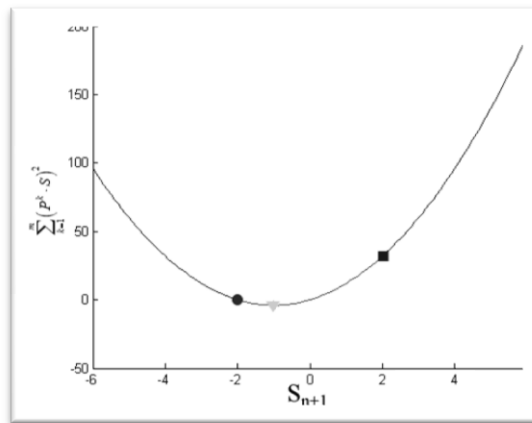


Figura 8. Solución gráfica al problema de minimización.

En la Figura 9 se presentan las hiper-esferas (círculos) de cada patrón.

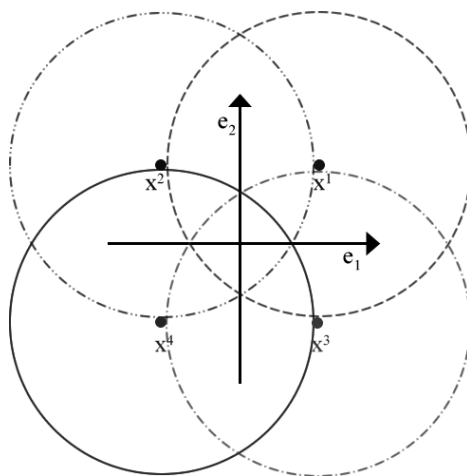


Figura 9. Híper-esferas correspondientes para los patrones del Ejemplo 5.

Nótese que los círculos mostrados en la Figura 9 son óptimos. Al restaurar los puntos del conjunto fundamental siempre se obtiene restauración perfecta. Al considerar los siguientes patrones afectados con ruido:

$$\begin{aligned}
 \tilde{x}^1 &= x^1 + 0 \quad 1.5 &= 1 \quad 2.5 . \\
 \tilde{x}^2 &= x^2 + 0 \quad -1.9 &= 1 \quad -2.9 . \\
 \tilde{x}^3 &= x^3 + 0.5 \quad -0.5 &= -0.5 \quad 0.5 . \\
 \tilde{x}^4 &= x^4 + -1 \quad -0.5 &= -2 \quad -1.5 .
 \end{aligned}
 \tag{0.15}$$

Se puede ver que para todos estos patrones se cumple la condición **¡Error! No se encuentra el origen de la referencia.**. Al aplicar la ecuación **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** se obtiene:

$$v^1 = \begin{bmatrix} -1.13 \\ -\infty \\ -\infty \\ -\infty \end{bmatrix}, v^2 = \begin{bmatrix} -\infty \\ -1.81 \\ -\infty \\ -\infty \end{bmatrix}, v^3 = \begin{bmatrix} -1.25 \\ -\infty \\ -0.25 \\ -1.25 \end{bmatrix}, v^4 = \begin{bmatrix} -\infty \\ -\infty \\ -\infty \\ -0.63 \end{bmatrix}.
 \tag{0.16}$$

Finalmente, los patrones restaurados se obtienen gracias a la expresión **¡Error! No se encuentra el origen de la referencia.**, para $\tilde{x}^1, \tilde{x}^2, \tilde{x}^3$ y \tilde{x}^4 se obtiene $j=1,2,3,4$ respectivamente. Por lo cual se regresa el correspondiente centro de la hiper-esfera. En estos casos la restauración es correcta aún cuando los patrones están afectados con ruido (mixto).

Ejemplo 6. Sea, ahora, el siguiente conjunto fundamental de patrones en \mathbb{R}^3 :

$$x^1 = 2 \quad 2 \quad 2, x^2 = 2 \quad 2 \quad -2, x^3 = -2 \quad -2 \quad 2.
 \tag{0.17}$$

La representación gráfica de estos patrones se observa en la Figura 10. Al utilizar el método descrito en las secciones **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** se obtuvieron las hiper-esferas (esferas) cuyos radios se presentan a continuación.

$$\begin{aligned}
 c_1 &= 2 \quad 2 \quad 2 & \gamma_1 &= 3.99. \\
 c_2 &= 2 \quad 2 \quad -2 & \gamma_2 &= 3.99. \\
 c_3 &= -2 \quad -2 \quad 2 & \gamma_3 &= 4.61.
 \end{aligned}$$

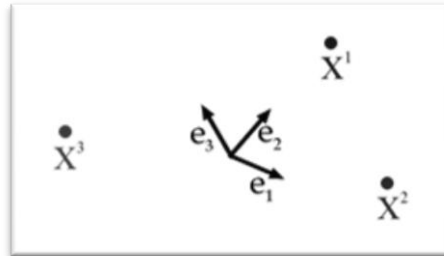


Figura 10. Híper-esferas correspondientes para los patrones del Ejemplo 6.

Como se puede observar, los centros corresponden a los patrones correspondientes. Al igual que en el ejemplo anterior, estos centros se pueden obtener al calcular analíticamente el mínimo. En la gráfica de la Figura 11 se presenta la solución para las híper-esferas S^1 y S^2 .

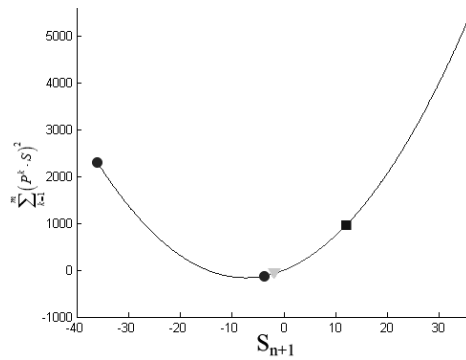


Figura 11. Solución gráfica para las híper-esferas 1 y 2.

En la gráfica de la Figura 12 se presenta la solución para la híper-esfera restante.

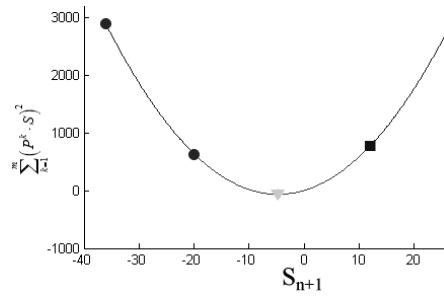


Figura 12. Solución gráfica para la hiper-esfera 3.

La memoria asociativa geométrica es:

$$\mathbf{M} = \begin{bmatrix} S^1 = C^1 - \gamma^1 \cdot e_\infty \\ S^2 = C^2 - \gamma^2 \cdot e_\infty \\ S^3 = C^3 - \gamma^3 \cdot e_\infty \end{bmatrix} = \begin{bmatrix} S^1 = 2e_1 + 2e_2 + 2e_3 - 1.99e_\infty + e_0 \\ S^2 = 2e_1 + 2e_2 - 2e_3 - 1.99e_\infty + e_0 \\ S^3 = -2e_1 - 2e_2 + 2e_3 - 4.67e_\infty + e_0 \end{bmatrix}. \quad (0.18)$$

Las hiper-esferas, en este caso esferas como tales, correspondientes se observan en la Figura 13, como se puede observar, al igual que en los ejemplos anteriores, las hiper-esferas son óptimas.

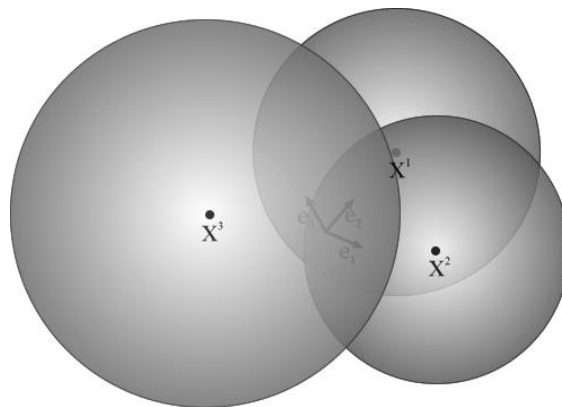


Figura 13. Híper-esferas correspondientes a los patrones del Ejemplo 6

Ahora, sean los siguientes patrones a ser restaurados. Éstos son patrones, del conjunto fundamental, afectados con algún tipo de ruido.

$$\begin{aligned}
\tilde{x}^1 &= x^1 + 2 \quad -1 \quad 2 &= 4 \quad 1 \quad 4 . \\
\tilde{x}^2 &= x^2 + -2 \quad -1 \quad -3 &= 0 \quad 1 \quad -5 . \\
\tilde{x}^3 &= x^3 + -1 \quad 3 \quad 0 &= -3 \quad 1 \quad 2 .
\end{aligned}
\tag{0.19}$$

Si se aplican las expresiones **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** se obtienen los siguientes vectores:

$$v^1 = \begin{bmatrix} -4.5 \\ -\infty \\ -\infty \end{bmatrix}, v^2 = \begin{bmatrix} -\infty \\ -7 \\ -\infty \end{bmatrix}, v^3 = \begin{bmatrix} -\infty \\ -\infty \\ -5 \end{bmatrix}.
\tag{0.20}$$

Los patrones restaurados se obtienen al regresar el centro de la j -ésima hiper-esfera. En este caso j es el valor obtenido por la expresión **¡Error! No se encuentra el origen de la referencia.**, que en este caso para \tilde{x}^1 , \tilde{x}^2 y \tilde{x}^3 se obtiene $j = 1, 2, 3$ respectivamente. Lo cual es correcto.

5.2.2 Ejemplos con patrones reales

Para probar el método propuesto se utilizaron tres conjuntos de imágenes.

Ejemplo 7. Para este ejemplo se utilizaron fotografías de animales. Se trata de 25 imágenes, de 300 píxeles de ancho por 300 de alto en escala de grises. Algunas muestras se pueden observar en la Figura 14. En este caso no se utilizó ningún pre-procesamiento a las imágenes.

Los patrones se obtuvieron al hilar fila a fila de cada imagen hasta formar un vector de tamaño 90,000 (300×300). Con estos vectores se formó el conjunto fundamental.

Se construyeron seis conjuntos de prueba, cada uno formado por las mismas imágenes del conjunto fundamental, pero afectadas con 5%, 10%, 15%, 30%, 40%, y 50%, respectivamente, de ruido mezclado aleatorio. Al igual que con el conjunto fundamental, se formaron los vectores con las filas de las imágenes.



Figura 14. Muestras de algunas de las imágenes utilizadas para el Ejemplo 7.

Ejemplo 8. Adicionalmente, se realizó un segundo experimento con las mismas características que el Ejemplo 7. Solo que en este caso se usaron 22 retratos de matemáticos y científicos famosos, como los mostrados en la Figura 15. En este caso, las imágenes son de 300 por 400 píxeles, también en escala de grises.

De la misma forma que en el Ejemplo 7, los patrones se formaron al unir las filas de cada imagen, hasta formar un vector de tamaño 12,000 (300×400) para el conjunto fundamental. Y, para los conjuntos de prueba se afectaron las imágenes con la misma cantidad de ruido que en el ejemplo anterior.



Figura 15. Muestras de algunas de las imágenes utilizadas para el Ejemplo 8.

En la Figura 16 se puede observar una imagen afectada por los porcentajes de ruido mencionados, por cada ejemplo.

Los resultados de la restauración se muestran en la Tabla 3. La primera columna presenta el conjunto de imágenes utilizado, las siguientes columnas muestran el número de imágenes recuperadas, en forma correcta, para los conjuntos de prueba correspondientes.



Figura 16. Muestras de imágenes afectadas con ruido para el Ejemplo 7 (arriba) y el Ejemplo 8 (abajo). De izquierda a derecha, imagen con 0% (conjunto fundamental), 5%, 10%, 15%, 30%, 40% y 50% de ruido, respectivamente.

Como se puede observar, la recuperación para el caso de las imágenes de animales fue correcta cuando las imágenes fueron afectadas hasta con 30% de ruido. Solo una

imagen para el caso de matemáticos y científicos no fue recuperada para el caso de un 30% de ruido. En ambos casos se puede observar que al aumentar la cantidad de ruido en las imágenes, la cantidad de patrones recuperados tiende a disminuir considerablemente (ver gráfica de la Figura 17).

Tabla 3. Resultados de la restauración para los Ejemplo 7 y Ejemplo 8

Conjunto	0%	5%	10%	15%	30%	40%	50%
Animales	25	25	25	25	24	12	7
Científicos	22	22	22	22	22	15	10

Este experimento se repitió cien veces variando la posición de los píxeles de ruido sobre las imágenes. Los resultados fueron los mismos que los que se muestran en la Tabla 3 para 0% a 25% de ruido. Esto ocurre porque el **¡Error! No se encuentra el origen de la referencia.** se cumple para esas imágenes con esa cantidad de ruido. Para 30% de ruido los resultados oscilan entre 25 y 21 imágenes restauradas (el promedio fue de 24) para el conjunto uno. Para el conjunto dos, los resultados son entre 22 y 21 imágenes restauradas (en este caso el promedio fue de 22). Para 40% y 50% de ruido el promedio de las imágenes restauradas es el que se observa en la Tabla 3.

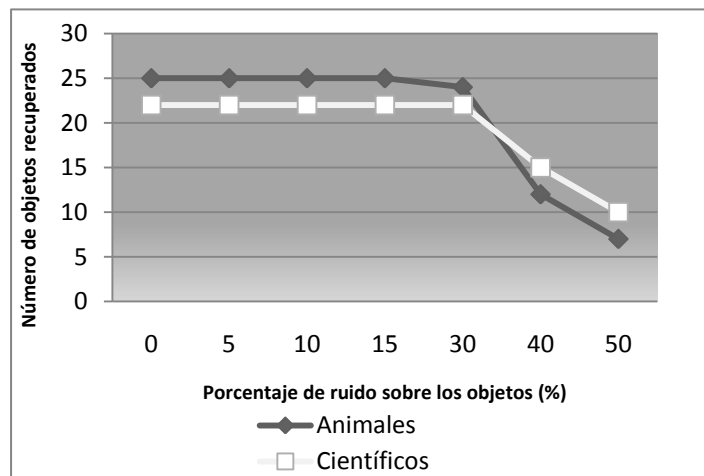


Figura 17. Gráfica con los resultados de la clasificación para los conjuntos de animales y científicos.

Algo que hay que notar, como una característica de las memorias asociativas geométricas, es que la restauración es correcta o incorrecta solamente. No es posible

llegar a una recuperación parcial, como sucede cuando se utiliza algún tipo de filtro, como se verá en la siguiente sección.

Ejemplo 9. Para mostrar la eficacia de modelo ante perturbaciones en imágenes como el desenfoque o el difuminado se decidió hacer este experimento. En este caso se utilizaron 18 imágenes como las mostradas en la Figura 18.



Figura 18. Muestras de las imágenes utilizadas en el Ejemplo 9.

Estas imágenes se caracterizan por ser las más utilizadas en la comunidad de procesamiento de imágenes. Dichas imágenes son de 200 X 200 píxeles y, al igual que en los ejemplos previos, están en escala de grises.

De la misma forma que en los ejemplos anteriores, los patrones en este ejemplo se formaron encadenando las filas de cada imagen hasta formar un vector de tamaño 40,000. Estos patrones se utilizaron para formar el conjunto fundamental.

Para formar los conjuntos de prueba se distorsionaron las imágenes con desenfoque artificial mediante la herramienta *desenfoque Gaussiano* del programa Macromedia Fireworks, la cual permite variar el radio de desenfoque (k) de una imagen. Algunos ejemplos de este desenfoque en las imágenes de la Figura 18 se presentan en la Figura 19. En este caso el valor de k fue de 8.



Figura 19. Muestras de imágenes desenfocadas.

Para realizar las pruebas se utilizó el desenfoco variando el valor del radio k con esa herramienta. Se obtuvieron 10 conjuntos de imágenes desenfocadas, para eso se modifico el valor de k desde 0 (conjunto fundamental) hasta 18 con intervalos de 2. Ejemplo de dos imágenes con estos desenfocos se observan en la Figura 20 para esos valores de k . Note el nivel de desenfoco de las imágenes con el valor de $k = 12, \dots, 18$.

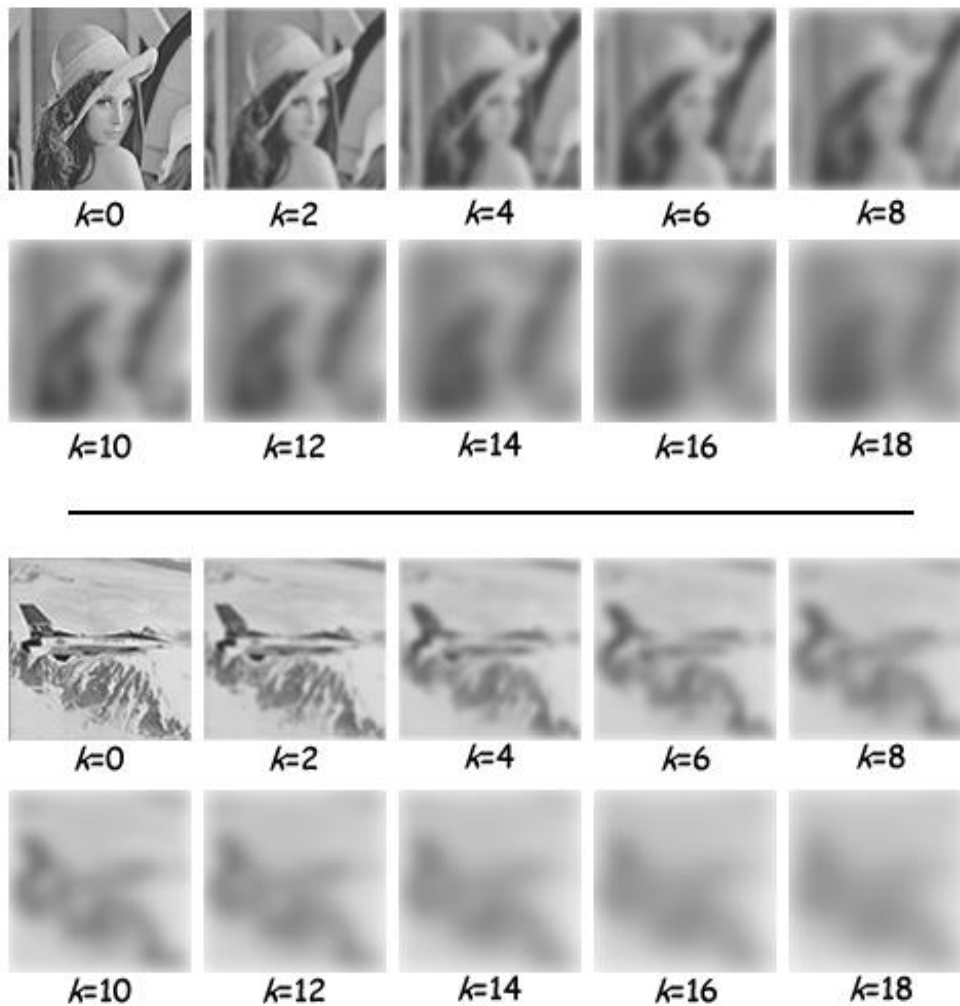


Figura 20. Dos imágenes desenfocadas para distintos valores de k .

La memoria asociativa geométrica se entrenó con el conjunto fundamental. Los demás conjuntos de imágenes desenfocadas se utilizaron para realizar la fase de restauración. Los resultados se observan en la gráfica de la Figura 21. Se puede observar que, hasta el nivel de desenfoco con $k=10$, la recuperación es correcta. Con un valor de $k=12$ solamente una de las imágenes no puede ser recuperada, conforme aumenta el nivel de distorsión, el número de imágenes recuperadas correctamente disminuye hasta casi un 55%.

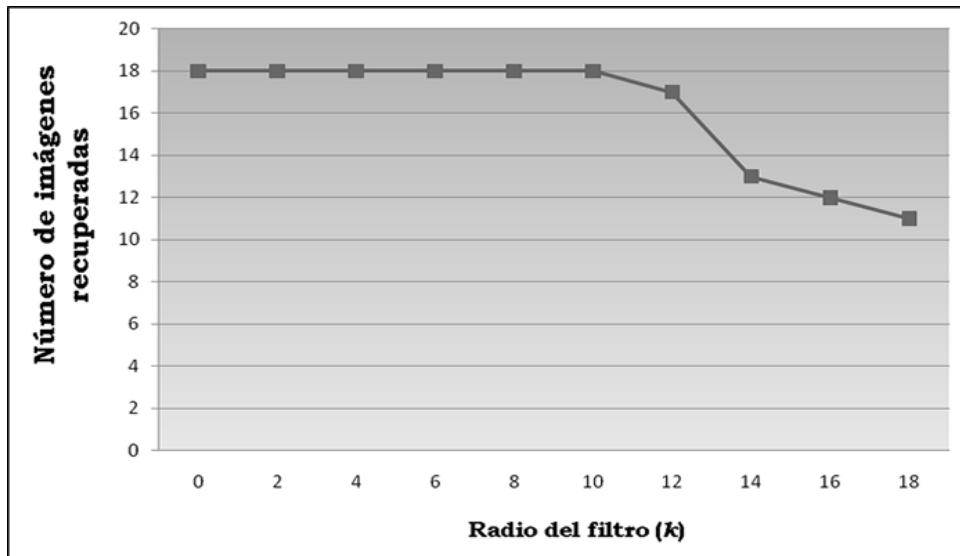


Figura 21. Resultado de la recuperación de las imágenes de los conjuntos de prueba en imágenes desenfocadas.

5.2.3 Comparación con otros modelos















































En esta sección se presentan los resultados obtenidos al realizar la restauración de los dos conjuntos de imágenes utilizados en el Ejemplo 7 y en el Ejemplo 8. En este caso se utilizaron los mismos conjuntos de imágenes, afectadas con las mismas proporciones de ruido (Figura 16), para probar la funcionalidad con los ya conocidos filtros de promedio, mediana, morfológico (apertura-clausura) (información sobre estos filtros puede verse en (Gonzalez y Woods 2008)); se utilizaron, también, dos herramientas del programa comercial Paint Shop Pro X2: *removedor de puntos* y *removedor de polvo*; además, se utilizó la memoria asociativa basada en el método de sub-patrones (Cruz, Sossa y Barrón 2007).

En la Tabla 4 y en la Tabla 5 se observan los resultados de la restauración de una de las imágenes de cada conjunto. La primera columna muestra la imagen utilizada, las siguientes columnas, de izquierda a derecha, muestran las imágenes restauradas usando el filtro de promedio, la herramienta remover polvo, la herramienta remover puntos, el filtro de mediana, el filtro morfológico (apertura-clausura), la técnica de los sub-patrones y, por último, la memoria asociativa geométrica (MAG). Cada renglón de arriba hacia abajo representa los resultados obtenidos, para todos los métodos

mencionados, al restaurar una imagen con 5%, 10%, 15%, 30%, 40% y 50% de ruido sal y pimienta aleatorio, respectivamente.


A partir de estas tablas se puede observar que las imágenes recuperadas con las memorias asociativas geométricas presentaron los mejores resultados ya que se recuperó fielmente la imagen original; a diferencia de los otros métodos donde se recupera parte de la información original.

Tabla 4. Comparación, entre varios modelos, para la restauración de una imagen, afectada con ruido, del conjunto de animales.

Porcentaje de ruido	Imagen con ruido	Filtro de promedio	Remover polvo	Remover puntos	Filtro de mediana	Filtro morfológico	Sub-patrones	MAG
5%								
10%								
15%								
30%								
40%							NA	
50%							NA	

El método de sub-patrones presenta resultados similares a los de la memoria asociativa geométrica debido a que la información de los patrones (las imágenes) se codifica, precisamente, dentro de la propia memoria. Por lo tanto, al restaurar un objeto afectado con ruido, la memoria regresa la información del objeto que previamente había codificado.

Tabla 5. Comparación, entre varios modelos, para la restauración de una imagen, afectada con ruido, del conjunto de científicos.

Porcentaje de ruido	Imagen con ruido	Filtro de promedio	Remover polvo	Remover puntos	Filtro de mediana	Filtro morfológico	Subpatrones	MAG
5%								
10%								
15%								
30%							NA	
40%							NA	
50%							NA	

A diferencia de los filtros, en donde no es posible restaurar al cien por ciento el objeto original, una memoria asociativa permite restaurarlo completamente. Sin embargo, esto puede tener inconvenientes; ya que, si el objeto está muy afectado por ruido, la memoria podría regresar un objeto completamente diferente al original. En el caso de una memoria asociativa no es posible hablar de una restauración con cierto porcentaje de error, como en el caso de algún filtro; es decir, con memorias asociativas solo hay dos posibles resultados: a) o se restaura bien (al cien por ciento) el objeto original, o b) no se restaura y, en este caso, regresa un objeto diferente.

En la Tabla 6 se puede observar el error cuadrático medio de las imágenes restauradas de la Tabla 4. Las columnas de esta tabla muestran el error obtenido entre

las imágenes restauradas de cada método utilizado y su correspondiente imagen original. Cada renglón de esta misma tabla corresponde a una imagen con 5%, 10%, 25%, 30%, 40% y 50% de ruido respectivamente, como en la Tabla 4 y en la Tabla 5. La Tabla 7 muestra la misma información que la Tabla 6 pero con las imágenes de la Tabla 5. Como se puede observar, el error cuadrático medio para las memorias asociativas geométricas y para los sub-patrones siempre es cero, esto se debe a que las imágenes originales están codificadas dentro de la memoria. Ésta es, precisamente, la principal ventaja de las memorias asociativas.

Los otros modelos muestran buenos resultados, pero cuando los patrones presentan pérdida parcial de la información, estos métodos no funcionan correctamente. Por ejemplo, las imágenes mostradas en la Figura 26 no se pueden restaurar por medio de esos métodos, pero las memorias asociativas geométricas son capaces de hacerlo.

Tabla 6. Error cuadrático medio para las imágenes de la Tabla 4

Ruido	Filtro de promedio	Remover polvo	Remover puntos	Filtro de mediana	Filtro morf.	Sub-patrones	MAG
5%	1165529	632256	328043	461169	3352608	0	0
10%	1833253	894465	503496	509266	4980297	0	0
25%	2858467	1312237	1312237	717805	6712505	0	0
30%	6306509	3498588	2893229	1730484	12392659	0	0
40%	8877902	6272391	4837072	3252628	15759837	NA	0
50%	12031332	10442603	8683319	6375399	19111305	NA	0

Como se puede observar, el error cuadrático medio para las memorias asociativas geométricas y para los sub-patrones siempre es cero, esto se debe a que las imágenes originales están codificadas dentro de la memoria. Ésta es, precisamente, la principal ventaja de las memorias asociativas.

Los otros modelos muestran buenos resultados, pero cuando los patrones presentan pérdida parcial de la información, estos métodos no funcionan

correctamente. Por ejemplo, las imágenes mostradas en la Figura 22 no se pueden restaurar por medio de esos métodos, pero las memorias asociativas geométricas son capaces de hacerlo.

Tabla 7. Error cuadrático medio para las imágenes de la Tabla 5

Ruido	Filtro de promedio	Remover polvo	Remover puntos	Filtro de mediana	Filtro morf.	Sub-patrones	MAG
5%	738906	673584	331396	516934	754959	0	0
10%	926361	686920	435422	562626	1058108	0	0
25%	1187129	703110	512046	572439	1397366	0	0
30%	2109969	963170	1058996	801835	2692502	NA	0
40%	2988743	1410100	1534965	1079258	3620676	NA	0
50%	3911128	2018000	2263242	1578445	4812627	NA	0

Ejemplo 10. Se realizó otro experimento para probar el potencial de las memorias asociativas geométricas contra los filtros. En este caso las imágenes de la Figura 14 fueron afectadas con una excesiva cantidad de ruido con el propósito de eliminar parte de su información. Ejemplos de este tipo de imágenes se observan en la Figura 22. Se utilizó, también, la misma memoria asociativa geométrica de los experimentos anteriores.

El experimento en sí consiste en tomar cada una de las imágenes de la Figura 14 y, quitarle parte de su información. Este efecto se logra de varias maneras, la que se utilizará en este experimento consiste en elegir un pixel al azar en la imagen, que será el centro de la distorsión. Usando dicho punto como centro, a continuación se dibuja un círculo o un rectángulo de tamaño, también, aleatorio (usualmente un número grande), al cual se le agregará un efecto de difuminado en los bordes. Este proceso repite dos o tres veces en una misma imagen, para eliminar varias regiones de la imagen.

En todos los casos, siempre que el **¡Error! No se encuentra el origen de la referencia.** se cumpla, las imágenes se restauraron correctamente con las memorias asociativas geométricas. Es decir, siempre y cuando el ruido agregado a la imagen no

sobrepase el radio de la esfera, siempre habrá recuperación correcta y total de la imagen.

También se realizaron experimentos con el conjunto de científicos aplicando el mismo procedimiento para obtener distorsiones similares. Al igual que con el conjunto de animales, si el **¡Error! No se encuentra el origen de la referencia.** se cumple, las memorias asociativas geométricas son capaces de restaurar correctamente y completamente las imágenes.



Figura 22. Muestras de imágenes con información parcialmente eliminada

A partir de estos resultados se puede observar que, usando una memoria asociativa, es posible recuperar imágenes a partir de una parte de ellas. Una posible aplicación de este experimento es la restauración de imágenes a partir de una versión más pequeña de la misma.

5.3 Experimentación para el modo de clasificación no supervisada

Se va a probar, ahora, el desempeño de las memorias asociativas propuestas en la tarea de clasificación no supervisada. Esta tarea es importante ya que muchas veces no se tiene una *etiqueta* para los patrones de entrada; el objetivo de la clasificación no

supervisada es, precisamente, encontrar esa etiqueta o clase para los patrones de entrada a partir de las características de los mismos.

Ejemplo 11. Sea el conjunto de puntos en \mathbb{R}^2 mostrados en la Figura 23. Se puede observar que dicha imagen presenta dos cúmulos visibles de puntos. Estos puntos se generaron por medio de dos distribuciones gaussianas, una con centro en $4,4$ y otra con centro en $-4,-4$.

Al aplicar el algoritmo de la sección **¡Error! No se encuentra el origen de la referencia.**, se obtienen las gráficas mostradas en las imágenes de la Figura 24, Figura 25 y la Figura 26. La solución en cada caso, como en la mayoría de los algoritmos de clasificación no supervisada, depende, principalmente, de la posición de los primeros centros generados aleatoriamente. La mejor solución se presenta en la gráfica de la Figura 26.

Ya que los puntos se generaron a través de una distribución gaussiana, se considera la mejor solución aquella que aproxime, precisamente, a esa gaussiana. De ahí que el mejor resultado sea el que se presenta en la Figura 26. Este experimento se repitió 100 veces, variando en cada caso los centros de inicio, la mejor solución se obtuvo 34 veces, con diferentes valores de inicio, las otras 64 veces se obtuvieron resultados a las gráficas de la Figura 24 y la Figura 25.

Sin embargo, como los centros de inicio se generan aleatoriamente, no se puede hablar de un desempeño correcto o incorrecto del algoritmo, simplemente se desea mostrar que el algoritmo logra soluciones correctas en algún momento.

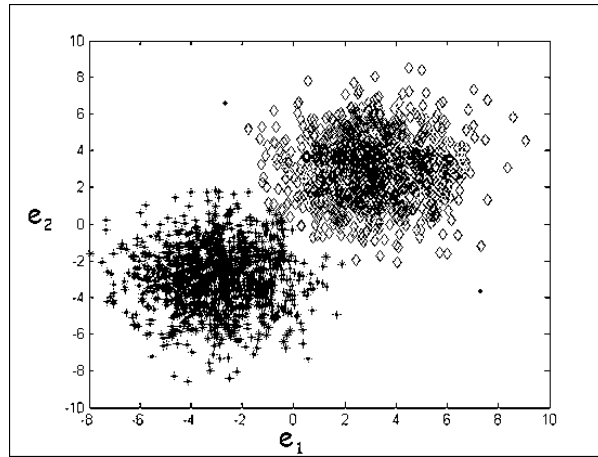


Figura 23. Conjunto de puntos en \mathbb{R}^2 a agrupar.

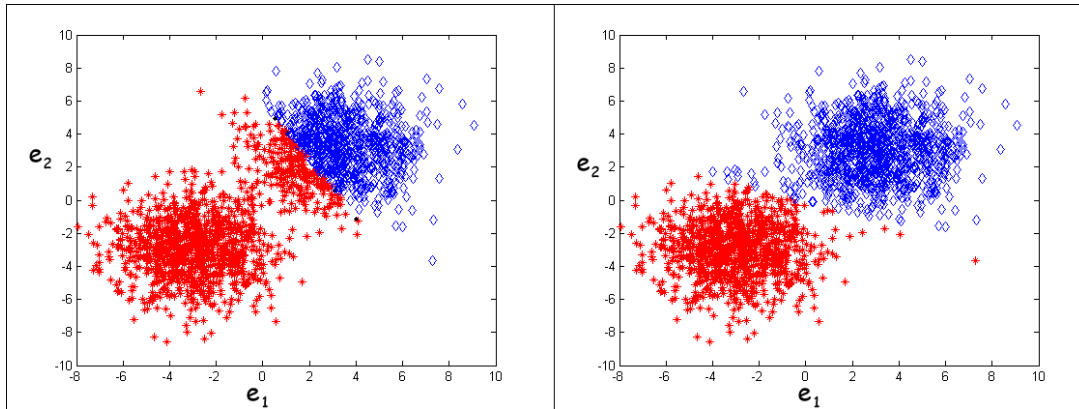


Figura 24. Diferentes soluciones que resuelven el problema de agrupamiento de la Figura 23.

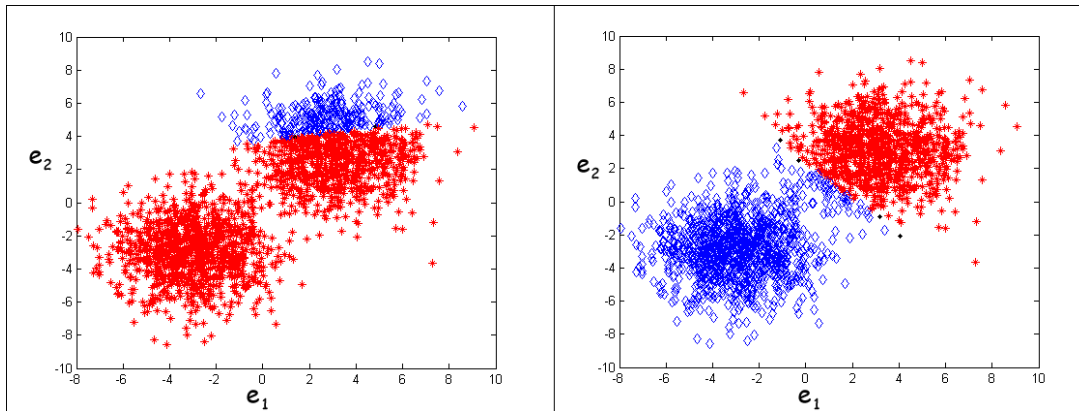


Figura 25. Otras soluciones que resuelven el problema de agrupamiento de la Figura 23.

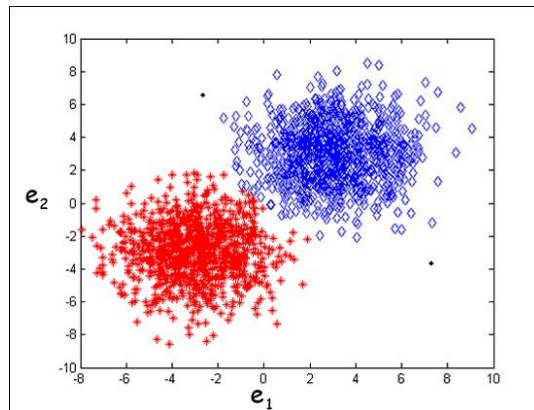


Figura 26. Mejor solución que resuelve el problema de agrupamiento de la Figura 23.

Ejemplo 12. Ahora sean los puntos en \mathbb{R}^2 mostrados en la. En dicha gráfica se pueden observar, a primera vista, cuatro cúmulos de puntos principales. En este caso se utilizaron cuatro distribuciones gaussianas con centros en $4,4$, $4,-4$, $-4,4$ y $-4,-4$.

Si se aplica el algoritmo de la sección **¡Error! No se encuentra el origen de la referencia.**, se obtienen las gráficas mostradas en las imágenes de la Figura 28, la Figura 29 y la Figura 30. De nuevo, la solución en cada caso, depende, principalmente, de la posición de los primeros centros generados aleatoriamente. La mejor solución se presenta en la gráfica de la Figura 30.

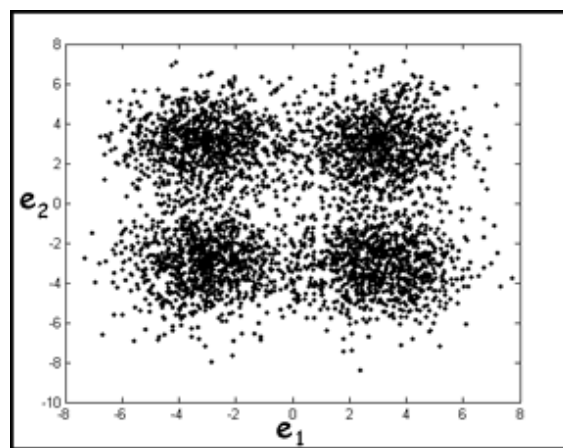


Figura 27. Conjunto de puntos para el Ejemplo 12.

Al igual que en el Ejemplo 11, como los puntos de inicio se generaron por medio de una distribución gaussiana, se considera la mejor solución aquella que aproxime, precisamente, a esas gaussianas. De ahí que el mejor resultado sea el que se presenta en la Figura 30. Este experimento también se repitió 100 veces, variando en cada caso los centros de inicio, la mejor solución se obtuvo 41 veces, con diferentes valores de inicio, las otras 59 veces se obtuvieron resultados similares a las gráficas de la Figura 28 y la Figura 29.

Sin embargo, al igual que con el ejemplo anterior, como los centros de inicio se generan aleatoriamente, el algoritmo logra encontrar soluciones correctas en algún momento.

En algunas de las gráficas presentadas, se presentan algunos puntos que no pertenecen a ningún agrupamiento; son puntos que se encuentran afuera de todas las hiper-esferas, por lo tanto, no pertenecen a una clase específica. Esto se debe, en algunos casos, al tipo de superficies de separación que se están utilizando. Dependiendo del tipo de problema esto puede ser o no una ventaja.

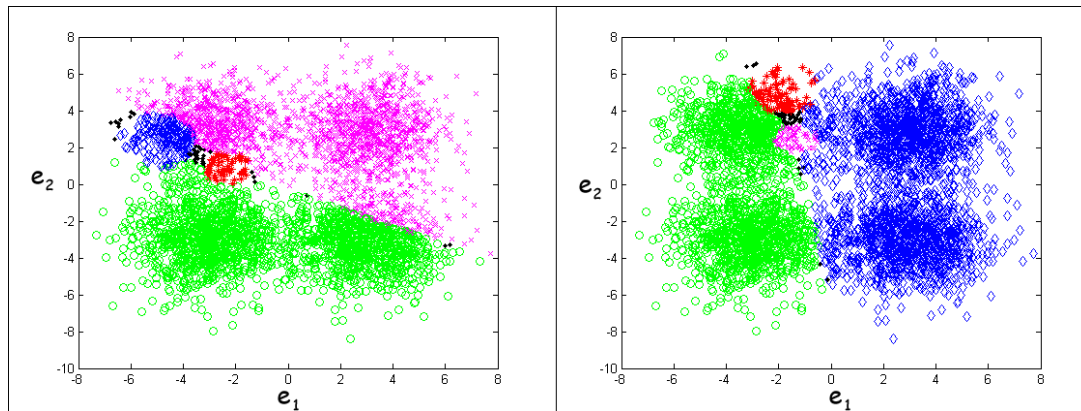


Figura 28. Diferentes soluciones que resuelven el problema de agrupamiento de la Figura 27.

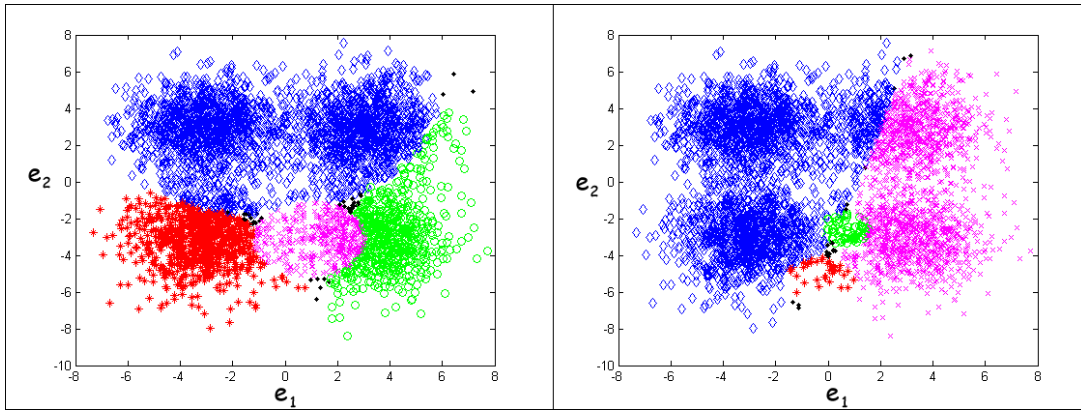


Figura 29. Otras soluciones que resuelven el problema de agrupamiento de la Figura 27.

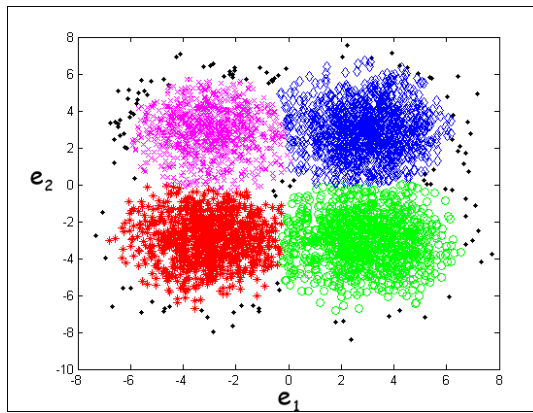


Figura 30. Mejor solución que resuelve el problema de agrupamiento de la Figura 27.

5.4 Discusión

Con los ejemplos numéricos se puede observar más claramente el carácter geométrico de las memorias propuestas. A diferencia de otros modelos de memorias asociativas, o inclusive de otros modelos de clasificación o restauración, la solución de las memorias asociativas geométricas está basada en la geometría de los patrones, ya que se utilizan hiper-esferas como superficies de separación. Además, para determinar que un patrón dado pertenece a una clase específica simplemente es necesario determinar si el patrón está dentro de la hiper-esfera que representa, precisamente, a esa clase.

Los tres modelos presentados son, de alguna manera, similares entre sí. Los tres se dividen en dos fases. En la fase de entrenamiento se construyen las hiper-esferas de separación, mediante una función de optimización, pero tomando en cuenta las

restricciones de cada modelo. Para la fase de clasificación o restauración se utiliza un producto interior entre el patrón nuevo a clasificar y la memoria, y una función de máximo.

Como se pudo observar, por medio de los experimentos, el modelo de restauración es muy similar al modelo de clasificación. Las hiper-esferas obtenidas con el proceso descrito son óptimas, en algún sentido. Como se pudo observar, por los experimentos presentados, el modelo siempre presenta ya se recuperación correcta o incorrecta. A diferencia de un filtro tradicional que recupera la información parcialmente, una memoria asociativa recupera la información original. Esto se puede comprobar al observar la Tabla 6 y la Tabla 7, donde el error cuadrático medio obtenido en las memorias asociativas es siempre cero y en los demás modelos es mayor que cero.

En cuanto al modo de clasificación supervisada se mostró que, al igual que los métodos tradicionales, depende, en gran medida, de los primeros centros generados. Una diferencia del modelo presentado es que puede considerar algunos puntos como no pertenecientes a alguna de las clases seleccionadas. Esto puede ser, dependiendo de la naturaleza del problema a resolver, una ventaja o una desventaja.

Aunque el modelo de clasificación no supervisada no es perfecto, se puede considerar como un nuevo algoritmo de agrupamiento que utiliza, además de una memoria asociativa, operadores y operaciones del álgebra geométrica conforme para su funcionamiento.

También se mostró que el algoritmo generó, en algún momento, la solución correcta al problema.

CAPÍTULO

6. CONCLUSIONES Y TRABAJO

FUTURO

En esta sección se dan las conclusiones a las que se llegaron a lo largo del desarrollo de esta investigación. Así mismo, se enlista el conjunto de aportaciones a que se llegó a través de la realización de esta tesis. De igual forma se da un conjunto de directivas para la realización de trabajos futuros relacionados con este trabajo de tesis. Finalmente, se da una lista de las publicaciones derivadas de este trabajo.

6.1 Conclusiones

En este trabajo se describe un nuevo modelo de memoras asociativas que utilizan operadores y operaciones del álgebra geométrica, en particular del modelo conforme. Las memorias asociativas geométricas utilizan superficies de separación esféricas para su operación. De hecho, la memoria asociativa es una matriz cuyos renglones son, precisamente, las híper-esferas de clase.

A diferencia de una memoria tradicional, que puede verse como una red neuronal de una sola capa, las memorias asociativas geométricas constan de dos capas. Y, a diferencia de una red neuronal, estas memorias convergen en un solo paso. Se desarrollaron tres modos de operación para estas memorias, para resolver los problemas de clasificación supervisada y no supervisada y la restauración de patrones.

El modelo de clasificación, como cualquier memoria asociativa, se divide en dos fases: la fase de entrenamiento y la fase de clasificación. En la fase de entrenamiento se construyen las vecindades esféricas. Se presentaron varias maneras de construir las vecindades esféricas, en particular se desarrolló un método que utiliza una función de optimización que toma en cuenta tanto los patrones pertenecientes a una clase (puntos dentro de la hiper-esfera) como los patrones de las demás clases (puntos fuera de la hiper-esfera). La hiper-esfera obtenida de esta manera es óptima, en términos de programación cuadrática.

La fase de clasificación consiste en aplicar un producto interior entre la memoria generada y un patrón en su representación conforme. Al vector obtenido se le aplica un ajuste para determinar si el patrón se encuentra dentro o fuera de la hiper-esfera. Finalmente se utiliza una función de argumento máximo para obtener el índice que indica la clase a la que pertenece ese patrón.

Se presentaron condiciones formales bajo las cuales el modelo puede funcionar. En particular, se presentaron los casos de la clasificación perfecta y robusta. El modelo funciona bien cuando las clases son esféricamente separables.

También se presentaron algunos ejemplos numéricos y con patrones reales. El modelo funciona mejor cuando los patrones a clasificar se encuentran más cerca del centro de la hiper-esfera. Los patrones que se encuentran más cerca del borde y están afectados con ruido podrían no ser clasificados correctamente. No es posible clasificar a aquellos patrones que se encuentren afuera de todas las hiper-esferas. Esto se puede solucionar al quitar el ajuste en el segundo paso de la clasificación, pero entonces el modelo funcionaría como un clasificador de distancia mínima.

El modelo de restauración es, de alguna manera, similar al modelo de clasificación. También se divide en dos fases. En la fase de entrenamiento se construyen las hiper-esferas de separación. Al igual que el modelo de clasificación, se presentaron algunas estrategias para construir las hiper-esferas, al final se optó por una función de optimización, como en el modelo de clasificación, pero tomando en cuenta que el centro de la hiper-esfera es el patrón mismo. De la misma forma que en el modelo de clasificación, la hiper-esfera obtenida por este método es óptima.

La fase de clasificación es la misma que en el modelo de clasificación pero agregando un paso más. Después de obtener el índice por medio de la función máximo, la memoria regresa el centro de la hiper-esfera como patrón restaurado.

Se mostraron las condiciones para restauración perfecta y robusta. La cantidad de ruido que un patrón puede admitir se encuentra dada por el radio de la hiper-esfera correspondiente. Como en el modelo de clasificación, cuando un patrón afectado por ruido se encuentra afuera de todas las hiper-esferas no es posible la restauración.

Debido a que la función de optimización depende de una sola variable, se puede encontrar la hiper-esfera óptima utilizando la derivada de la función. Se mostraron también los resultados gráficamente del valor mínimo, que coincidieron con los valores obtenidos por medio de la función de optimización.

Se mostraron ejemplos con patrones reales, en este caso imágenes, para mostrar la eficiencia del modelo descrito. Como se pudo observar en estos experimentos, las imágenes recuperadas siempre presentan un error cuadrático medio de cero; es decir siempre se restaura la imagen original. Con un filtro, recuperar una imagen con error cuadrático medio no es posible.

Se probaron imágenes afectadas con ruido mezclado aleatorio, imágenes con desenfoque e imágenes con falta de información. En estos tres casos el modelo presentado presentó mejores resultados que al usar los filtros tradicionales. Además, los filtros no son capaces de restaurar imágenes con falta de información.

Hay que recalcar que el método de restauración descrito en este trabajo es un método de restauración de patrones, no es un método de restauración de imágenes. El hecho de que en los experimentos se utilicen imágenes, para mostrar el desempeño, se debe a que visualmente son más fáciles de interpretar y los resultados se observan con mayor claridad que al utilizar otro tipo de objetos.

Aunque el modo de operación para clasificación no supervisada no es propiamente una memoria asociativa, éstas juegan un papel importante para el algoritmo desarrollado. El nuevo algoritmo, basado en el algoritmo de k -medias, resuelve el problema del agrupamiento.

Los agrupamientos se generan por medio de un producto interior y una función de máximo, entre un punto y la memoria asociativa geométrica. A diferencia del algoritmo tradicional no se calculan las distancias entre los puntos y cada centro generado. De alguna forma, esto reduce la complejidad del algoritmo.

A diferencia del algoritmo tradicional, en el algoritmo propuesto algunos puntos de inicio podrían no ser considerados como pertenecientes a una clase específica. En algunos tipos de problemas esto puede ser una ventaja porque dichos puntos podrían ser considerados como ruido o como puntos espurios.

Se presentaron dos ejemplos para probar el algoritmo. En la mayoría de los casos los resultados dependen, en gran medida, de la posición de los primeros centros generados de manera aleatoria.

Aunque los resultados de los modelos presentados no fueron los mejores, en este trabajo se dio un primer paso para desarrollar modelos reconocimiento de patrones utilizando el álgebra geométrica. Si bien, los modelos presentados no pueden rivalizar con modelos de reconocimiento de patrones más sofisticados se espera que más adelante se logren mejorar combinándolos con otras ideas.

La idea es desarrollar nuevos modelos de clasificación (o reconocimiento) de patrones, no necesariamente memorias asociativas, que utilicen operadores del álgebra geométrica para su funcionamiento.

6.2 Aportaciones

El trabajo de tesis aporta las siguientes contribuciones a la ciencia, en particular en el área de la inteligencia artificial.

- Un nuevo modelo de memorias asociativas.
- Un algoritmo de clasificación que utiliza superficies de separación esféricas.
- Un nuevo modelo de clasificación supervisada de patrones en el marco del álgebra geométrica.
- Un nuevo modelo de clasificación no supervisada de patrones en el marco del álgebra geométrica.
- Un nuevo modelo de restauración de patrones en el marco del álgebra geométrica.

- Un modelo de memorias asociativas que funcionan ante ruido mezclado.
- Una mejora significativa al algoritmo de k-medias.
- Un algoritmo para resolver un problema de optimización por mínimos cuadrados con álgebra geométrica en el álgebra convencional.

6.3 Consideraciones para trabajos futuros

El siguiente paso es realizar una memoria asociativa de tipo hetero-asociativa, la diferencia con el modelo auto-asociativo, que se presentó en este trabajo, es que los patrones de entrada y salida son diferentes, incluso pueden ser de diferente dimensión. La idea consistiría en utilizar el producto exterior de alguna manera para almacenar el vector de salida dentro de la hiper-esfera.

Para mejorar la clasificación de patrones se puede considerar una hiper-esfera por cada patrón en el conjunto fundamental, estas hiper-esferas crecerán de manera individual hasta encontrarse con otras hiper-esferas. En caso de que las otras hiper-esferas sean de la misma clase se deben fusionar, si son de otra clase deberán dejar de crecer. La clasificación sería la misma que en el modelo mostrado en este trabajo, pero teniendo en cuenta la clase a la que pertenece cada patrón fundamental. Una ventaja de este método es que podría funcionar aún cuando las clases no sean esféricamente separables.

Además de las hiper-esferas se pueden utilizar otros objetos geométricos más complejos como elipsoides o paraboloides. O, inclusive, combinaciones entre diferentes tipos de objetos. De esta forma se podrían utilizar con conjuntos de puntos agrupados en formas irregulares, lo cual daría mejores resultados al momento de clasificar tanto en el modo supervisado como en el no supervisado.

Una forma de mejorar la restauración de patrones, en el caso de las imágenes, es hacer un pre-procesamiento al objeto antes de procesarlo por la memoria asociativa. Esto es, aplicar los filtros con los que se comparo el modelo junto con la memoria asociativa. También se puede combinar este modelo con la técnica de sub-patrones ya que ambos modelos obtuvieron los mejores resultados al hacer la comparación.

En el modo de operación no supervisado se utilizó como base el algoritmo de k-medias. Los resultados podrían mejorar notablemente si se utiliza un mejor algoritmo

como los mapas auto-organizados. Así mismo al combinar con otra forma irregular de superficie de separación en lugar de las hiper-esferas.

Una forma de disminuir el tiempo de cómputo es utilizar alguna técnica bio-inspirada para obtener el mínimo en la función de optimización. También se puede obtener mejores resultados incluyendo ambas restricciones al resolver el problema.

La clasificación presentada en este trabajo es una clasificación dura; es decir, un patrón dado pertenece o no a un clase cuando está dentro o fuera, respectivamente, de la hiper-esfera correspondiente. Sin embargo, con el cálculo del producto interior se puede tener, sin necesidad de más cálculos adicionales, una clasificación difusa usando las hiper-esferas como funciones de pertenencia en dos o más dimensiones.

Desarrollar otros modelos de “*soft-computing*” como los algoritmos genéticos, programación evolutiva, etc. pero que utilicen operadores y operaciones del álgebra geométrica. La idea sería utilizar multi-vectores como los elementos de operación básicos.

6.4 Publicaciones generadas

Este trabajo generó las siguientes publicaciones:

- R. Barrón, B. Cruz, H. Sossa y G. Laguna. «Conformal Geometric Algebra for Spherical Convex Hull Optimization.» *In Proc. of 3rd Internat. Conf. on Appl. of Geom. Algebras in Comput. Sci. and Eng., AGACSE 2008, 2008.*
- B. Cruz, R. Barron y H. Sossa. «Pattern Classification Based on Conformal Geometric Algebra and Optimization Techniques.» *MICAI 2008, Springer Verlag LNAI 5317, 2008: 273-283.*
- B. Cruz, R. Barron y H. Sossa. «Geometric Associative Memories and their Applications to Pattern Classification.» En *Geometric Algebra Computing for Computing Science and Engineering*, de Eduardo Bayro-Corrochano y G. Sheuermann. London: Springer Verlag, 2009.

- B. Cruz, H. Sossa y R. Barron. «Geometric Associative Processing Applied to Pattern Classification.» *ISNN 2009, Part II, Springer Verlag LNCS 5552*, 2009: 977-985.
- B. Cruz, R. Barrón y H. Sossa. «. «A New Unsupervised Learning for Clustering using Geometric Associative Memories.» *A Publicarse en CIARP 2009, Springer Verlag LNCS 5856*, 2009:239-246.

REFERENCIAS

Anderson, James A. «A simple Neural Network Generating an Interactive Memory.» *Mathematical Bioscience* 5 (1972): 14:197-220.

Anderson, James A., and Edward Rosenfeld. *Neurocomputing: Foundations of Research*. Cambridge, Massachusetts and Londond, England: The MIT Press, 1988.

Asuncion, A., y D. J. Newman. *UCI Machine Learning Repository*. 2007. <http://www.ics.uci.edu/~mllearn/MLRepository.html> (último acceso: 19 de November de 2008).

Banarer, Vladimir, Christian Perwass, and Gerald Sommer. "The Hypersphere Neuron." *11th European Symposium on Artificial Neural Networks*. Evere, Belgium: d-side publications, 2003. 469-474.

Barrón, Ricardo. *Memorias Asociativas y Redes Neuronales Morfológicas para la Recuperación de Patrones*. Ph.D Thesis, México, D. F.: National Institute Politechnic - Center of Computing Research, 2006.

Barrón, Ricardo, Benjamín Cruz, Humberto Sossa, y Gerardo Laguna. «Conformal Geometric Algebra for Spherical Convex Hull Optimization.» *In Proc. of 3rd Internat. Conf. on Appl. of Geom. Algebras in Comput. Sci. and Eng., AGACSE 2008*, 2008.

Bayro-Corrochano, Eduardo, and Refugio Vallejo. "Geometric Feedforwrd Neural Networks and Support Vector Machines." In *Geometric Algebra with Applications in Science and Engineering*, by Eduardo Bayro Corrochano and Garret Sobczyk, 309-325. Boston: Birkhauser, 2001.

Biswas, G., J. B. Weinberg, and Douglas H. Fisher. "Iterate: A conceptual clustering algorithm for data mining." *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 28 (1998): 100-111.

Buchholz, Sven. *A Theory of Neural Computation with Clifford Algebras*. Tesis, Kiel: Christian-Albrechts-Universität, 2005.

Buchholz, Sven, Kanta Tachibana, y Eckhard Hitzer. «Optimal Learning Rates for Clifford Neurons.» *Proceedings of ICANN 2007, Part I, LNCS 4668*, 2007: 864-873.

Carpenter, Gail A., y Stephen Grossberg. «Adaptative Resonance Theory.» Editado por Michael A. Arbib. *The Handbook of Brain Theory and Neural Networks* (The MIT Press) 2nd Edition (2003): 87-90.

Carrillo-Reid, Luis, y otros. «Encoding Network States by Striatal Cell Assemblies.» *Journal of Neurophysiology* 99, n° 3 (April 2008): 1435-1450.

Center for Earth Observation. *Landcover Classification Project*. Yale University. 20 de Julio de 2007. <http://www.yale.edu/ceo/Projects/swap/landcover/index.html> (último acceso: 28 de Septiembre de 2009).

Chinarov, Vladimir, y Michael Menzinger. «Reconstruction of noisy patterns by bistable gradient neural-like network.» *Biosystems* (Elsevier Science) 68, n° 2-3 (2003): 147-153.

Clifford, William Kingdon. "Applications of Grassmann's Extensive Algebra." *American Journal of Mathematics* 1, no. 4 (1878): 350-358.

Cover, Thomas M., and P. E. Hart. "Nearest Neighbor Pattern Classification." *IEEE Transactions on Information Theory* 13, no. Issue 1 (1967): 21-27.

Cruz, Benjamín, Humberto Sossa, and Ricardo Barrón. "A new Two Level Associative Memory for Efficient Pattern Restoration." *Neural Processing Letters* 25 (2007): 1-16.

Cruz, Benjamín, Ricardo Barrón, and Humberto Sossa. "Geometric Associative Memories and their Applications to Pattern Classification." In *Geometric Algebra Computing for Computing Science and Engineering*, by Eduardo Bayro-Corrochano and G. Sheuermann. London: Springer Verlag, 2009.

Cruz, Benjamín, Ricardo Barrón, and Humberto Sossa. "Pattern Classification Based on Conformal Geometric Algebra and Optimization Techniques." Edited by A. Gelbukh and E. F. Morales. *MICAI 2008, LNAI* (Springer-Verlag) 5317 (2008): 273-283.

Doran, Chris. «Circle and sphere blendign with conformal geometric algebra.» *eprint arXiv:cs/0310017* (ARXIV), october 2003: 1-19.

Dorst, Leo, and Daniel Fontijne. "3D Euclidean Geometry through Conformal Geometric Algebra (a GAVIEWER tutorial)." *University of Amsterdam Geometric Algebra Website*. 2005. <http://www.science.uva.nl/ga/tutorials/CGA/> (accessed Octubre 2, 2008).

Dorst, Leo, Daniel Fontijne, and Stephen Mann. *Geometric algebra for computer science: An object-oriented approach to geometry*. San Francisco, California: Elsevier INC., 2007.

Duda, R., P. Hart, and D. Stork. *Pattern Classification*. New York: Jhon Wiley & Sons, 2001.

Fukushima, Kunihiro. "Restoring partly occluded patterns: a neural network model." *Neural Networks* (Elsevier Science) 18, no. 1 (2005): 33-43.

Gonzalez, Rafael C., and Richard E. Woods. *Digital Image Processing*. Third Edition. Upper Saddle River, New Jersey: Pearson Prentice Hall, 2008.

Gonzalez, Rafael C., y Richard E. Woods. *Digital Image Processing*. Third Edition. Upper Saddle River, New Jersey: Pearson Prentice Hall, 2008.

Grassmann, Hermann. "Die lineare Ausdehnungslehre." In *English Translation: A new branch of mathematics*, by Lloyd Kannenberg. Leipzig: Wiegand: Chicago: Open Court, 1844.

Hebb, Donald Olding. "Introduction and Chapter 4, The first stage of perception: growth of the assembly." In *The Organization of Behavior*, by Donald Olding Hebb, xi-xix, 60-78. New York: Wiley, 1949.

Hestenes, David. "Oersted Medal Lecture2002: Reforming the mathematical language of physics." *American Journal of Physics* 73, no. 2 (Febrero 2003): 104-121.

Hestenes, David. «Old Wine in New Bottles.» En *Geometric Algebra: A Geometric Approach to Computer Vision, Quantum and Neural Computing, Robotics, and Engineering*, de Eduardo Bayro-Corrochano y Garret Sobczyk, 498-520. Boston: Birkhauser, 2001.

Hestenes, David. «Unified Language for Mathematics and Physics.» En *Clifford Algebras and their Applications in Mathematical Physics.*, de J.S.R. Chisholm y A. K. (Eds) Commons, 1-23. Boston: Reidel, Dordrecht, 1986.

Hestenes, David. "Universal Geometric Algebra." Edited by Simon Stevid. *A Quarterly Journal of Pure and Applied Mathematics* 62, no. 3-4 (September - December 1988).

Hestenes, David, and Garret Eugene Sobczyk. *Clifford Algebra to Geometric Calculus*. Kluwer: Springer Verlag, 1984.

Hildenbrand, Dietmar. *Geoemtric Computing in Computer Graphics using Confromal Geoemtric Algebra*. Tutorial, TU Darmstadt, Alemania: Interactive Graphics Systems Group , 2005.

Hitzer, Eckhard. "Euclidean Geometric Objects in the Clifford Geometric Algebra of {Origin, 3-Space, Infinity}." *Bulletin of the Belgian Mathematical Society* 11, no. 5 (2004): 653-662.

Hitzer, Eckhard. «Geometric Algebra - Leibnitz' Dream.» 2003.

Hopfield, John Joseph. "Neural Networks and physicals systems with emergent collective computational abilities." *Proceedings of the National Academy of Sciences* C-79 (1982): 2554-2558.

Jain, A. "Data Clustering: 50 years Beyond K-means." *Lecture Notes in Computer Science* (Springer Verlag) 5211, no. 1 (2008): 3-4.

Jardine, N., and R. Sibson. "The construction of hierarchic and non-hierarchic classifications." *The Computer Journal* 11 (1968): 177.

Jensen, J. R. *Introductory Digital Image Processing-A Remote Sensing Perspective*. New Jersey: Prentice Hall, Inc., 1996.

Kaymak, U., and M. Setnes. *Extended Fuzzy Clustering Algorithms*. Research Paper, Erasmus University, Rotterdam School of Management: Erasmus Research Institute of Management (ERIM), 2000.

Kishan, Mehrotra, K. Mohan Chilukuri, and Ranka Sanjay. *Elements Of Artificial Neural Network*. Cambridge: The MIT Press, 1997.

Kohonen, Teuvo. "Correlation Matrix Memories." *IEEE Transactions on Computer C-21*, no. 4 (1972): 353-359.

—. *Self-Organizing Maps*. Third Extended Edition. Vol. 30. Berlin, Heidelberg, New York: Springer Series in Information Sciences, 2001.

Kosko, Bart. "Bidirectional Associative Memories." *Transactions On Systems, Man , and Cybernetics* 18, no. 1 (1988): 49-60.

Li, Hongbo, David Hestenes, and Alyn Rockwood. "Generalized Homogeneous Coordinates for Computational Geometry." *Geometric Computing with Clifford Algebras*, 2001.

Lipson, H., and H. T. Siegelmann. "Clustering irregular shapes using high-order neurons." *Neural Computation* 12, no. 10 (2000): 2331-2353.

McCulloch, Warren, and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." *Bulletin of Mathematical Biophysics* 5 (1943): 115-133.

McQueen, J. "Some Methods for Classification and Analysis of Multivariate Observations." *Proceedings of 5-th Berkeley Symposium on Mathematical Statics and Probabiliy* (University of California Press) 1 (1967): 281-297.

Minsky, Marvin Lee, and Seymour A. Papert. *Perceptrons*. Cambridge: MIT Press, 1988.

Nakano, Kaoru. "Associatron, a model or associative memory." *IEEE Transactions on Systems, Man, and Cybernetics SMC-12*, 1972: 380-388.

Perwass, Christian, and Dietmar Hildebrand. *Aspects of Geometric Algebra in Euclidean, Projective and Conformal Space*. Technical Report Number 0310, Christian-

Albrechts-Universität zu Kiel; Institut für Informatik und Praktische Mathematik, 2004.

Perwass, Christian, Vladimir Banarer, and Gerard Sommer. "Spherical Decision Surfaces Using Conformal Modelling." Edited by Bernard Michaelis and Gerald Krell. *DAGM-Symposium*. Magdeburg, Alemania: Springer-Lecture Notes in Computer Science, 2003. 9-16.

Ritter, Gerhard X. "A New Auto-associative memory based on lattice algebra." *Proc. 9th CIARP'04*, 2004: 148-155.

Ritter, Gerhard X., Dong Li, and Joseph N. Wilson. "Image Algebra and its Relationship to Neural Networks." *Proc. SPIE, Aerospace Pattern Recognition 1098* (1989): 90-101.

Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological Review*, 1958: 386-408.

Rumelhart, David E., Geoffrey E. Hinton, y Ronald J. Williams. «Learning representations by back-propagating errors.» *Nature* 323 (1986): 533-536.

Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, no. 8 (2000): 888-905.

Sossa, Humberto, Ricardo Barrón, and Roberto A. Vázquez. "New Associative Memories to Recall Real-Valued Patterns." Edited by Alberto Sanfeliu, José Francisco Martínez Trinidad and Jesús Ariel Carrasco-Ochoa. *Lecture Notes in Computer Science (MICAI 2005: Advances in Artificial Intelligence)* 3287 (October 2004): 195-202.

Sossa, Juan Humberto, and Ricardo Barrón. "New Associative Model for Pattern Recall in the presence of Mixed Noise." *Singal and Image Processing*, 2003: 485-490 .

Steinbuch, Karl. "die Lernmatrix." *Kybernetik* 1, no. 1 (1961): 26-45.

Sussner, Peter. "Observations on Morphological Associative Memories and the kernel method." *Neurocomputing*, no. 31 (2000): 167-183.

Vázquez, Roberto A., and Humberto Sossa. "A New Associative Model with Dynamical Synapses." *Neural Processing Letters* 28 (2008): 189-207.

Vázquez, Roberto Antonio, y Humberto Sossa. «A Bidirectional Heteroassociative Memory for True-Color Patterns.» *Neural Processing Letters*, 2008: DOI: 10.1007/s11063-008-9086-9.

Villegas-Cortez, Juan, Juan Humberto Sossa-A, Carlos Avilez-Cruz, and Gustavo Olague-Caballero. "Automatic Synthesis of Associative Memories by Genetic Programming, a first approach." *Research in Computer Science. Advances in Computer Science and Engineering* 42 (2009): 91-102.

Werbos, Paul. *The Roots of Backpropagation*. Ph.D. Thesis, Cambridge, Massachussets: Harvard University, 1974.

Yáñez, Cornelio, and Juan Luis Díaz de León. *Introducción a las Memorias Asociativas*. México: Research in Computing Science, 2003.

APÉNDICE

A. CÓDIGOS

A continuación se presentan los códigos de las funciones de las memorias asociativas geométricas. Estos códigos se presentan en sintaxis de MatLab, por ser más entendible matemáticamente y porque es más sencillo de implementar en la práctica.

Clasificación supervisada

La fase de aprendizaje de una memoria asociativa geométrica para el modo de clasificación consiste en construir una hiper-esfera óptima que cubra a los patrones de una clase específica y los patrones de las demás clases estén fuera.

La función *separa* construye una esfera de separación donde los patrones dados por la variable p están dentro de la hiper-esfera y los patrones dados por q están fuera de ella. La función se basa en los desarrollos mostrados en la sección **¡Error! No se encuentra el origen de la referencia..**

```
function [c,r] = separa(p,q)
% [c,r]=separa(p,q)
% Dados los puntos interiores p y exteriores q de una hiper-esfera,
% se construye la hiper-esfera óptima de separación, en este caso el
% arreglo de puntos exteriores puede ser vacío, por lo que la
% superficie que se genera es una cubierta.
% Se usa la función del toolbox de optimización quadprog.
% por lo que se construyen las matrices h, a y los vectores f, b.
% Se regresan el centro y el radio de la hiper-esfera generada
[n,d] = size(p); % n - núm. de patrones en p;
```

```

                % d- dimensión de los patrones
m = size(q , 1); % m- núm. de patrones en q;
% Se construyen h y f.
h = [q ; p];
h(:, d + 1) = -ones(m + n , 1);
f = -h;
for i = 1 : m + n
    f(i,:) = sum(h(i , 1 : d) .* h(i , 1 : d)) * f(i , :);
end
f = sum(f)';
h = h' * h;
% Se construyen a y b.
a0 = []; % Puntos fuera de la esfera
b0 = []; % Condición para puntos fuera de la esfera
if m > 0
    b0 = 0;
    a0 = [q , -ones(m , 1)];
    for j = 1 : d
        b0 = b0 + q(:, j) .* q(:, j);
    end
    b0 = b0 / 2 - 1.0e-10; %Valor de e
end
a1 = -[p , -ones(n , 1)]; % Puntos dentro de la esfera
b1 = 0; % Condición para puntos dentro de la esfera
for j = 1 : d
    b1 = b1 + p(:, j) .* p(:, j);
end
b1 = b1 / (-2);
a = [a1 ; a0];
b = [b1 ; b0];
% Con los parametros definidos se invoca a quadprog.
s = quadprog(2 * h , f , a , b);
% Se extraen el centro y el radio del círculo optimo.
c = s(1 : d);
r = sqrt(sum(s(1 : d) .* s(1 : d)) - 2 * s(d + 1));

```

La función *EntrenaGam* realiza la fase de entrenamiento de las memorias asociativas geométricas, como se describe en la sección **Error! No se encuentra el origen de la referencia.**

```
function S = EntrenaGAM(P)
% Función Memoria=EntrenaGAM(Set_Patrones)
%
%           | Patron-1, clase |
%           | Patron-2, clase |
% Set_Patrones = |   ...   ,   ...   |
%           | Patron-K, clase |
%
% Patron-i = [Rasgo1, Rasgo2, ... , RasgoN]
%
[M , N] = size(P); % N - Tamaño del patron, M - Numero de Patrones
C = max(P(: , N)); % Obtener el número de clases
N = N - 1;          % Se elimina el atributo Clase del patrón
S = [];
for i = 1 : C %Recorrer para encontrar los patrones de la misma clase
    Ti = []; % Guarda los patrones que están dentro de la esfera
    To = []; % Guarda los patrones que están fuera de la esfera
    k = 1;
    j = 1;
    for m = 1 n M          % Recorrer todos los patrones
        if P(m, N + 1) == i % En caso de encontrar los de la clase i
            Ti(k , :) = P(m , 1 : N); % Guardarlos en Ti
            k = k + 1;
        else                % En caso de que no sean de la clase i
            To(j , :) = P(m , 1 : N); % Guardarlos en To
            j = j + 1;
        end
    end
end
[c , r] = separa(Ti, To); %#ok<NOPRT> % Obtener el centro y el
                                %radio de la esfera de separación
S(i , :) = [c', -r * r / 2 + c' * c / 2 , 1]; %#ok<AGROW>
                                % Almacena la esfera optima en el arreglo S
end
```

La función *Interior* calcula el producto interior entre dos objetos en álgebra geométrica conforme.

```
function d = Interior(P1 , P2)
% Función int=Interior(Pat1,Pat2)
%
% Aplica el producto interior entre dos conjuntos de vectores
% El resultado es un vector de tamaño max(size(P1,1), size(P2,1))
% Los vectores deben estar en el espacio conforme
%
m1 = size(P1 , 1); % Número de vectores en P1
m2 = size(P2 , 1); % Número de vectores en P2
d = zeros(m1 , m2); % Vector de salida

n = size(P1 , 2) - 2; % dimensión de los patrones
for k = 1 : m1
    for j = 1 : m2
        for i = 1 : n
            % Producto interior tradicional entre los vectores
            d(k , j) = d(k , j) + P1(k , i) * P2(j , i);
        end
        % Calcular los valores de las componentes e1f y e0
        d(k , j) = d(k , j) - P1(k , n + 1) * P2(j , n + 2)
            - P1(k , n + 2) * P2(j , n + 1);
    end
end
end
```

La función *Euclid2Conf* convierte el patrón euclidiano a su correspondiente notación conforme, ver ecuación .

```
function P = Euclid2Conf(p)
% Funcion P = Euclid2Conf(p)
% cambia p a su representación conforme P
%   | p1 |
%   | p2 |
% p = | .. |
%   | pm |
% Los puntos a cambiar a su representación conforme
```

```

% Donde
%  $p(i) = |r_1, r_2, \dots, r_n|$ 
%
%      | P1 |
%      | P2 |
% P = | .. |
%      | .. |
%      | Pm |
% Los puntos en representación conforme
% Donde
%  $P(i) = |r_1, r_2, \dots, r_n, r_{n+1}, r_{n+2}|$ 
%
[m , n] = size(p);
for i = 1 : m          % Recorre el arreglo de puntos p
    P(i , 1 : n) = p(i , 1 : n); % La parte euclidiana es la misma
    P(i , n + 1) = p(i , 1 : n) * p(i , 1 : n)'; % Para einf
    P(i , n + 2) = 1;          % Para e0
end

```

La función *ClasificaGam* es la encargada de decidir si un patrón dado pertenece a alguna de las clases definidas por las hiper-esferas construidas en la fase de entrenamiento, regresa el índice al que pertenece un patrón dado.

```

function c = ClasificaGAM(M , p)
% Función clase=ReconoceGAM(MemoriaAG,PatronEntrada)
% Fase de clasificación de una MAG
%
%      | S1 |
%      | S2 |
% MemoriaAG= | S3 |
%      | ... |
%      | Sk |
%
% Donde: MemoriaAg(i) = [S1, S2, S3, ... Sn+1, Sn+2]
%      PatronEntrada = [P1, P2, P3, ... Pn]
%      n = La dimensión del espacio de trabajo
%
C = size(M , 1); % C - Numero de clases % N - Tamaño del patrón

```

```

N = size(p , 2);
radios = zeros(N , 1); % Inicializa la matriz con radios 0
Mayor = -5000;
c = 0;
%Convierte Patron a su representación conforme
p = Euclid2Conf(p)
clases = [];
%Calcular los radios de las esferas
for i = 1 : C
    radios(i) = M(i, 1 + N) - M(I , 1 : N) * M(i, 1 : N)' / 2;
    % El radio -r/2 = S4-(P*P)/2
end
% obtener los productos interiores entre las esferas y el patrón

for i = 1 : C
    d = Interior(M(i , :), p); % Calcula el producto interior entre
    % la esfera y el patrón conforme
    if d < 0 % Verifica que el patrón esté dentro de la esfera
        clases(i) = -Inf;
        % Si no está dentro de la esfera, no utilizarlo
    else
        clases(i) = d + radios(i);
        % #ok<AGROW> Calcula la distancia hacia el centro de
        % la esfera en signo negativo
    end
    if clases(i) > Mayor
        % Obtiene el valor más cercano al centro de la esfera
        Mayor = clases(i);
        c=i; % La clase a la que pertenece
    end
end
end

```

Restauración de patrones

La fase de aprendizaje de una memoria asociativa geométrica para el modo de restauración de patrones consiste en construir una hiper-esfera óptima que cubra a

cada uno de los patrones, con la condición de que dos patrones no estén dentro de una misma hiper-esfera.

La función *separaR* construye una esfera de separación donde el patrón p es el centro de de una hiper-esfera óptima con la condición de que los patrones en la variable q están fuera de esa hiper-esfera. La función se basa en los desarrollos mostrados en la sección **¡Error! No se encuentra el origen de la referencia..**

```
function [c , r] = separaR(p , q)
% [c,r]=separacion1(p,q)
% Dados el punto interior p y los puntos exteriores q de una hiper-
% esfera, se construye la hiper-esfera óptima de separación, donde el
% centro de la esfera es, precisamente, el punto p.
% Se usa la función del toolbox de optimización
% quadprog(h,f,a,b,[],[]),
% por lo que se construyen los valores de h, a, f, b.
d = size(p , 2); % d dimensión de los patrones
m = size(q , 1); % m número de patrones en q
p0 = [p ; q];
% Se construyen h y f.
% h = m - Numero total de puntos
h = m + 1;
% f = sum (2(pj.pi) +(pj)^2)
f = 0;
for i = 1 : m + 1
    f = f + 2 * p0(i , :) * p(1 , :) + p0(i , :) * p0(i , :);
end
% Se construyen a y b.
% Construcción de a
a = ones(1 , m + 1);
a(1) = 2; % Para el punto dentro de la esfera
for j = 2 : m + 1 % Para los puntos fuera de la esfera
    a(j) = -2;
end
% Construcción de b
b = ones(1 , m + 1);
b(1) = p(1 , :) * p(1 , :);
% Para el punto dentro de la esfera
```

```

for j = 2 : m + 1      % Para los puntos fuera de la esfera
    b(j) = q(j-1 , :) * q(j-1 , :)' - 2* q(j-1, :) * p(1, :)' - 1.0e-10;
end
% Con los parámetros definidos se invoca a quadprog.
s = quadprog(2 * h, f, a' , b);
% Se extraen el centro y el radio de la hiper-esfera óptima.
% El resultado es S(n+1), de ahí se extrae el radio, el centro es p
c = p(1 , :);
r = sqrt(p(1 , :) * p(1 , :)' - 2 * s);

```

La función *EntrenaGamR* realiza la fase de entrenamiento de las memorias asociativas geométricas para restauración, como se describe en la sección **¡Error! No se encuentra el origen de la referencia..**

```

function S = EntrenaGAMR(P)
% Función Memoria=EntrenaGAMR(Set_Patrones)
% Entrenamiento de una MAG para Restauración
%
%          | Patron-1 |
%          | Patron-2 |
% Set_Patrones = | ... |
%          | ... |
%          | Patron-K |
% Patron-i = [Rasgo1, Rasgo2, ... , RasgoN]
[M , N] = size(P); % N - Tamaño del patrón, M - Numero de Patrones
S = [];
R = [];
for i = 1 : M      % Recorrer todos los patrones a clasificar
    Ti = [];      % Guarda el patrón que está dentro de la esfera
    To = [];      % Guarda los patrones que están fuera de la esfera
    m = 1;
    for j = 1 : M      % Recorrer todos los demás patrones
        if i == j      % En caso de ser el patrón a clasificar
            Ti = P(j , 1 : N); % Guarda el patrón en Ti
        else            % En caso de que no sean de la clase i
            To(m , : ) = P(j , 1 : N); %#ok<AGROW> % Guardarlos en To
            m = m + 1;
        end
    end
end

```

```

end
[c , r] = separaR(Ti , To); %#ok<NOPRT> % Obtener el centro y el
                                % radio de la esfera de separación
S4 = (c * c' - r * r) / 2;
S(i , :) = [c, S4 , 1]; %#ok<AGROW>
end

```

La función *ReconoceGam* es la encargada de restaurar un patrón dado, regresando el centro de la hiper-esfera que lo cubre. A diferencia de su contraparte de clasificación, esta función regresa el centro de la hiper-esfera como el patrón restaurado.

```

function p = ReconoceGam(M , p)
% Función PatronSalida= ReconoceGam(MemoriaAG,Patronentrada)
% Fase de restauración de una MAG
%
%           | S1 |
%           | S2 |
% MemoriaAG= | S3 |
%           | ... |
%           | Sk |
% Donde: S(i) = [S1, S2, S3, ... Sn+1, Sn+2]
%   Patron = [P1, P2, P3, ... Pn]
%           n = La dimensión del espacio de trabajo
C = size(M , 1); % C - Numero de clases % N - Tamaño del patrón
N = size(p , 2);
radios = zeros(N , 1); % Inicializa la matriz con radios 0
Mayor = -5000;
c = 0;
p = Euclid2Conf(p) %Convierte p a su representación conforme
clases = [];
%Calcular los radios de las esferas
for i = 1 : C
    radios(i) = M(i, 1 + N) - M(i, 1 : N) * M(i, 1 : N)' / 2;
                % El radio -r/2 = S4-(P*P)/2
end
% obtener los productos interiores entre las esferas y el patrón
for i = 1 : C
    d = Interior(M(i , :), p); % Calcula el producto interior entre

```

```

                                % la esfera y el patrón conforme
if d < 0 % Verifica que el patrón esté dentro de la esfera
    clases(i) = -Inf;
    % Si no está dentro de la esfera, no utilizarlo
else
    clases(i) = d + radios(i);
    % #ok<AGROW> Calcula la distancia hacia el centro de
    % la esfera en signo negativo
end
if clases(i) > Mayor
    % Obtiene el valor más cercano al centro de la esfera
    Mayor = clases(i);
    c=i; % La esfera a la que pertenece
end
end
p = M(c , 1 : N); % Se regresa el centro de la esfera

```

Clasificación no supervisada

El algoritmo de clasificación no supervisada o agrupamiento es una sola función, se basa en el algoritmo presentado en la sección **¡Error! No se encuentra el origen de la referencia..**

La función *GAM_Cluster* utiliza una memoria asociativa geométrica para agrupar una serie de punto sin etiquetar modificando el algoritmo tradicional de *k*-medias.

```

function [S , j , C] = GAM_cluster(p , K)
% Funcion Clases=GAM_cluster(Puntos,K)
%
% Memorias Asociativas Geométricas con aprendizaje no supervisado,
% utiliza las características de las MAG para obtener la clase más
% cercana a la que pertenecen los puntos y usa un convex hull para
% agrupar los puntos en las agrupaciones correspondientes.
%           | P1 |
%           | P2 |
% Puntos = | . |
%           | . |

```

```

%          | Pm |
% Conjunto de puntos a ser agrupados, donde
% P(i) = |r1, r2, ..., rn|
% K -> El número de agrupaciones a generar
% Clases -> Los índices de las clases que corresponden a cada punto
%
[m , n] = size(p);
% Seleccionar el valor máximo para los vectores aleatorios
x = max(max(abs(p)));
% Generar k centroides aleatorios:
C = rand(K , n) * x * 2 - x;
% Formar la MAG con los centroides:
M = Euclid2Conf(C);
P = Euclid2Conf(p);
% Determinar la clase a la que pertenecen cada punto de acuerdo a la
% distancia a los centros
for j = 1 : 1000 %Número máximo de iteraciones
    clase = Interior(M , P);
    [temp, c] = max(clase);
    for i = 1 : m
        r(i , 1 : n) = p(i , 1 : n); %#ok<AGROW>
        r(i, n + 1) = c(i); %#ok<AGROW>
    end
    S = EntrenaGAM(r);
    if S == M      % Si hay convergencia
        break;    % Condición para terminar
    else
        M = S;    % Reemplazar los nuevos centroides
    end
end
end

```