



15889-121  
TESIS-01



CINVESTAV-IPN  
Biblioteca de Ingeniería Eléctrica



FB000014031

CENTRO DE INVESTISACION Y DE  
ESTUDIOS AVANZADOS DEL  
I. P. N.  
BIBLIOTECA  
INGENIERIA ELECTRICA



**Centro de Investigación y de Estudios Avanzados  
del I.P.N.**

Departamento de Ingeniería Eléctrica  
Sección Computación

**Representación Isoanalógica de  
Objetos N Dimensionales**

**TESIS QUE PRESENTA:  
José de Jesús Lavalle Martínez**

**PARA OBTENER EL GRADO DE:  
Maestro en Ciencias con Especialidad en Ingeniería Eléctrica**

**DIRECTORES DE TESIS:  
Dr. José Ruíz Shulcloper  
Dr. Sergio V. Chapa Vergara**

México D.F., Noviembre de 2000

CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL  
I.P.N.  
BIBLIOTECA  
INGENIERIA ELECTRICA





XM

CLASIF.	00.26
ADONS.	B1 15889
FECHA:	27 - April - 2001
PROCED:	Tesis - 2001
\$	.....

# Índice General

<b>Resumen</b>	<b>13</b>
<b>Introducción</b>	<b>15</b>
<b>1 Métodos Previos</b>	<b>19</b>
1.1 Historia del Concepto de Representación	19
1.1.1 Fisher	19
1.1.2 Sammon	20
1.1.3 Chang & Lee	21
1.1.4 Lee, Slagle & Blum	22
1.1.5 Niemann & Weiss	23
1.1.6 Wismath, Soong & Akl	24
1.1.7 Kakusho & Mizoguchi	25
1.1.8 Siedlecki, Siedlecka & Sklansky	26
1.2 Conceptos Preliminares	28
1.3 Clasificación de los Métodos Previos	30
1.3.1 Métodos Restrictivos	31
1.3.2 Métodos Generales	31
1.3.3 Métodos de Alto Riesgo	32
1.4 Minimización No Lineal Sin Restricciones	34
1.4.1 Gradiente Conjugado	34
1.4.2 Quasi-Newton	35
1.4.3 Newton	36
<b>2 Propuesta</b>	<b>37</b>
2.1 Representación Isoanalógica	37
2.2 Un Primer Acercamiento	40
2.3 Propuesta Mejorada	42

<b>3</b>	<b>Implantación Computacional</b>	<b>53</b>
3.1	Estructuras de Datos . . . . .	53
3.2	Funciones más Importantes . . . . .	54
3.3	Dependencias Funcionales . . . . .	57
3.4	Herramientas Utilizadas . . . . .	57
3.5	Ejemplos . . . . .	58
	<b>Conclusiones</b>	<b>67</b>
	<b>Trabajo Futuro</b>	<b>69</b>
	<b>Bibliografía</b>	<b>71</b>

# Índice de Figuras

2.1	Representaciones obtenidas al fijar $P_1$ y $P_2$ ; $P_1$ y $P_3$ ; $P_1$ y $P_4$ , para la matriz de disimilaridades (2.10). . . . .	44
2.2	Representaciones obtenidas al fijar $P_2$ y $P_3$ ; $P_2$ y $P_4$ ; $P_3$ y $P_4$ , para la matriz de disimilaridades (2.10). . . . .	45
2.3	Representación obtenida al fijar $P_1$ y $P_4$ , para la matriz de disimilaridades (2.11). . . . .	47
2.4	Representación obtenida al fijar $P_3$ y $P_4$ , para la matriz de disimilaridades (2.11). . . . .	47
2.5	Representación obtenida al fijar $P_1$ y $P_3$ , para la matriz de disimilaridades (2.11). . . . .	48
2.6	Representación obtenida al fijar $P_2$ y $P_4$ , para la matriz de disimilaridades (2.11). . . . .	48
2.7	Representación obtenida al fijar $P_1$ y $P_2$ , para la matriz de disimilaridades (2.11). . . . .	49
2.8	Representación obtenida al fijar $P_2$ y $P_3$ , para la matriz de disimilaridades (2.11). . . . .	49
3.1	Representación isoanalógica obtenida por nuestro algoritmo para el ejemplo <i>ISAS</i> . . . . .	59
3.2	Representación obtenida al fijar $P_1$ y $P_{55}$ , para el ejemplo <i>ISAS</i> . . . . .	60
3.3	Representación obtenida al fijar $P_{55}$ y $P_{109}$ , para el ejemplo <i>ISAS</i> . . . . .	61
3.4	Representación obtenida al fijar $P_{36}$ y $P_{105}$ , para el ejemplo <i>ISAS</i> . . . . .	62
3.5	Representación obtenida al fijar $P_1$ y $P_2$ , para el ejemplo <i>ISAS</i> . . . . .	63
3.6	Representación obtenida al fijar $P_{108}$ y $P_{109}$ , para el ejemplo <i>ISAS</i> . . . . .	64
3.7	Representación obtenida al fijar $P_{37}$ y $P_{50}$ , para el ejemplo <i>ISAS</i> . . . . .	65

3.8	Representación obtenida al fijar $P_1$ y $P_{109}$ , para el ejemplo <i>ISAS</i> . . . . .	66
-----	---	----

# Índice de Tablas

3.1	Resumen de los experimentos hechos con el ejemplo <i>ISAS</i> . . .	61
-----	---	----

CENTRO DE INVESTIGACION Y DE  
ESTUDIOS AVANZADOS DEL  
I. P. N.  
BIBLIOTECA  
INGENIERIA ELECTRICAS



# Dedicatoria

A María Zarrabal Méndez† por heredarnos su carácter.

A Guillermina Martínez Benítez por enseñarme a deducir propiedades no triviales.

A Benito Lavalle Zarrabal por convencerme que lo más importante es el conocimiento.

A Alejandro Reyes Cristian† por mostrarme lo que es un computólogo.

A María Esther Guevara Cruz y Saturnino Job Morales Escobar por tanto que hemos caminado juntos.

A Emma Isabel Altamirano López por tanto que nos falta caminar.

A Isabel e Isadora Lavalle Altamirano porque son mis hijas.

A Selene Lavalle Martínez por insinuarme la semántica de ser padre.



# Agradecimientos

Soy una persona muy afortunada y me ha costado mucho trabajo escribir esta parte del documento, ya que hay demasiadas personas a las que debo agradecer y poco espacio para hacerlo, siempre está presente además, el temor de olvidar a alguien, sin embargo debo de escribir al menos lo siguiente.

Mi ineludible agradecimiento a mi familia, todos ellos, hermana(o)s, cuñadas y sobrina(o)s, somos tantos que sería inoportuno escribir los nombres, sobre todo que es obvio quienes somos.

A Ernesto Germán Zapata Ledezma, Héctor Vázquez Briones, Harold V. McIntosh, Gerardo Cisneros Staionovsky, Guillermo Morales Luna y Sergio V. Chapa Vergara; por haber contribuido especialmente en mi formación.

A Rodolfo Reyes Sánchez, Francisco Javier Albores Velasco e Isidro Romero Médina, por su apoyo desinteresado en mi desarrollo.

A Mädy Fuerbringer Bermeo y Héctor Jiménez Salazar, MIS MAESTROS.

Ahora le toca a la gente de Cuba, José Ramón Pico y Teresita Fernández; mi hermano y madre Cubana. Manuel Lazo Cortés mi amigo de siempre y a José Ruiz Shulcloper que usó al Reconocimiento de Patrones para encontrar un poco de todo y mucho de nada.

A mi compadre sin compromiso Jesús García Fernández siempre en pie de lucha.

A los miembros del jurado, por sus comentarios que ayudaron a darle brillo al documento, especialmente al Dr. Arturo Díaz Pérez.

A mis tesisistas, por haberse atrevido a ser.

Finalmente pero no menos importante a Sofia Reza Cruz, por su paciencia, eficacia, eficiencia y solidaridad.



# Resumen

Este trabajo trata con el problema de representar objetos de dimensión  $n$  en un subconjunto de  $\mathbb{R}^2$ , luego de hacer un estudio del estado del arte sobre este tema, se encontró que los algoritmos existentes para tal propósito eran inadecuados, al menos desde la perspectiva del Enfoque Lógico-Combinatorio al Reconocimiento de Patrones.

Como resultado de este estudio surge una nueva clasificación de los algoritmos previos, además la necesidad de formular una definición de representación más general, a la que hemos llamado *isoanalógica*.

Se desarrolla con detalle un algoritmo que satisface tal definición, y que, a pesar de ser más general que los algoritmos previos, necesita minimizar sólo la mitad de las variables con respecto a otros algoritmos. No obstante el algoritmo fue desarrollado bajo la mirada del Enfoque Lógico-Combinatorio al Reconocimiento de Patrones, se puede aplicar a resultados obtenidos por otros enfoques.

Finalmente se da una implantación computacional del algoritmo y se presentan experimentos.



# Introducción

En términos muy generales, cuando se modelan matemáticamente fenómenos del mundo real, la información que se tiene de los objetos en estudio se codifica en un  $n$ -uplo, además el experto en dichos fenómenos tiene una noción de la relación que éstos guardan entre sí, o al menos de cómo obtener dicha relación. Dependiendo de la aplicación la relación puede ser: distancia, cercanía, semejanza, parecido, analogía, similaridad, lejanía, disimilaridad, entre otras.

Así, este trabajo enfrenta el problema de representar objetos  $n$  dimensionales en 2 dimensiones. La idea es contar con un método que nos permita “visualizar” en un plano la relación que guardan los objetos en la dimensión  $n$ . De tal manera que entre mayor sea la disimilaridad de dos objetos, mayor debe ser la distancia de los puntos que los representan en el plano.

Note que el problema surge desde dos perspectivas igualmente legítimas. Por un lado el experto quiere tener una imagen del modelo que se ha construido, para constatar que se “corresponde” (al menos intuitivamente) con lo que él sabe del fenómeno en consideración. Desde el lado del modelador, dicha imagen le permite observar la “estructura” que guardan los objetos modelados, para poder decidir qué método(s) de análisis de información utilizar. Por ejemplo, la imagen puede revelar: el número de agrupamientos que conforman los objetos modelados; que tan disimilares son los objetos de cada agrupamiento o que tan disimilares son los agrupamientos.

En la práctica profesional de muchas ciencias aplicadas, como por ejemplo las Geociencias, la visualización de los datos ha cobrado una importancia capital.

Entre otras cosas motivada por el incremento sustancial de los volúmenes de datos que se generan en casi todas las ramas del conocimiento, fruto de los procesos de automatización que se realizan en todo el mundo en todas las ramas del saber.

Entre los efectos prácticos de esta visualización se puede subrayar la posibilidad que le brinda al usuario de confirmar o reajustar su propio modelo de la realidad, tomar decisiones que complementan las sugeridas por algún algoritmo, etc. El problema de la representación en el plano de objetos *n*-dimensionales es uno de esos problemas de visualización.

Varios métodos se han propuesto para resolver este problema [Sam69], [CL73], [OF73], [L+77], [NW79], [W+81], [KM83], [Gow84], [S+88b], [S+88a], [TM90], pero al menos desde el enfoque Lógico-Combinatorio al Reconocimiento de Patrones (*ELCRP*) [RS+95], que tiene como principales características que los objetos se pueden codificar por medio de variables tanto cuantitativas como cualitativas e incluso se puede trabajar con codificaciones incompletas de los objetos en estudio. Estos métodos poseen una serie de inconvenientes que evitan hacer un uso generalizado de ellos.

Unos exigen que los objetos se codifiquen como vectores en  $\mathbb{R}^n$ , lo cual puede ser útil en aplicaciones de Ciencias Exactas e Ingeniería, pero insuficiente en Ciencias Sociales o de la Salud, donde frecuentemente se trabaja además con variables de tipo cualitativo, por ejemplo.

Otros obtienen diferentes representaciones para el mismo modelo, el problema con estos métodos es que en lugar de ser una ayuda en el proceso de modelación, pueden ser causales de errores y confusión. Por ejemplo, una representación pudiera mostrar que un modelo contiene tres agrupamientos y otra representación mostrar que el mismo modelo tiene cuatro agrupamientos; o que, bajo una representación  $k$  objetos pertenecen a cierto agrupamiento y con otra representación esos mismos  $k$  objetos pertenecen a otro(s) agrupamiento(s). Además generarían el problema de saber cuál es la mejor representación.

Lo anterior motivó a proponer una definición de las características que desde el *ELCRP* debe cumplir un método de representación para poder hacer un uso generalizado del mismo. En concreto, se define formalmente lo que entendemos por **Representación Isoanalógica**, por supuesto que el método que aquí se propone satisface esta definición.

El resto del documento tiene la siguiente estructura: el primer capítulo presenta cómo ha evolucionado el concepto de representación desde 1936 hasta nuestros días, también contiene conceptos propios del *ELCRP* que permitirán clasificar los métodos de representación previos a este trabajo, ahí mismo se dará un panorama de ellos y un bosquejo de los métodos de minimización no lineal sin restricciones.

En el segundo se define el concepto de Representación Isoanalógica, tam-

bién contiene el desarrollo de nuestro método.

Se da en el tercer capítulo la implantación computacional de nuestro algoritmo. Se concluye con los alcances y limitaciones del método construido y se proponen nuevas tareas a resolver.





# Capítulo 1

## Métodos Previos

En este capítulo se describe cómo han conceptualizado el problema de la representación de objetos  $n$  dimensionales los autores de algunos métodos analizados, y el lugar que ocupa la representación en el reconocimiento de patrones.

Asimismo se propone una nueva clasificación de los métodos de representación que se han desarrollado con anterioridad, bajo la perspectiva del *ELCRP* [RS<sup>+</sup>95] y de la aplicabilidad que ellos tienen.

Es pertinente también explorar las distintas técnicas de minimización no lineal sin restricciones, para elegir la más adecuada y con ella implantar nuestra propuesta de representación.

### 1.1 Historia del Concepto de Representación

Para Siedlecki et. al. [S<sup>+</sup>88b] una *operación de mapeo* es aquella que transforma un conjunto de vectores reales de dimensión  $d$  sobre un plano, y al resultado de dicha operación se le llama *mapeo*. Anotan también que es común usar el término mapeo tanto para la operación como para su resultado.

#### 1.1.1 Fisher

En [S<sup>+</sup>88b] se afirma que la primera técnica de mapeo usada ampliamente se basa en el algoritmo para la clasificación lineal de datos propuesto por R. A. Fisher en 1936, ahora conocido como discriminante de Fisher. Este método toma  $m$  vectores de dimensión  $n$  y los transforma en  $m$  vectores de

dimensión dos.

Fisher trabajaba con especialistas en taxonomía, uno de sus objetivos era clasificar flores llamadas Iris, basándose en tres especies (Iris Versicolor, Iris Virginica e Iris Setosa) conformó 150 vectores de dimensión 4 (como muestra para la clasificación), 50 vectores por cada especie distinta. Su propósito era resolver problemas de *reconocimiento de patrones supervisado*. Fisher no tenía en mente el problema de la representación de objetos. Ya que en esos años no había computadoras, mucho menos dispositivos y técnicas de computadora para visualizar información.

Note que el año en que se publica el artículo de Fisher es el mismo en que Alan M. Turing publica “*On computable numbers, with an application to the Entscheidungsproblem*”, lugar donde se expone uno de los modelos matemáticos de lo que ahora conocemos como *computadora*.

Recordemos que es hasta mediados de los cuarenta cuando se construyen las primeras computadoras (basadas en bulbos) bajo la dirección de Howard Aiken en Harvard, John von Neumann en el Instituto de Estudios Avanzados de Princeton, J. Presper Eckert y William Mauchley en la Universidad de Pensilvania, sin olvidar a Konrad Zuse<sup>1</sup> en Alemania [Tan93].

### 1.1.2 Sammon

Tres décadas después gracias al desarrollo de la electrónica, de algoritmos y programas de computadora para graficar en dos y tres dimensiones, se plantea el problema de la representación de objetos como una herramienta “visual”, por ello Sammon plantea en [Sam69].

“El problema de análisis de datos multivariados es detectar e identificar la “estructura” que está presente en una lista de  $m$  vectores de dimensión  $n$ .

Donde la palabra estructura se refiere a la relación geométrica entre subconjuntos de vectores (datos) en el espacio  $n$ . Algunos ejemplos de estructuras son agrupamientos (clusters en inglés) hiperesféricos y hiperelipsoidales, también ciertas relaciones lineales y no lineales entre los vectores de algún subconjunto.

---

<sup>1</sup>En su honor es que una de las distribuciones más robustas de Linux lleva el nombre de SuSE.

... el algoritmo se basa en mapear los  $m$  vectores de dimensión  $n$  a un espacio de dimensión menor, tal que la estructura inherente de los datos sea mantenida aproximadamente bajo el mapeo.

El interés primordial está en mapeos a espacios de dos y tres dimensiones, ya que la configuración de datos resultante puede ser fácilmente evaluada por observadores humanos.”

Sammon también hace referencia a que el primero en usar en lugar de distancia el concepto más relajado de similaridad (disimilaridad) es Shepard, cuando en 1962 publica su artículo “*The analysis of proximities: multidimensional scaling with an unknown distance function*”.

Shepard modelaba experimentos psicológicos, en los que es frecuente tener solamente una medida de similaridad (disimilaridad) entre cada par de objetos, y no un conjunto con  $m$  vectores de dimensión  $n$ .

Note que en general Sammon habla de mapear los vectores en la dimensión  $n$  a vectores en un espacio de menor dimensión, por lo que se puede identificar su propuesta con el problema general en reconocimiento de patrones conocido como *reducción de variables* (*feature reduction* en inglés).

No obstante, Sammon utilizó su algoritmo para producir mapeos en dos y tres dimensiones, con el afán de encontrar visualmente los agrupamientos que estaban inherentes en los datos, es decir, lo utilizó como un algoritmo para el *reconocimiento de patrones no supervisado*.

Es pertinente aclarar de una vez, que hemos preferido usar el término más general de *representación* (el cual definiremos en la sección (2.1)), con tal de no violar la noción de mapeo dada en [S+88b] ( $M : \mathbb{R}^n \rightarrow \mathbb{R}^2$ ), la cual es muy restrictiva por exigir como entrada vectores de dimensión  $n$  y por ofrecer como resultado vectores en el plano.

### 1.1.3 Chang & Lee

A partir de Chang & Lee [CL73] se empieza a reconocer la importancia de la representación como una técnica poderosa para agrupar objetos, basándose en la habilidad humana de interpretar imágenes en una, dos o tres dimensiones, en concreto ellos opinan que:

“Generalmente estamos de acuerdo en que los seres humanos no tienen dificultad en agrupar datos representados en una, dos o tres dimensiones, usando solamente la percepción visual.

No obstante, para datos en una dimensión mayor a tres, ya que los datos no se pueden ver, dicha habilidad perceptiva deja de existir.

Para remediar esta limitación podemos tratar de reducir la dimensión de los datos y mapearlos a un espacio de una, dos o tres dimensiones.”

Así estos autores ven la representación como un método con el cual se pueden resolver problemas de reconocimiento de patrones no supervisado, donde la decisión de cuántos grupos existen en la muestra y cuáles objetos pertenecen a cada grupo, es tomada por un observador humano.

#### 1.1.4 Lee, Slagle & Blum

Conforme va pasando el tiempo se gana experiencia y se quieren resolver aplicaciones donde los objetos no se pueden modelar via variables reales, por ejemplo Lee, Slagle & Blum [L<sup>+</sup>77] tienen la siguiente posición.

“Considere el siguiente problema. Damos un conjunto de datos y queremos saber la relación geométrica que guardan los puntos que representan a los datos.

Si los datos son de dimensión dos y numéricos, podemos encontrar dichas relaciones simplemente desplegándolos.

En otro caso, es deseable contar con alguna técnica de mapeo que preserve la relación entre los datos.

Una técnica de despliegue muy famosa es la de análisis de las componentes principales, que si bien, es excelente en muchos aspectos, tiene los siguientes inconvenientes:

1. Involucra encontrar los vectores propios de la matriz de covarianza de los datos. Si la dimensionalidad de los datos es grande, resulta muy difícil encontrar dichos vectores propios.
2. La técnica solamente trabaja con datos numéricos.

Recientemente una técnica llamada mapeo no lineal ha sido usada para desplegar datos<sup>2</sup>. El método se basa en el escalado mu-

---

<sup>2</sup>Se refieren a los algoritmos en [Sam69] y [CL73].

tidimensional de los datos, la matriz de distancias es la única información necesaria para trabajar con esta técnica.

La técnica del mapeo no lineal trata de preservar todas las distancias. Esto puede ser muy ambicioso ya que usualmente termina en el caso donde ninguna distancia se preserva exactamente.

Si el propósito del análisis de datos es detectar la existencia o no existencia de grupos (clusters), el mapeo lineal es adecuado.

Por otro lado, si estamos interesados en conocer la estructura detallada de los datos es usualmente inadecuado.

Correspondientemente, introduciremos una técnica de mapeo de datos que también se basa en el concepto de distancia.

No obstante esta técnica es distinta del mapeo no lineal en los siguientes sentidos:

1. No intenta preservar todas las distancias, en lugar de eso solamente se preserva exactamente un subconjunto de ellas.
2. Permite al usuario hacer diferentes mapeos de los mismos datos, cada mapeo enfatizando un punto de vista particular."

Si bien es cierto que es imposible que el mapeo preserve exactamente todas las distancias, también es cierto que puede ser muy engañoso un método que dé diferentes representaciones para los mismos datos.

El hecho de distintas representaciones o puntos de vista nos encara a un problema de subjetividad, note que para fines prácticos sería imposible explorar todas las posibilidades, así que no hay forma de saber cuál es la mejor representación, ni siquiera si la mejor representación está dentro de las que se hayan obtenido.

### 1.1.5 Niemann & Weiss

Es en Niemann & Weiss [NW79] donde se enuncia de una forma muy clara la relevancia que tiene el problema de la representación dentro del reconocimiento de patrones, veamos lo que ellos enfatizan.

“Un enfoque para analizar la estructura de una muestra de patrones (objetos o puntos representando datos) es mapear los patrones desde un espacio de variables de dimensión alta hacia el plano. Después de esto, los patrones pueden ser representados por puntos en el plano y su estructura se verá directamente.

Por supuesto se requiere que el mapeo preserve la estructura de los patrones tan bien como sea posible, aunque se debe tolerar cierta cantidad de distorsión de la estructura, ya que en general no es posible mapear datos de dimensión alta hacia el plano, sin que haya distorsión.

Aquellas regiones del plano donde la densidad de los puntos sea alta revelarán agrupamientos, dichas regiones indicarán los puntos que deben conformar a cada clase.

Si se da una muestra de patrones que pertenecen a un número desconocido de clases, es posible obtener el número de clases a partir del conjunto de puntos en el plano.

También es posible identificar exactamente que puntos pertenecen a una clase y convertir un proceso de aprendizaje no supervisado en supervisado.”

Cabe señalar que también proponen una metodología a seguir para aprovechar al máximo una representación, que si bien ahora está claro son ellos los primeros que lo escriben tan explícitamente.

### 1.1.6 Wismath, Soong & Akl

Una nueva aplicación para los métodos de representación surge en [W<sup>+</sup>81], los autores proponen usar a éstos para hacer reducción de variables, veamos cómo plantean el problema.

“En la fase de diseño de un clasificador para propósitos de reconocimiento de patrones, se cuenta con un conjunto de patrones  $P_1, P_2, \dots, P_l$  donde cada uno de éstos pertenece a una de varias clases  $C_1, C_2, \dots, C_r$ .

Cada patrón  $P_i$  se representa por un vector de tamaño  $N$  cuyos componentes  $p_{ij}$ ,  $j = 1, 2, \dots, N$ , son medidas tomadas al objeto

del mundo real al cual describe el patrón. Incluso se puede asumir que se conoce a qué clase pertenece cada patrón.

Usualmente no se sabe *a priori* qué medidas serán más útiles para discriminar entre las distintas clases. Lo que inevitablemente conduce a que se tomen un gran número de medidas, consecuentemente surgen serios problemas de almacenamiento y tiempo en el proceso de clasificación.

Por lo tanto es deseable seleccionar un subconjunto de medidas de tamaño  $n$  del conjunto más grande de tamaño  $N$ , donde  $n$  es mucho más pequeño que  $N$ . Las medidas seleccionadas son llamadas *variables* y la operación es llamada *selección de variables* o *reducción de la dimensionalidad*.

La idea básica para nuestra selección de variables involucra el uso de gráficas de computadora y la habilidad humana en reconocimiento de patrones para elegir interactivamente un número de variables, no necesariamente determinadas de antemano.”

Su propuesta implica hacer primero una representación bidimensional de los objetos en base a las  $N$  mediciones originales, después ir seleccionando subconjuntos de mediciones, sobre la base de éstos hacer nuevas representaciones, si se “preservan” las clases con respecto a la representación original, entonces ese subconjunto de mediciones tiene el mismo poder de discriminación que el conjunto original.

Esto a todas luces es un proceso de prueba y error, pero como muchos autores mencionan, se tiene una gran confianza en la habilidad humana para interpretar imágenes en una, dos o tres dimensiones.

### 1.1.7 Kakusho & Mizoguchi

Kakusho & Mizoguchi [KM83] observan que los métodos de representación pueden ser usados también como verificadores o evaluadores del desempeño de algoritmos de reducción de variables y de agrupamiento, como el siguiente planteamiento lo muestra.

“Al tratar con problemas del mundo real en reconocimiento de patrones, frecuentemente necesitamos extraer mucha información de un número de datos con propiedades estadísticas desconocidas.

En estos casos frecuentemente se usan métodos de análisis no paramétricos, tales como análisis de clusters y análisis de mínima dimensión.

Encontrar la dimensionalidad intrínseca de un conjunto de datos es uno de los objetos más importantes en el problema de extracción de variables y clasificación de patrones, con el propósito de conocer el número mínimo de variables necesarias para representar a los datos y la posibilidad de representarlos en dos o tres dimensiones.

Verificar el resultado del agrupamiento también es importante en análisis de clusters. El mapeo de datos en dos y tres dimensiones es uno de los métodos de verificación más confiables. Cuando tal mapeo de datos de dimensionalidad alta se hace exitosamente, podemos verificar fácilmente al propio algoritmo de agrupamiento.”

Es conveniente ser muy cuidadoso, encontrar la dimensionalidad intrínseca de un conjunto de datos, significa no solamente conocer el número mínimo de variables necesarias para representar a los datos, también significa que con dichas variables podremos seguir distinguiendo la clase a la que pertenece cada objeto.

Representar los datos en dos o tres dimensiones significa “ver” con cierto grado de distorsión la relación de disimilaridad que tienen los datos, de ninguna manera significa que hemos reducido su dimensionalidad a dos o tres dimensiones.

Visto de otra forma, podemos tener el mismo conjunto de datos y distintas nociones de disimilaridad entre ellos, por lo que tendremos distintas representaciones para el mismo conjunto de datos.

No obstante, es rescatable de estos autores ver a los métodos de representación como poderosas herramientas de evaluación de algoritmos para clasificación y agrupamiento.

### 1.1.8 Siedlecki, Siedlecka & Sklansky

En [S<sup>+</sup>88b] encontramos una reseña de distintos métodos de representación y en [S<sup>+</sup>88a] los mismos autores muestran una serie de experimentos con dichos métodos, este trabajo exhaustivo los llevó a plantear lo siguiente.



“Existen dos aplicaciones principales para los mapeos en reconocimiento de patrones: agrupamiento y diseño interactivo de clasificadores.

Hay varias ventajas de hacer agrupamiento con la ayuda de mapeos bidimensionales.

Primero, el observador humano es capaz de identificar y extraer con la ayuda de herramientas gráficas patrones que son indetectables por los buscadores de agrupamientos clásicos.

La segunda ventaja es la mayor facilidad que se tiene al comparar agrupamientos mediante métodos de mapeo que mediante métodos clásicos (dendrogramas).

Una tercera ventaja es que el mapeo bidimensional frecuentemente le da al usuario humano una experiencia de mayor profundidad y mayor confianza en los resultados que cuando se usan técnicas clásicas de agrupamiento.

Entre las desventajas está la tendencia a sesgar el resultado a causa de las proyecciones específicas a cada mapeo. No obstante, esto se puede evitar examinando los datos con diferentes métodos, los cuales tienen diferencias inherentes significativas y por tanto se obtendrán distintos mapeos bidimensionales.

El diseño interactivo de clasificadores automáticos de patrones es la segunda aplicación principal de los métodos de mapeo. Las ventajas y desventajas de su uso son equivalentes al caso de agrupamiento, pero agregando aquí consideraciones acerca de cómo estimar el error que tendrá el clasificador.

Desgraciadamente el uso de estimadores en conjunción con los métodos de mapeo no ha sido tan bueno como en el caso de su uso con los métodos de diseño clásicos. Pero esta situación puede ser compensada por la habilidad y tendencia de los diseñadores humanos para crear hipersuperficies de separación robustas.”

De aquí podemos resaltar que se confía mucho en la habilidad humana para descubrir la “estructura inherente” en un conjunto de objetos y que hace falta algún criterio para decidir cuál algoritmo de representación arroja mejores resultados, o cuál es más adecuado para una aplicación en específico.

Por ello en la sección (2.1) propondremos un criterio que pretende salvar esta deficiencia.

## 1.2 Conceptos Preliminares

Para tener un marco de referencia que permita clasificar a los métodos previos, es necesario enunciar algunos conceptos que se tienen en el *ELCRP*

**Definición 1.2.1** Sea  $U$  un universo de objetos,  $A \subset U$  una **muestra** de objetos, donde  $|A| = m$ , y sean  $E_1, E_2, \dots, E_n$  conjuntos cualesquiera no necesariamente disjuntos, entonces a  $E_1 \times E_2 \times \dots \times E_n$  se le llama **Espacio de Representación Inicial (ERI)** de los objetos. Así, cada objeto del universo se describe por medio de un  $n$ -uplo  $O \in ERI$ .

Note que no se ha puesto restricción alguna sobre los conjuntos  $E_i, i = 1, \dots, n$ , por lo que estos pueden ser del tipo que mejor refleje la naturaleza de las variables con las que se están describiendo los objetos en estudio.

Los objetos en un problema de *RP* están relacionados por medio de una función de disimilaridad<sup>3</sup>, la cual puede ser desde una función analítica hasta un algoritmo muy complejo, éste puede incluso contener una buena cantidad de heurísticas<sup>4</sup>.

**Definición 1.2.2** Sea  $\Delta$  un conjunto totalmente ordenado. Definimos la **función de disimilaridad** como  $disim : ERI \times ERI \rightarrow \Delta$ , que además cumple con las propiedades siguientes, para  $i, j = 1, \dots, m$ :

1.  $disim(O_i, O_i) = \min_{x \in \Delta} \{x\}$
2.  $disim(O_i, O_j) = disim(O_j, O_i)$

Definir la similaridad es muy fácil ya que es un concepto dual, por lo que tenemos la siguiente definición.

---

<sup>3</sup>Dependiente de una aplicación en específico.

<sup>4</sup>En problemas reales interesantes, éste es casi siempre el caso.

**Definición 1.2.3** Sea  $\Delta$  un conjunto totalmente ordenado. Definimos la **función de similaridad** como  $\text{sim} : ERI \times ERI \rightarrow \Delta$ , que además cumple con las propiedades siguientes, para  $i, j = 1, \dots, m$ :

1.  $\text{sim}(O_i, O_i) = \max_{x \in \Delta} \{x\}$
2.  $\text{sim}(O_i, O_j) = \text{sim}(O_j, O_i)$

La definición del  $ERI$  es lo suficientemente general, como para permitir describir objetos, mediante aquellos conjuntos de valores que mejor reflejen la naturaleza de los objetos de un universo dado.

Debemos enfatizar que nuestra definición de disimilaridad (similaridad) es una manera muy general de relacionar objetos en problemas de reconocimiento de patrones, note que si los objetos se representan por vectores en  $\mathbb{R}^n$  la función de disimilaridad podría corresponder a la distancia euclídea.

La definición de función de disimilaridad no exige que ésta cumpla con la desigualdad triangular, dicho de otra manera que sea una norma, ya que es frecuente que este fenómeno ocurra en aplicaciones reales.

Si bien es deseable eliminar la condición de simetría, como se requiere, por ejemplo en [SS95], nosotros no lo abordaremos en este trabajo.

Enfatizando, las definiciones (1.2.1) y (1.2.2), nos permiten modelar un espectro bastante amplio de aplicaciones, además de ser los principales criterios para poder clasificar los métodos de representación previamente existentes, cosa que se hará en la sección (1.3),

Por otro lado, en la bibliografía aparecen dos propuestas de clasificación de algunos métodos de representación, brevemente, en [KM83], se propone la siguiente:

- Algoritmos Globales
  - Métricos
  - No Métricos
- Algoritmos Locales
  - Métricos
  - No Métricos

los algoritmos globales representarán con mayor precisión a objetos con disimilaridad mayor, contrariamente, los algoritmos locales prefieren representar con mayor precisión a objetos con menor disimilaridad, dejando en segundo término a los de mayor disimilaridad.

Por algoritmos métricos se entiende a aquellos que asumen que la función de disimilaridad cumple con las propiedades de una norma, es decir, que el *ERI* es un espacio normado, los no métricos no piden tal restricción.

En esta clasificación se afirma que no hay algoritmos que representen la disimilaridad entre objetos independientemente de la magnitud de ésta. Como veremos en la sección (1.3), es una afirmación falsa. Además no refleja las características que debe cumplir el *ERI* para poder aplicar alguno de los algoritmos.

Por otro lado en [S<sup>+</sup>88b] se presenta la siguiente clasificación:

- Algoritmos Analíticos
- Algoritmos No Analíticos

por algoritmos analíticos los autores entienden aquellos que encuentran una representación para todo objeto en el *ERI*, aunque no esté en la muestra; los algoritmos no analíticos son aquellos que sólo representan a los  $m$  objetos en la muestra, por lo que si la muestra cambia se tiene que recalcular la representación.

Esta es una clasificación demasiado general como para poder tomar alguna decisión sobre qué algoritmo usar, no dice propiedad alguna de la representación obtenida, ni qué propiedades debe cumplir la función de disimilaridad.

Concluyendo, ambas clasificaciones se construyen tomando en cuenta las propiedades matemáticas que los algoritmos poseen, no en su aplicabilidad, por esto presentaremos nuestra propia clasificación en la siguiente sección.

## 1.3 Clasificación de los Métodos Previos

Ahora propondremos nuestra propia clasificación de los métodos de representación previamente existentes, hay tres clases; *restrictivos*, aquellos que toman como *ERI* a  $\mathfrak{R}^n$ ; *generales*, los que cumplen con las definiciones (1.2.1) y (1.2.2); por último los de *alto riesgo*, métodos que obtienen diferentes representaciones para la misma muestra, efecto debido a parámetros extra que requieren para su funcionamiento.

### 1.3.1 Métodos Restrictivos

Estos métodos son aquellos que asumen de entrada que el *ERI* es el propio  $\mathfrak{R}^n$ , por lo cual su aplicación está restringida a problemas que cumplan con esta condición tan restrictiva. Por lo que solamente se muestra el planteamiento general de su funcionamiento.

Hacen una transformación lineal de  $\mathfrak{R}^n$  en  $\mathfrak{R}^2$ . Para representar a un objeto  $x \in \mathfrak{R}^n$  por un punto  $y \in \mathfrak{R}^2$  se hace la multiplicación matricial  $y = Ax + b$ , donde

$$A = \begin{pmatrix} r_1^T \\ r_2^T \end{pmatrix}$$

y  $r_1, r_2 \in \mathfrak{R}^n$ , son los vectores que se obtienen al optimizar un criterio específico (dependiente de cada algoritmo), los detalles los puede ver en [S+88b].

### 1.3.2 Métodos Generales

Dentro de éstos tenemos dos algoritmos, según [KM83] el de Kruskal que es no métrico y el de Sammon que sí lo es.

El de Kruskal (citado en [KM83]) sólo pide como entrada la disimilaridad entre cada par de objetos y asume que ésta es simétrica, por lo que sólo se requieren  $m(m-1)/2$  valores, de los  $m^2$  posibles. La propuesta de Kruskal es minimizar el siguiente funcional de  $2m$  variables:

$$\sqrt{\frac{\sum_{i < j, i=1, j=2}^m (d_{ij} - \bar{d}_{ij})^2}{\sum_{i < j, i=1, j=2}^m d_{ij}^2}} \quad (1.1)$$

donde  $d_{ij}$  es la disimilaridad entre los objetos  $i$ -ésimo y  $j$ -ésimo en el *ERI*,  $\bar{d}_{ij}$  es la distancia euclídea entre el punto  $i$ -ésimo y  $j$ -ésimo en  $\mathfrak{R}^2$ . Kruskal usó el método de descenso por gradiente para hallar el mínimo.

El algoritmo de Sammon [Sam69], usa la idea de minimizar un funcional muy parecido al de Kruskal, siguiendo la misma notación que para (1.1) sería:

$$\frac{1}{\sum_{i < j, i=1, j=2}^m d_{ij}} \sum_{i < j, i=1, j=2}^m \frac{(d_{ij} - \bar{d}_{ij})^2}{d_{ij}} \quad (1.2)$$

si bien Sammon en su artículo original pedía que cada  $d_{ij}$  correspondiera a la norma euclídea en  $\mathfrak{R}^n$ , su algoritmo se puede usar con cualquier función de disimilaridad que cumpla con la definición (1.2.2), él usó el método de descenso más rápido para encontrar el mínimo.

Estos dos algoritmos fueron “inspiración” para el desarrollo posterior de algunas alternativas para aproximarse al mínimo. El problema en ambos casos es tener que minimizar un funcional de  $2m$  variables y guardar  $m(m-1)/2$  valores para la disimilaridad entre objetos, cosa que para esos años (Kruskal en 1964 y Sammon en 1969) era mucho pedir<sup>5</sup>.

### 1.3.3 Métodos de Alto Riesgo

Los métodos que corresponden a esta sección tienen el problema de necesitar, además de la disimilaridad entre los objetos, parámetros extras que conllevarán a producir distintas representaciones, cosa que creemos, es poco recomendable dado que la realidad es única. Los clasificamos como de *alto riesgo*, porque basarse en ellos para tomar alguna decisión puede dar resultados muy sesgados.

El primero es el método de relajación heurística [CL73], trabaja con el funcional (1.2), requiere un parámetro extra  $k < m$ , la idea es que sólo se minimiza (1.2) para los objetos en  $\{1, \dots, k\}$ , los puntos encontrados se fijan en lo que los autores llaman un *marco*, los restantes  $m - k$  objetos son representados con respecto a dicho marco, pero no con respecto a los objetos en  $\{k+1, \dots, m\}$ . Así, conforme demos distintos valores de  $k$  tendremos distintas representaciones, también es sensible a cuáles  $k$  objetos se seleccionen.

El método de triangulación [L<sup>+</sup>77] se apoya en el hecho de que si tenemos tres objetos cualesquiera en el *ERI*, digamos  $O_i$ ,  $O_j$  y  $O_k$ , sus tres disimilaridades se pueden preservar exactamente en  $\mathfrak{R}^2$ , situando a los puntos  $P_i$  y  $P_j \in \mathfrak{R}^2$  a una distancia igual a su disimilaridad, para calcular la ubicación de  $P_k$ , se intersectan dos círculos, uno con centro en  $P_i$  y radio  $disim(O_i, O_k)$ , el otro con centro en  $P_j$  y radio  $disim(O_j, O_k)$ .

Posteriormente los restantes  $m - 3$  objetos son representados con respecto a  $P_i$  y  $P_j$ , preservando exactamente un total de  $3 + 2(m - 3) = 2m - 3$  disimilaridades de las posibles  $m(m - 1)/2$ . Entonces dependiendo de cuales  $2m - 3$

<sup>5</sup>Sammon incluso señala que una limitación de su algoritmo era que podía representar a lo más 250 objetos, si bien empezaba a buscar algoritmos de paginación para manejar memoria virtual.

puntos se elijan para preservar exactamente su disimilaridad, se tendrán distintas representaciones. Otro problema de entrada con este método es que asume que la disimilaridad entre objetos cumple con la desigualdad triangular, de lo contrario no habría intersección posible entre dichos círculos.

En [NW79] se presenta una generalización a (1.2), se propone minimizar el funcional:

$$\frac{1}{\sum_{i < j, i=1, j=2}^m d_{ij}^{p+2}} \sum_{i < j, i=1, j=2}^m d_{ij}^p (d_{ij} - \bar{d}_{ij})^2 \quad (1.3)$$

Note la introducción del parámetro  $p$ , en [S+88b] se afirma que cuando  $p < 0$  el algoritmo representará con mayor precisión objetos con menor disimilaridad<sup>6</sup> (**local**), con  $p > 0$  a objetos con mayor disimilaridad (**global**), con  $p = 0$  no tendrá preferencias<sup>7</sup> (**neutral**), es decir, no importa qué tan grande o pequeña sea la disimilaridad entre los objetos, el algoritmo la representará con la misma exactitud. Nuevamente dependiendo de  $p$  obtendremos diversas representaciones.

Finalmente en [KM83] se presenta una alternativa para el funcional en (1.1), los autores en lugar de preservar las disimilaridades de todos los objetos con respecto a los demás, utilizan la idea de los  $k$ -vecinos más cercanos, de la siguiente manera, si el objeto  $i$  tiene  $k$ -vecinos más cercanos en el *ERI*, su respectivo punto en la representación deberá tener los correspondientes  $k$ -vecinos más cercanos, el mismo argumento se aplica para todos los  $k$ -vecinos más cercanos. Nuevamente dependiendo de  $k$  se obtendrán distintas representaciones.

De esta manera hemos establecido una clasificación de los métodos de representación, tomando en cuenta su aplicabilidad y no sus propiedades de representación (local/global) o cualidades matemáticas (analíticos/no analíticos).

---

<sup>6</sup>De esta manera es erróneo que en [KM83] se tome al algoritmo de Sammon como global, note que (1.2) es igual a (1.3) cuando  $p = -1$ .

<sup>7</sup>También hay que señalar que en este caso la raíz cuadrada de (1.3) es igual a (1.1).

## 1.4 Minimización No Lineal Sin Restricciones

Este espacio está dedicado a los métodos para encontrar el mínimo de una función  $f(\mathbf{x}) : \Re^n \rightarrow \Re[B^+92]$ . Asuma que  $f$  tiene al menos derivadas de segundo orden continuas y que está acotada inferiormente. Así, el problema de

$$\min_{\mathbf{x} \in \Re^n} f(\mathbf{x}) \quad (1.4)$$

tiene una solución  $\mathbf{x}^*$  para la cual  $\nabla f(\mathbf{x}^*) = 0$  y  $\nabla^2 f(\mathbf{x}^*) \geq 0$ .

Se dice que un método para calcular  $\mathbf{x}^*$  es de descenso, si dado un punto inicial  $\mathbf{x}_0$  genera una secuencia de la forma:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \quad (1.5)$$

donde  $\mathbf{d}_k$  es una dirección de descenso en  $\mathbf{x}_k$ , es decir,  $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0$ .

Dependiendo de cuánta información se tenga sobre  $f$  y/o el número de variables que se desean minimizar, se calcula la dirección de descenso  $\mathbf{d}_k$ , dando lugar a diferentes métodos de minimización. También es importante la elección del tamaño de paso  $\alpha_k$  en (1.5), lo que se desea es que estando en  $\mathbf{x}_k$  y descendiendo en la dirección  $\mathbf{d}_k$  se obtenga el mayor decrecimiento posible para  $f$ , lo cual se logra haciendo:

$$\alpha_k = \min_{\alpha \geq 0} f(\mathbf{x}_k + \alpha \mathbf{d}_k) \quad (1.6)$$

lo que se conoce como búsqueda lineal.

En las siguientes secciones se enunciarán los distintos métodos de minimización que como ya se dijo dependen de la forma de obtener  $\mathbf{d}_k$ . Como el aspecto de la minimización es el medio y no el fin de esta investigación, no se presenta cómo se obtienen los diferentes métodos, si quiere más detalles de cómo se desarrollan éstos, puede consultar [B<sup>+</sup>92] y [NL87], son dos excelentes reseñas de los métodos que a continuación se enuncian. Si quiere detalles más profundos (tanto de teoría como de implantación) y ejemplos, debe consultar [Fle87] y/o [DB83].

### 1.4.1 Gradiente Conjugado

Esta familia de métodos se basa en que las direcciones de descenso sean conjugadas. En general, sea  $\mathbf{G}$  una matriz simétrica, definida positiva, entonces



los vectores  $\mathbf{u} \neq 0$ ,  $\mathbf{v} \neq 0$  y  $\mathbf{u} \neq \mathbf{v}$  son conjugados con respecto a  $\mathbf{G}$  si  $\mathbf{u}^T \mathbf{G} \mathbf{v} = 0$ .

Sea

$$\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k) = \nabla f(\mathbf{x}_k),$$

entonces el vector de descenso  $\mathbf{d}_{k+1}$  para el método de gradiente conjugado es

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$$

donde si  $\beta_k$  se calcula como

$$\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k} = \frac{\mathbf{g}_{k+1}^2}{\mathbf{g}_k^2} \quad (1.7)$$

se dice que es el algoritmo Fletcher-Reeves. Pero si se calcula como

$$\beta_k = \frac{\mathbf{g}_{k+1}^T (\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{g}_k^T \mathbf{g}_k} \quad (1.8)$$

corresponde al algoritmo de Polak-Ribiere. Cuando  $f$  es cuadrática (1.7) y (1.8) son iguales, sin embargo, es preferible (1.8) cuando  $f$  es no cuadrática.

La ventaja de esta familia de métodos es que no requieren del Hessiano de  $f$ , almacenarlo tiene un costo en memoria de  $n(n-1)/2$  valores, cantidad que es prohibitiva cuando  $n$  es muy “grande”, peor aún en nuestro caso, ya que la matriz de disimilaridades requiere de igual cantidad de espacio en memoria. La desventaja es que no convergerá tan rápido como los métodos de las siguientes subsecciones.

### 1.4.2 Quasi-Newton

Si el número de variables a minimizar no es “muy grande” y no cuenta con la forma analítica del Hessiano, puede usar la familia de métodos quasi-newton, su principal característica es que obtienen aproximaciones al Hessiano, éstas son actualizadas en cada iteración.

En esta familia  $\mathbf{d}_{k+1}$  se calcula resolviendo:

$$\mathbf{B}_{k+1} \mathbf{d}_{k+1} = -\mathbf{g}_{k+1}$$

donde  $\mathbf{B}_{k+1}$  es una aproximación al Hessiano de  $f$  y se actualiza, por ejemplo, de acuerdo a la fórmula BFGS

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{1}{s_k^T \mathbf{B}_k s_k} \mathbf{B}_k s_k s_k^T \mathbf{B}_k + \frac{1}{y_k^T s_k} y_k y_k^T$$

donde  $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k = \alpha_k \mathbf{d}_k$  y  $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ .

Estos métodos también se pueden construir actualizando la inversa de la aproximación al Hessiano, es decir, si  $\mathbf{H}_k$  es una aproximación a la inversa del Hessiano, entonces

$$\mathbf{d}_{k+1} = -\mathbf{H}_{k+1} \mathbf{g}_{k+1},$$

$\mathbf{H}_{k+1}$  se actualiza de acuerdo a la fórmula BFGS

$$\mathbf{H}_{k+1} = \mathbf{H}_k - \frac{\mathbf{H}_k \mathbf{y}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} + \left( 1 + \mathbf{y}_k^T \frac{\mathbf{H}_k \mathbf{y}_k}{\mathbf{s}_k^T \mathbf{y}_k} \right) \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k}.$$

### 1.4.3 Newton

Si se tiene el Hessiano de  $f$  en forma analítica y no es “muy grande” el número de variables a minimizar,  $\mathbf{d}_{k+1}$  se obtiene de la siguiente manera:

$$\mathbf{d}_{k+1} = -\mathbf{G}_{k+1}^{-1} \mathbf{g}_{k+1}$$

donde se asume que la inversa del Hessiano analítico  $\mathbf{G}_{k+1}^{-1}$  existe. Es decir, el método involucra la solución al sistema de ecuaciones

$$\mathbf{G}_{k+1} \mathbf{d}_{k+1} = -\mathbf{g}_{k+1}.$$

El método tiene convergencia cuadrática, pero no siempre se tiene la segunda derivada en forma analítica. Conceptualmente es el más simple de las tres familias que se han mencionado.

Ya que se espera que el número de objetos a representar sea “muy grande” y que el funcional a minimizar en (2.12) es no cuadrático, se optó por utilizar el método de minimización de gradiente conjugado, calculando  $\beta_k$  como en (1.8). Este método ya se encuentra codificado en varios lenguajes, en particular en ANSI C en la referencia [Sof92].

# Capítulo 2

## Propuesta

Una vez analizados los métodos previos a este trabajo, se dará una definición de lo que creemos se debe esperar de los algoritmos de representación, posteriormente enfocaremos el problema desde dicha perspectiva y concluiremos con nuestro algoritmo de representación isoanalógica.

### 2.1 Representación Isoanalógica

La clasificación de los métodos de representación en base a su utilidad y adecuación metodológica, permite proponer la siguiente

**Definición 2.1.1** Sean:

- $ERI = E_1 \times E_2 \times \dots \times E_n$  el espacio de representación inicial,
- $ERF = E'_1 \times E'_2 \times \dots \times E'_l$  el espacio de representación final,
- $\Delta, \Delta'$  conjuntos totalmente ordenados,
- $disim_{ERI} : ERI \times ERI \rightarrow \Delta$  la función de disimilaridad en el ERI que cumple:
  1.  $disim_{ERI}(O_i, O_i) = \min_{x \in \Delta} \{x\}$ ,
  2.  $disim_{ERI}(O_i, O_j) = disim_{ERI}(O_j, O_i)$ ,
- $disim_{ERF} : ERF \times ERF \rightarrow \Delta'$  la función de disimilaridad en el ERF que cumple:

1.  $\text{disim}_{ERF}(O'_i, O'_i) = \min_{x \in \Delta'} \{x\}$ ,
2.  $\text{disim}_{ERF}(O'_i, O'_j) = \text{disim}_{ERF}(O'_j, O'_i)$ ,

•  $a : \Delta \rightarrow \Delta'$ , una función parcial de analogía de  $\Delta$  en  $\Delta'$ , que satisface:

1.  $a(\min_{x \in \Delta} \{x\}) = \min_{x \in \Delta'} \{x\}$ ,
2.  $\forall x, y \in \Delta$  si  $x <_{\Delta} y$  entonces  $a(x) <_{\Delta'} a(y)$ .

•  $\pi_1 : A \times B \rightarrow A$  y  $\pi_2 : A \times B \rightarrow B$  proyecciones tales que si  $(x, y) \in A \times B$  entonces:

$$\pi_1(x, y) = x, \pi_2(x, y) = y.$$

decimos que una representación  $R \subset ERI \times ERF$  es isoanalógica cuando cumple:

1.  $|R| < \infty$ ,

2.  $\forall r_i, r_j \in R$ , si

$$\text{disim}_{ERI}(\pi_1(r_i), \pi_1(r_j)) = x \in \Delta$$

entonces

$$\text{disim}_{ERF}(\pi_2(r_i), \pi_2(r_j)) \leq a(x) \in \Delta',$$

3. Si  $\exists r_i, r_j \in R, i \neq j$ , tal que

$$\pi_1(r_i) = \pi_1(r_j)$$

entonces

$$\text{disim}_{ERF}(\pi_2(r_i), \pi_2(r_k)) = \text{disim}_{ERF}(\pi_2(r_j), \pi_2(r_k)), \forall r_k \in R,$$

4.  $R$  sólo debe depender de  $\text{disim}_{ERI}$ ,  $\text{disim}_{ERF}$  y de  $a$ .

Que  $R \subset ERI \times ERF$  y no  $R : ERI \rightarrow ERF$ , es debido a que se busca un algoritmo que asocie objetos en la dimensión  $n$  con objetos en la dimensión  $l$ , pero posiblemente el algoritmo en distintas ejecuciones asocie un objeto en el  $ERI$  con distintos objetos en el  $ERF$ , para evitar que el número de asociaciones posibles sea infinito se establece la condición (1) de la definición (2.1.1), esta condición la violan los métodos que hemos clasificado como generales.

El sentido de la condición (2) de la definición (2.1.1) es el de garantizar que la disimilaridad de los objetos en el  $ERI$  sea preservada en medida de lo posible por los objetos en el  $ERF$ , vía la disimilaridad en el  $ERF$  y de acuerdo a la analogía dada entre  $\Delta$  y  $\Delta'$ .

La condición (3) de la definición (2.1.1) propone que si acaso los objetos en el  $ERI$  son asociados a más de un objeto en el  $ERF$ , los objetos asociados deben de mantener la misma disimilaridad en el  $ERF$  con respecto a todos los demás objetos asociados, dicho de otra manera que las distintas asociaciones deben ser rígidas. Esta condición junto con la condición (1), nos garantizan que es suficiente ejecutar un algoritmo de representación isoanalógica una sola vez, ya que cualquier otra posible asociación entre objetos, obtenida por tal algoritmo, mantendrá las mismas disimilaridades en el  $ERF$ .

Como queremos que la representación muestre por medio de objetos en el  $ERF$  la estructura que tienen los objetos en el  $ERI$ , en base a la disimilaridad que en ese espacio guardan y a la analogía que existe entre  $\Delta$  y  $\Delta'$ , la condición (4) de la definición (2.1.1) es necesaria.

Los métodos que clasificamos como restrictivos son los que exigen de entrada que el  $ERI$  sea  $\mathfrak{R}^n$ . Los que se clasificaron como generales violan la condición (1) y (4) de (2.1.1), porque  $R$  no es finita y las asociaciones que harán dependerán del punto inicial que se dé como mínimo. Por otro lado los métodos que hemos clasificado de alto riesgo violan las condiciones (3) y (4) de (2.1.1), como ya se explicó a causa de requerir parámetros extras se producen asociaciones no rígidas.

En este lugar es conveniente decir que, el camino que se construye desde el momento en que un especialista en cualquier rama del conocimiento le plantea una aplicación a un modelador de problemas de reconocimiento de patrones, hasta que se obtiene una matriz de disimilaridades, es muy largo e independiente del enfoque que se tenga al reconocimiento de patrones.

Para el problema de representación isoanalógica lo único que se necesita conocer de inicio son:

1. El número  $m$  de objetos a representar,
2. La matriz de disimilaridades en el *ERI* entre cada par de objetos ya analogizada, es decir:  $a(\text{disim}_{ERI}(O_i, O_j)); i, j \in \{1, \dots, m\}$ .

Será responsabilidad del algoritmo de representación isoanalógica en base a estos datos, calcular  $m$  objetos en el *ERF* que satisfagan la definición (2.1.1).

## 2.2 Un Primer Acercamiento

Asociemos cada objeto en el *ERI* con un punto en  $\mathfrak{R}^2$  el cual será nuestro *ERF*, i.e.,

$$O_i \mapsto P_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}, i = 1, \dots, m;$$

usaremos la norma euclídea en  $\mathfrak{R}^2$  para medir la disimilaridad en el *ERF*

$$\| P_i - P_j \| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2};$$

$\Delta'$  será  $[0, b]$ ,  $b \in \mathfrak{R}$  y sea  $\text{disim}_{ij} = a(\text{disim}_{ERI}(O_i, O_j))$ .

Como primer acercamiento podemos plantear nuestro problema como uno de minimización no lineal sin restricciones de la siguiente manera:

$$\min f(P_1, P_2, \dots, P_m) = \sum_{i < j, i=1, j=2}^m (\| P_i - P_j \|^2 - \text{disim}_{ij}^2)^2 \quad (2.1)$$

La semántica del funcional en (2.1) dice que debemos de situar los puntos en el *ERF* a una distancia igual a su disimilaridad en el *ERI*, note que el cuadrado sobre la norma es para facilitar el desarrollo analítico-numérico; para no alterar la relación  $\| P_i - P_j \| = \text{disim}_{ij}$  también elevamos al cuadrado la disimilaridad. Por otro lado, cada término del funcional se eleva al cuadrado para que éste sea mayor o igual que cero garantizando que existe mínimo.

Ahora procederemos a encontrar el gradiente del funcional en (2.1) para analizar sus puntos críticos.

**Proposición 2.2.1** *La primera derivada parcial de  $f$  con respecto a cada punto es:*

$$\frac{\partial f}{\partial P_i} = 4 \sum_{k=1}^m f_{ik}(P_i - P_k); \quad i = 1, \dots, m; \quad i \neq k. \quad (2.2)$$

*Prueba:* Sea  $f_{rs} = \| P_r - P_s \|^2 - \text{disim}_{rs}^2$ ;  $r, s = 1, \dots, m$ ;  $r \neq s$ ; entonces

$$\frac{\partial f}{\partial P_i} = \left( \frac{\partial f}{\partial x_j} \right) = \sum_{k=1}^{i-1} \frac{\partial f_{ki}^2}{\partial P_i} + \sum_{k=i+1}^m \frac{\partial f_{ik}^2}{\partial P_i} \quad (2.3)$$

dado que  $f_{rs} = f_{sr}$  tenemos la siguiente igualdad

$$\begin{aligned} \frac{\partial f_{ki}^2}{\partial P_i} &= 2f_{ki}(2(P_k - P_i)(-1)) = \\ &= 4f_{ki}(P_i - P_k) = \end{aligned} \quad (2.4)$$

$$\begin{aligned} \frac{\partial f_{ik}^2}{\partial P_i} &= 2f_{ik}(2(P_i - P_k)) = \\ &= 4f_{ik}(P_i - P_k) \end{aligned} \quad (2.5)$$

sustituyendo (2.4) y (2.5) en (2.3) obtenemos (2.2).■

Con estos elementos analíticos podemos encontrar los puntos críticos del funcional, note que (2.2) nos genera el siguiente sistema de ecuaciones que hay que resolver para ver cuando  $\nabla f = \bar{0}_{2m}$ .

$$\begin{aligned} f_{12}(P_1 - P_2) + f_{13}(P_1 - P_3) + \dots \\ \dots + f_{1m}(P_1 - P_m) &= \bar{0}_2 \\ f_{21}(P_2 - P_1) + f_{23}(P_2 - P_3) + \dots \\ \dots + f_{2m}(P_2 - P_m) &= \bar{0}_2 \\ &\vdots \\ f_{m1}(P_m - P_1) + f_{m2}(P_m - P_2) + \dots \\ \dots + f_{mm-1}(P_m - P_{m-1}) &= \bar{0}_2 \end{aligned} \quad (2.6)$$

Sumando todas las ecuaciones de (2.6) y en virtud de que  $f_{rs}(P_r - P_s) = -f_{sr}(P_s - P_r)$  hayamos que

$$\nabla f = \bar{0}_{2m}; \quad \forall P_i \in \mathfrak{R}^2; \quad i = 1, \dots, m.$$

Es decir cualesquiera  $m$  puntos en  $\mathbb{R}^2$ , conforman un punto crítico para el funcional (2.1), por lo que la idea expresada por dicho funcional viola las condiciones (1) y (4) de la definición (2.1.1)<sup>1</sup>.

Lo anterior sucede dado que tenemos cualquier lugar del plano para situar los puntos que conformarán nuestra representación, de tal manera que adoptamos una convención análoga a la que en Física se denomina sistema de referencia inercial.

## 2.3 Propuesta Mejorada

Empecemos analizando qué sucede al elegir dos puntos cualesquiera de los  $m$  y fijarlos en alguna parte del plano, por ejemplo podemos hacer lo siguiente:

$$P_a = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, P_b = \begin{pmatrix} disim_{ba} \\ 0 \end{pmatrix}; \\ a, b \in \{1, \dots, m\}; a \neq b;$$

es decir, un punto lo situamos en el origen y otro en el eje  $x$  a una distancia del origen igual a la disimilaridad entre los objetos elegidos.

Así que al sumar todas las ecuaciones de (2.6), y ya que las derivadas parciales de  $f$  con respecto a  $P_a$  y  $P_b$  son cero obtenemos:

$$\begin{aligned} f_{1b}(P_1 - P_b) + f_{1a}(P_1) + \\ f_{2b}(P_2 - P_b) + f_{2a}(P_2) + \\ \vdots \\ f_{mb}(P_m - P_b) + f_{ma}(P_m) = \bar{0} \end{aligned}$$

lo cual podemos escribir como:

$$\sum_{k=1, k \neq a, k \neq b}^m f_{kb}(P_k - P_b) + f_{ka}(P_k) = \bar{0} \quad (2.7)$$

Notemos que (2.7) se cumple en particular cuando  $f_{kb} = 0 = f_{ka}; k = 1, \dots, m; k \neq a; k \neq b$ ; por lo que tenemos el siguiente desarrollo:

$$\| P_k - P_b \|^2 - disim_{kb}^2 - (\| P_k - P_a \|^2 - disim_{ka}^2) = 0 \Rightarrow$$

<sup>1</sup>Lo mismo ocurre con los algoritmos de Kruskal y Sammon.



$$(x_k - disim_{ba})^2 + y_k^2 - disim_{kb}^2 - (x_k^2 + y_k^2 - disim_{ka}^2) = 0 \Rightarrow$$

$$f_k^2 - 2x_k disim_{ba} + disim_{ba}^2 + f_k^2 - disim_{kb}^2 - f_k^2 - f_k^2 + disim_{ka}^2 = 0 \Rightarrow$$

$$-2x_k disim_{ba} + disim_{ba}^2 - disim_{kb}^2 + disim_{ka}^2 = 0 \Rightarrow$$

$$x_k = \frac{disim_{ab}^2 - disim_{kb}^2 + disim_{ka}^2}{2disim_{ab}} \quad (2.8)$$

sustituyendo en  $f_{ka} = 0$  tenemos:

$$y_k = \pm \left( disim_{ka}^2 - \left( \frac{disim_{ba}^2 - disim_{kb}^2 + disim_{ka}^2}{2disim_{ba}} \right)^2 \right)^{1/2} \quad (2.9)$$

El problema al que ahora nos enfrentamos es que la relación en (2.9) puede dar como resultado un número complejo. Pero podemos minimizar (2.1) sólo con respecto a  $y_i$ ,  $i = 1, \dots, m$ ,  $i \neq b$ ,  $i \neq a$ , es decir, en lugar de minimizar un funcional de  $2m$  variables, minimizaremos solamente  $m - 2$  variables, aprovechando la ecuación (2.8).

Desgraciadamente no siempre funciona el esquema de fijar cualesquiera dos puntos, desarrollemos un ejemplo simple. Supongamos que como *ERI* tenemos al propio  $\mathbb{R}^2$ , por lo tanto el funcional en (2.1) debe de comportarse como el mapeo identidad, salvo posibles transformaciones rígidas. Así, podemos tomar los puntos que definen un trapeoide, digamos que  $P_1 = (.25, .5)^T$ ,  $P_2 = (0, 0)^T$ ,  $P_3 = (1, 0)^T$  y  $P_4 = (.75, .5)^T$ .

Tomando la norma euclídea entre los puntos como su disimilaridad, obtenemos la matriz de disimilaridades siguiente:

$$\begin{pmatrix} 0 & .559 & .9014 & .5 \\ .559 & 0 & 1 & .9014 \\ .9014 & 1 & 0 & .559 \\ .5 & .9014 & .559 & 0 \end{pmatrix} \quad (2.10)$$

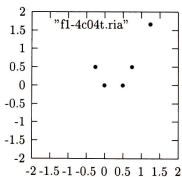
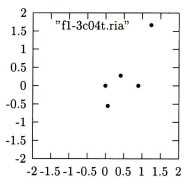
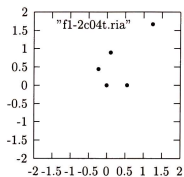


Figura 2.1: Representaciones obtenidas al fijar  $P_1$  y  $P_2$ ;  $P_1$  y  $P_3$ ;  $P_1$  y  $P_4$ , para la matriz de disimilaridades (2.10).

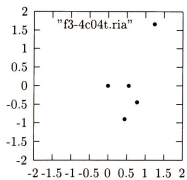
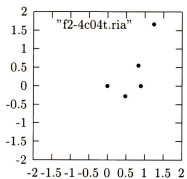
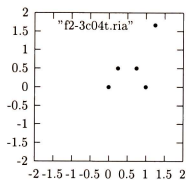


Figura 2.2: Representaciones obtenidas al fijar  $P_2$  y  $P_3$ ;  $P_2$  y  $P_4$ ;  $P_3$  y  $P_4$ , para la matriz de disimilaridades (2.10).

como se muestra en las figuras (2.1), (2.2), no importa cuáles puntos se fijen, obtenemos la misma figura salvo rotaciones, traslaciones, reflexiones y composición de éstas.

Veamos qué sucede si distorsionamos un poco la relación de disimilaridad que guardan los puntos del trapecio, por ejemplo, haciendo  $disim_{12} = .2^2$ . Intuitivamente esperamos ver una figura como las obtenidas para la matriz de disimilaridades (2.10), excepto que uno de los lados no paralelos del trapecio debe ser más corto, su matriz de disimilaridades sería:

$$\begin{pmatrix} 0 & .2 & .9014 & .5 \\ .2 & 0 & 1 & .9014 \\ .9014 & 1 & 0 & .559 \\ .5 & .9014 & .559 & 0 \end{pmatrix} \quad (2.11)$$

En esta ocasión observamos cinco tipos de representaciones por lo que se viola la condición (3) de la definición (2.1.1). En la figura (2.3) se fijaron los puntos  $P_1$  y  $P_4$ , efectivamente evoca un trapecio aunque no se corresponde con lo esperado, en la figura (2.4) se han fijado a  $P_3$  y  $P_4$ , vemos que mejora con respecto a lo esperado, pero el lado distorsionado es aún grande, note que  $disim_{14} = .5$  y  $disim_{34} = .559$  y mejora mucho la figura (2.4) a la figura (2.3).

Al fijar  $P_1$  y  $P_3$  se obtiene una representación isoanalógica a aquella resultado de fijar  $P_2$  y  $P_4$ , note que  $disim_{13} = disim_{24} = .9014$ , como se ve en las figuras (2.5) y (2.6) el lado distorsionado ya no es tan grande.

Como se puede observar en la figura (2.7), cuando se fija a  $P_1$  y  $P_2$  la representación obtenida dista mucho de parecer un trapecio.

Finalmente al tomar los puntos con disimilaridad máxima ( $P_2$  y  $P_3$ ) obtenemos una representación que sin discusión es superior a las otras como se observa en la figura (2.8), sin embargo, note el parecido con las figuras (2.5) y (2.6), y que  $disim_{23} = 1$  y  $disim_{13} = disim_{24} = .9014$ .

Hemos observado con este ejemplo que entre más lejanos sean los puntos que se fijan, las representaciones son mejores y que el algoritmo es muy sensible a ello, es decir, la representación mejoró sustancialmente cuando se fijaron los puntos con disimilaridad .559 a la de aquellos con disimilaridad .5. También que fijar a  $P_1$  y  $P_3$  da como resultado una representación isoanalógica a

<sup>2</sup>Esta relación entre estos puntos (objetos) es inverosímil en  $\mathfrak{R}^2$ , pero nada extraña en problemas reales.

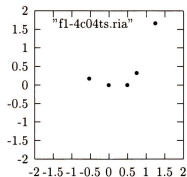


Figura 2.3: Representación obtenida al fijar  $P_1$  y  $P_4$ , para la matriz de disimilaridades (2.11).

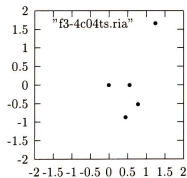


Figura 2.4: Representación obtenida al fijar  $P_3$  y  $P_4$ , para la matriz de disimilaridades (2.11).

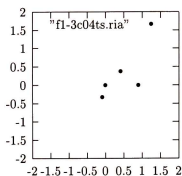


Figura 2.5: Representación obtenida al fijar  $P_1$  y  $P_3$ , para la matriz de disimilaridades (2.11).

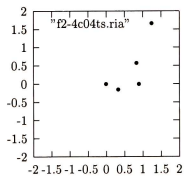


Figura 2.6: Representación obtenida al fijar  $P_2$  y  $P_4$ , para la matriz de disimilaridades (2.11).

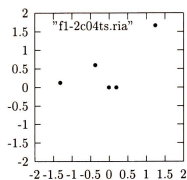


Figura 2.7: Representación obtenida al fijar  $P_1$  y  $P_2$ , para la matriz de disimilaridades (2.11).

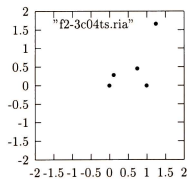


Figura 2.8: Representación obtenida al fijar  $P_2$  y  $P_3$ , para la matriz de disimilaridades (2.11).

la producida por fijar  $P_2$  y  $P_4$ , dado que tienen la misma disimilaridad. Por lo que la estrategia que elegimos es tomar siempre los objetos más lejanos en el *ERI* para fijarlos en el *ERF*. Con esto obtenemos una representación que sí es isoanalógica, salvo posiblemente una reflexión, debida a los puntos iniciales dados al algoritmo de minimización no lineal sin restricciones.



Ahora podemos presentar nuestro algoritmo.

**Algoritmo 2.3.1** *Algoritmo para la representación isoanalógica en  $\mathbb{R}^2$  de objetos  $n$ -dimensionales.*

**Entrada:** Una matriz de disimilaridades entre  $m$  objetos  $Mdisim^3$ , tal que  
 $Mdisim_{ij} = disim_{ij}; i, j = 1, \dots, m.$

**Salida:**  $P_i \in \mathbb{R}^2; i = 1, \dots, m.$

1. Elegir  $a, b \in \{1, \dots, m\}; a \neq b$ , si

$$disim_{ab} = \max_{x \in Mdisim} \{x\}$$

2. Definir

$$P_a = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, P_b = \begin{pmatrix} disim_{ab} \\ 0 \end{pmatrix}$$

3. Para  $k = 1, \dots, m; k \neq a; k \neq b$  hacer

$$x_k = \frac{disim_{ab}^2 - disim_{kb}^2 + disim_{ka}^2}{2disim_{ab}}$$

4. Minimizar<sup>4</sup>

$$f(y_{i_1}, y_{i_2}, \dots, y_{i_{m-2}}) = \sum_{i < j, i=i_1, j=i_2}^{i_{m-2}} (\|P_i - P_j\|^2 - disim_{ij}^2)^2 \quad (2.12)$$

donde  $i_1, i_2, \dots, i_{m-2} \in \{1, 2, \dots, m\} - \{a, b\}.$

---

<sup>3</sup>Simétrica y con ceros en la diagonal principal.

<sup>4</sup>En particular nosotros usamos el método de gradiente conjugado como aparece en [Sof92].



## Capítulo 3

# Implantación Computacional

En este capítulo se presentan los detalles de la implantación computacional, se verán las estructuras de datos utilizadas, las funciones más importantes, las dependencias entre las funciones, las herramientas que se utilizaron para su puesta a punto y ejemplos de aplicación.

### 3.1 Estructuras de Datos

Las estructuras de datos utilizadas son:

1. `float *fxs` arreglo de elementos de punto flotante para almacenar las coordenadas  $x_i$ ,  $i = 1, \dots, m$ , calculadas como en (2.8).
2. `float *fpini` arreglo de elementos de punto flotante para almacenar la aproximación inicial al mínimo, como se especifica en (2.12). En este mismo arreglo el algoritmo de gradiente conjugado de Polak-Riviere regresa la aproximación final al mínimo.
3. `float *fys` arreglo de elementos de punto flotante para almacenar las coordenadas  $y_i$ ,  $i = 1, \dots, m$ , este arreglo contiene el resultado de la minimización con las coordenadas  $y_a = 0$  y  $y_b = 0$  incluidas.
4. `float **flejanía` como se debe almacenar la matriz de disimilaridades y ésta es simétrica y con ceros en la diagonal principal, se optó por tener un arreglo de  $m - 1$  apuntadores a arreglos de punto flotante, el primero de dimensión  $m - 1$ , el segundo de dimensión  $m - 2, \dots$ , el último de dimensión 1. De esta manera sólo se almacena la parte

triangular superior de la matriz. Otra forma sería almacenar la parte triangular superior de la matriz en un sólo arreglo, pero conlleva tener que evaluar un polinomio de direccionamiento cada vez que se accede a la matriz.

5. float `fminimo` flotante donde se almacena el valor del funcional en el punto mínimo.
6. int `imobjetos` entero para almacenar el número de objetos a representar.
7. int `P_subindice_a`, `P_subindice_b` par de enteros para indicar a los objetos más disimilares en la representación.
8. int `iteraciones` entero donde se almacena el número de iteraciones que se llevaron a cabo para encontrar el mínimo.

## 3.2 Funciones más Importantes

En esta sección se especifica lo que hacen las funciones más importantes, estableciendo el nombre de cada función, el valor que regresan, sus parámetros de: entrada, salida y entrada/salida; además de su propósito.

1. **Función:** float `**get_farness_mat(int *iobjetos)`

**Regresa:** Un arreglo de apuntadores a los renglones que están en la parte triangular superior de la matriz de disimilaridades.

**Parámetros de Entrada:** Ninguno.

**Parámetros de Salida:**

- (a) int `*iobjetos` número de objetos a representar.

**Parámetros de Entrada/Salida:** Ninguno.

**Propósito:** Obtiene los arreglos necesarios para la matriz de disimilaridades, lee dicha matriz y el número de objetos a representar.

2. **Función:** void `max_farness_mat(float **fmatriz, int icuantos, int *iren-  
glon, int *icolumna)`

**Regresa:** void

**Parámetros de Entrada:**

- (a) float **\*\*fmatriz** Matriz de disimilaridades.
- (b) int **icuantos** Número de objetos a representar.

**Parámetros de Salida:**

- (a) int **\*irenglon** El primero de los dos objetos con disimilaridad mayor.
- (b) int **\*icolumna** El segundo de los dos objetos con disimilaridad mayor.

**Parámetros de Entrada/Salida:** Ninguno.

**Propósito:** Calcula los objetos con disimilaridad máxima.

3. **Función:** float **\*calculaxs**(float **\*\*fmatriz**, int **icuantos**, int **ip\_a**, int **ip\_b**)

**Regresa:** Un arreglo de punto flotante que almacena las coordenadas  $x_i, i = 1, \dots, m$ .

**Parámetros de Entrada:**

- (a) float **\*\*fmatriz** Matriz de disimilaridades.
- (b) int **icuantos** Número de objetos a representar.
- (c) int **ip\_a** El primero de los dos objetos con disimilaridad mayor.
- (d) int **ip\_b** El segundo de los dos objetos con disimilaridad mayor.

**Parámetros de Salida:** Ninguno.

**Parámetros de Entrada/Salida:** Ninguno.

**Propósito:** Calcular las coordenadas  $x_i, i = 1, \dots, m$ , como en (2.8).

4. **Función:** void **frprmn**(float **p**[], int **n**, float **ftol**, int **\*iter**, float **\*fret**, float **(\*func)**(float []), void **(\*dfunc)**(float [], float []))

**Regresa:** void

**Parámetros de Entrada:**

- (a) int **n** Número de variables a minimizar.
- (b) float **ftol** Tolerancia para la convergencia al valor de la función.
- (c) float **(\*func)**(float []) Función a minimizar.
- (d) void **(\*dfunc)**(float [], float []) Gradiente de la función a minimizar.

**Parámetros de Salida:**

- (a) `int *iter` Número de iteraciones efectuadas para alcanzar el mínimo.
- (b) `float *fret` Valor de la función en el mínimo.

**Parámetros de Entrada/Salida:**

- (a) `float p[]` A la entrada contiene el punto de aproximación inicial, a la salida contiene el punto donde la función alcanza el mínimo.

**Propósito:** Minimiza (2.12) por el método de gradiente conjugado de Polak-Riviere, extraído de [Sof92].

5. **Función:** `float fex(float *fvec_a_evaluar)`

**Regresa:** El valor del funcional (2.12).

**Parámetros de Entrada:**

- (a) `float *fvec_a_evaluar` Arreglo que contiene el punto en el que se desea evaluar el funcional.

**Parámetros de Salida:** Ninguno.

**Parámetros de Entrada/Salida:** Ninguno.

**Propósito:** Evalúa el funcional (2.12) tomando en cuenta que  $x_i, i = 1, \dots, m; y_a$  y  $y_b$  son constantes.

6. **Función:** `void dfex(float *fvec, float *fdvec)`

**Regresa:** `void`

**Parámetros de Entrada:**

- (a) `float *fvec` Arreglo que contiene el punto en el que se desea hayar el gradiente del funcional.

**Parámetros de Salida:**

- (a) `float *fdvec` Arreglo que contiene el gradiente del funcional en el punto dado por el parámetro de entrada.

**Parámetros de Entrada/Salida:** Ninguno.

**Propósito:** Calcula el gradiente (2.2) tomando en cuenta que  $x_i, i = 1, \dots, m; y_a$  y  $y_b$  son constantes.

### 3.3 Dependencias Funcionales

La relación que tiene cada función con respecto a las funciones que llama se establece a continuación.

1. `main(argv)` Función principal.
  - (a) `flejania = get_farness_mat(&imobjetos)` Obtiene los arreglos necesarios para la matriz de disimilaridades y la lee.
  - (b) `max_farness_mat(flejania, imobjetos, &P_subindice_a, &P_subindice_b)` Calcula los objetos con disimilaridad máxima.
  - (c) `fxs = calculaxs(flejania, imobjetos, P_subindice_a, P_subindice_b)` Calcula las coordenadas  $x_i, i = 1, \dots, m$ , como en (2.8).
  - (d) `frprmn(fpini - 1, imobjetos 2, 00000.1, &iteraciones, &fminimo, fenx, dfenx)` Minimiza (2.12) por el método de gradiente conjugado de Polak-Riviere, extraído de [Sof92].
  - (e) `fys = agrega(fpini)` Agrega las coordenadas  $y_a$  y  $y_b$  a su parámetro.

### 3.4 Herramientas Utilizadas

La infraestructura computacional utilizada fue: una computadora con procesador a 100 Mhz., sistema operativo Linux S.u.S.E., el algoritmo se implantó en ANSI C, se compiló con gcc y las gráficas se dibujaron con gnuplot. El documento se procesó con  $\LaTeX$ .

Note que todo el software usado es **Software Libre** disponible via internet, existe para plataformas diversas, por ejemplo, en una etapa intermedia de esta tesis se utilizó una computadora personal con sistema operativo MS-DOS, desde entonces se trabajó con gcc, gnuplot y  $\LaTeX$ .

La decisión de utilizar gnuplot se basó en que puede generar las gráficas en diferentes formatos, incluyendo por supuesto gráficas en código  $\LaTeX$ puro.

El compilador gcc ha mostrado en la práctica, ser, sino el mejor, sí uno de los mejores, no sólo para C, también para sus descendientes C++ y Objective C.

Finalmente, no requiere justificación el uso de  $\LaTeX$ .

### 3.5 Ejemplos

Aunque en Internet se encuentran disponibles matrices de disimilaridad, no están disponibles los expertos ni los modeladores que las obtuvieron, por lo tanto, utilizar dichos datos no es útil para evaluar a nuestro algoritmo para la representación isoanalógica, así mostraremos solamente 8 representaciones para un conjunto de 109 objetos obtenidos de la manera siguiente:

1. Se capturaron las coordenadas de los puntos donde se pulsaba el botón de un ratón en un plano cartesiano, al emular que se escribía la palabra *ISAS*.
2. Los puntos se enumeraron conforme se iban capturando.
3. Se tomó la norma euclídea entre los puntos y con estas distancias se formó la matriz de disimilaridades.

En la figura (3.1) observamos el resultado del algoritmo presentado, es decir, se fijaron los puntos con mayor disimilaridad ( $P_{15}$  y  $P_{86}$ ), note que efectivamente al hacer una reflexión sobre el eje  $x$  se lee *ISAS*. El valor del funcional en la solución es  $3.7724719 \times 10^{-12}$ , realizó 93 iteraciones y ocupó 3 minutos y 22 segundos de procesador.

La figura (3.2) se obtuvo fijando a  $P_1$  y  $P_{55}$ . El valor del funcional en la solución es  $6.4183337 \times 10^{-12}$ , realizó 64 iteraciones y ocupó 2 minutos y 10 segundos de procesador. Un hecho notable en este ejemplo es el siguiente, como efectuó sólo 64 iteraciones, se hicieron tres ejecuciones más, sorprendentemente en todas llegó a 64 iteraciones y al mismo valor del funcional en la solución. Ocupó en la siguiente ejecución 2 segundos más, en las últimas dos 1 segundo más. Tal parece que es insensible al punto inicial de aproximación, ya que éste se escoge aleatoriamente.

La figura (3.3) se obtuvo fijando a  $P_{55}$  y  $P_{109}$ . El valor del funcional en la solución es  $5.0113780 \times 10^{-12}$ , realizó 77 iteraciones y ocupó 2 minutos y 36 segundos de procesador.

La figura (3.4) se obtuvo fijando a  $P_{36}$  y  $P_{105}$ . El valor del funcional en la solución es  $6.1023786 \times 10^{-12}$ , realizó 71 iteraciones y ocupó 2 minutos y 19 segundos de procesador.

La figura (3.5) se obtuvo fijando a  $P_1$  y  $P_2$ . El valor del funcional en la solución es  $8.9629609 \times 10^{-10}$ , realizó 78 iteraciones y ocupó 2 minutos y 34 segundos de procesador. Por la forma en que se "escribió" *ISAS*, estos



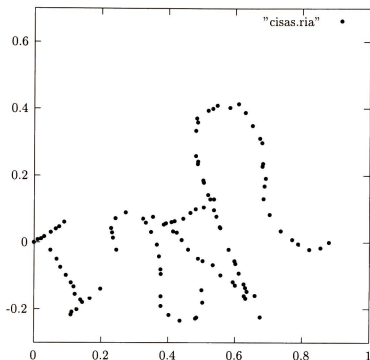


Figura 3.1: Representación isoanalógica obtenida por nuestro algoritmo para el ejemplo *ISAS*.

puntos están muy cercanos, note que la aproximación fue menos precisa, lo cual apoya nuestra idea de tomar los puntos más lejanos para que la representación sea isoanalógica.

La figura (3.6) se obtuvo fijando a  $P_{108}$  y  $P_{109}$ . El valor del funcional en la solución es  $3.2346820 \times 10^{-10}$ , realizó 95 iteraciones y ocupó 3 minutos y 1 segundo de procesador. El mismo comentario que se hizo para el ejemplo anterior se aplica aquí.

La figura (3.7) se obtuvo fijando a  $P_{37}$  y  $P_{50}$ . El valor del funcional en la solución es  $5.2470762 \times 10^{-12}$ , realizó 118 iteraciones y ocupó 4 minutos y 3 segundos de procesador. Esta vez por ser tan "excesivo" el número de itera-



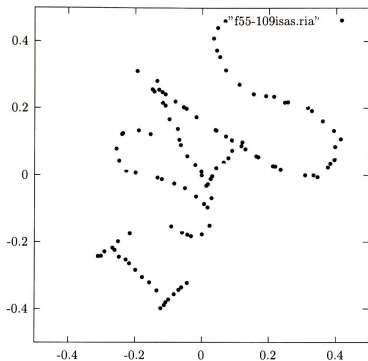


Figura 3.3: Representación obtenida al fijar  $P_{55}$  y  $P_{109}$ , para el ejemplo *ISAS*.

Prueba	Puntos fijados	Error	No. de Iteraciones	Tiempo
1	$P_{15}, P_{86}$	$3.7724719 \times 10^{-12}$	93	0:3:22
2	$P_1, P_{55}$	$6.4183337 \times 10^{-12}$	64	0:2:10
3	$P_{55}, P_{109}$	$5.0113780 \times 10^{-12}$	77	0:2:36
4	$P_{36}, P_{105}$	$6.1023786 \times 10^{-12}$	71	0:2:19
5	$P_1, P_2$	$8.9629609 \times 10^{-10}$	78	0:2:34
6	$P_{108}, P_{109}$	$3.2346820 \times 10^{-10}$	95	0:3:01
7	$P_{37}, P_{50}$	$5.2470762 \times 10^{-12}$	118	0:4:03
8	$P_1, P_{109}$	$5.3959584 \times 10^{-12}$	98	0:3:24

Tabla 3.1: Resumen de los experimentos hechos con el ejemplo *ISAS*.

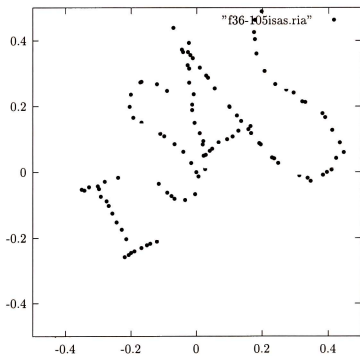


Figura 3.4: Representación obtenida al fijar  $P_{36}$  y  $P_{105}$ , para el ejemplo *ISAS*.

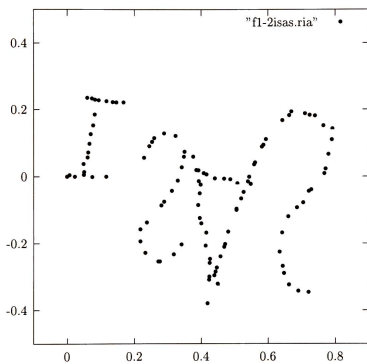


Figura 3.5: Representación obtenida al fijar  $P_1$  y  $P_2$ , para el ejemplo *ISAS*.

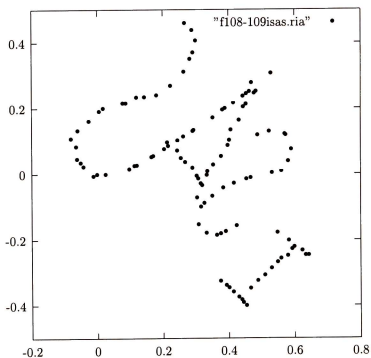


Figura 3.6: Representación obtenida al fijar  $P_{108}$  y  $P_{109}$ , para el ejemplo *ISAS*.

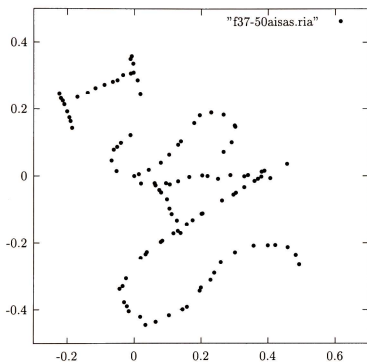


Figura 3.7: Representación obtenida al fijar  $P_{37}$  y  $P_{50}$ , para el ejemplo *ISAS*.

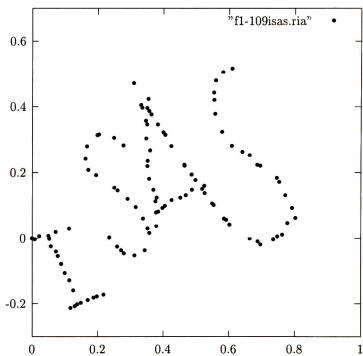


Figura 3.8: Representación obtenida al fijar  $P_1$  y  $P_{109}$ , para el ejemplo *ISAS*.



## Conclusiones

Pocas son las propuestas desarrolladas para resolver el problema de la representación de objetos  $n$ -dimensionales, al hacer la revisión bibliográfica hallamos que ninguno de los métodos reportados era del todo satisfactorio, al menos desde el punto de vista de su aplicabilidad y al abrigo del *ELCRP*; las clasificaciones (anteriormente propuestas) de dichos métodos atienden más a las técnicas matemáticas usadas para resolver el problema que a su espectro de aplicabilidad.

El objetivo de nuestra clasificación creemos que es claro: darle al modelador de problemas reales elementos de análisis, que le permitan elegir el método de representación que mejor se ajuste a las características de su aplicación y/o intereses.

Por este orden de cosas, propusimos las características que, estamos convencidos, debe de cumplir un algoritmo de representación, para que pueda clasificarse como de *Representación Isoanalógica*. Esta definición se ampara en los supuestos epistemológicos de que la realidad es única e independiente de nuestros sentidos.

El reto de construir un algoritmo que cumpliera con nuestra propia definición fue alcanzado a la luz de ella, en el camino fue posible hacer una aportación muy beneficiosa en la práctica: reducir la dimensionalidad del problema a minimizar, originalmente  $2m$  variables, actualmente sólo  $m - 2$  variables.

El análisis de las familias de métodos de optimización no lineal sin restricciones, conllevó a la utilización del denominado método de gradiente conjugado de Polak-Riviere, es el que mejor se ajustó a nuestras necesidades, es decir, minimizar problemas de dimensionalidad “grande” con un funcional no cuadrático.

Experimentalmente se encontró que el algoritmo funciona como el mapeo identidad si el *ERI* es  $\mathbb{R}^2$ . Que es insensible (hablando numéricamente)

al punto inicial de aproximación. Proponemos esto ya que dicho punto es siempre elegido aleatoriamente.

La aproximación más precisa se alcanza fijando a los puntos que corresponden a los objetos con disimilaridad máxima. La menos precisa cuando los puntos fijados tienen disimilaridad mínima. También que dos representaciones son isoanalógicas si las disimilaridades de los puntos fijados es igual, incluso si la disimilaridad con otros puntos es distinta.

No estamos afirmando que es el mejor, sino que, hasta donde sabemos, el único de representación isoanalógica.

Nuestro algoritmo de *Representación Isoanalógica* fue sometido al IV Simposio Iberoamericano de Reconocimiento de Patrones [LMRS99] y aceptado para exposición oral.

## Trabajo Futuro

Siendo la principal crítica a otros métodos el aspecto de aplicabilidad, el siguiente paso a dar es alimentar el algoritmo de Representación Isoanalógica con datos reales, para que pueda ser evaluada su eficacia.

También sería conveniente compararlo con otros métodos no gráficos de clasificación y agrupamiento; para confirmar o rechazar la hipótesis de que bien podría servir como un clasificador, agrupador o validador gráfico en problemas de Reconocimiento de Patrones. Si pasa estas pruebas, será imperativo demostrar rigurosamente los resultados experimentales obtenidos.

Analizar el caso en que no se exija simetría a la función de disimilaridad, como se expuso, este supuesto fue clave para desarrollar nuestro algoritmo.



# Bibliografía

- [B<sup>+</sup>92] Pablo Barrera et al. *Un Punto de Vista sobre el Desarrollo de los Métodos de Gradiente Conjugado No-Lineal*. Editorial Academia, 1992.
- [CL73] C. L. Chang and R. C. T. Lee. A heuristic relaxation method for nonlinear mapping in cluster analysis. *IEEE Trans. Syst., Man, Cybern.*, SMC(3):197–200, March 1973.
- [DB83] J. E. Dennis Jr. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1983.
- [Fle87] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, second edition, 1987.
- [Gow84] K. Chidananga Gowda. A feature reduction and unsupervised classification algorithm for multispectral data. *Pattern Recognition*, 17(6):667–676, 1984.
- [KM83] O. Kakusho and R. Mizoguchi. A new algorithm for non-linear mapping with applications to dimension and cluster analyses. *Pattern Recognition*, 16(1):109–117, 1983.
- [L<sup>+</sup>77] R. C. T. Lee et al. A triangulation method for the sequential mapping of points from n-space to two-space. *IEEE Trans. Comput.*, C(26):288–292, March 1977.
- [LMRS99] J. J. Lavalle-Martínez and J. Ruiz-Shulcloper. Representación isoanalógica de objetos n dimensionales. *Memorias IV Simposio Iberoamericano de Reconocimiento de Patrones, La Habana, Cuba, 22 al 26 de Marzo*, pages 279–292, 1999.

- [NL87] J. M. Navon and David M. Legler. Conjugate-gradient methods for large-scale minimization in meteorology. *American Meteorological Society*, 115(August):1479–1502, August 1987.
- [NW79] H. Niemann and J. Weiss. A fast converging algorithm for nonlinear mapping of high-dimensional data to a plane. *IEEE Trans. Comput.*, C(28):142–147, February 1979.
- [OF73] D. R. Olsen and K. Fukunaga. Representation of nonlinear data surfaces. *IEEE Trans. Comput.*, C(22):915–922, October 1973.
- [RS+95] J. Ruiz-Shulcloper et al. *Introducción al Reconocimiento de Patrones*. Number 51 in Verde. CINVESTAV-IPN, 1995.
- [S+88a] W. Siedlecki et al. Experiments on mapping techniques for exploratory pattern analysis. *Pattern Recognition*, 21(5):431–438, 1988.
- [S+88b] W. Siedlecki et al. An overview of mapping techniques for exploratory pattern analysis. *Pattern Recognition*, 21(5):411–429, 1988.
- [Sam69] J. W. Sammon Jr. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.*, C(18):401–409, May 1969.
- [Sof92] Cambridge University Software. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.
- [SS95] M. Sato and Y. Sato. Extended fuzzy clustering models for asymmetric similarity. *Fuzzy Logic and Soft Computing (Advances in Fuzzy-Applications and Theory)*, B. Bouchon-Meunier, R. Yager, L. Zadeh, Eds., World Scientific., 4:228–237, 1995.
- [Tan93] Andrew S. Tanenbaum. *Sistemas Operativos Modernos*. Prentice Hall, 1993.
- [TM90] C. Tseng and B. M. E. Moret. A new method for one-dimensional linear feature transformations. *Pattern Recognition*, 23(7):745–752, 1990.
- [W+81] S. K. Wismath et al. Feature selection by interactive clustering. *Pattern Recognition*, 14(1-6):75–80, 1981.

Los abajo firmantes, integrantes de jurado para el examen de grado que sustentará el **Sr. José de Jesús Lavalle Martínez** declaramos que hemos revisado la tesis titulada:

**“Representación Isoanalógica de Objetos N Dimensionales”.**

Y consideramos que cumple con los requisitos para obtener el Grado de Maestro en Ciencias en la especialidad en Ingeniería Eléctrica opción Computación.

Atentamente,

Dr. Sergio V. Chapa Vergara



---

Dr. Arturo Díaz Pérez



---

Dr. José Angel Lodegario Ortega Herrera



---

CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL  
INSTITUTO POLITECNICO NACIONAL

**BIBLIOTECA DE INGENIERIA ELECTRICA**  
FECHA DE DEVOLUCION

El lector está obligado a devolver este libro  
antes del vencimiento de préstamo señalado  
por el último sello.

DEVOLUCION





