UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

"Análisis Inteligente de Datos con Redes Neuronales"

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

M A T E M Á T I C O

P R E S E N T A :

ELIO ATENÓGENES VILLASEÑOR GARCÍA

DIRECTOR DE TESIS:
DR. HUMBERTO CARRILLO CALVET

2004

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ACT. MAURICIO AGUILAR GONZÁLEZ
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:

"ANALISIS INTELIGENTE DE DATOS CON REDES NEURONALES"

realizado por ELIO ATENOGENES VILLASEÑOR GARCIA

con número de cuenta 9758906-8 , quien cubrió los créditos de la carrera de: MATEMATICAS

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis
Propietario DR. HUMBERTO CARRILLO CALVET

Propietario DR. LUIS ANTONIO RINCON SOLIS

Propietario M. en C. NIEVES MARTINEZ DE LA ESCALERA CASTELLS

Suplente MAT. SALVADOR LOPEZ MENDOZA

Suplente ACT. JOSE GUADALUPE VAZQUEZ VAZQUEZ

Consejo Departamental de MATEMATICAS

M. en C. ALEJANDRO BRAVO MOJICA
FACULTAD DE CIENCIAS
CONSEJO DEPARTAMENTAL
DE
MATEMATICAS

A mi familia: José, Refugio, Reina, Panchita, Elio, Josefina y Arturo.

Por su confianza y apoyo incondicionales.

A mis amigos: Dany, Ojo, Pazos, Pato, Pavel y Tix.

Por ser como mis hermanos.

A los compadres: Don David, Don Alexei y Don Osvaldo.

Por ser quienes son y ser mis amigos.

A las siempre queridas: Pause y Vary.

Por su cariño y algo más.

A mi director de Tesis: Dr. Humberto Carrillo Calvet.

Por haberme orientado en la realización de este trabajo y en mi formación académica.

A mis sinodales: Nieves, Luis, Salvador y José.

Por sus comentarios y observaciones.

A todos mis maetr@s.

A los integrantes y exintegrantes del LDNL: Agustín, Carolina, Edgar, Gustavo, Heriberto, Itzel, Liliana, Maricarmen, Miguel y Toñito. Especialmente a José Luis por su apoyo en la programación y graficación.

A nuestros amigos cubanos: María Victoria y Gilberto.

A l@s que me faltaron...

A tod@s: ¡ MUCHAS GRACIAS !

Índice

1. Introducción

La tecnología moderna permite la creación de grandes almacenes de datos (crudos) que requieren ser explorados en búsqueda de información refinada (conocimiento). Desarrollar agentes que permitan procesar estos grandes volúmenes de datos y convertirlos en conocimiento útil para la toma de decisiones (inteligencia), constituye un reto colosal. Nuevas disciplinas han emergido y evolucionado para abordar este problema: Análisis Exploratorio de Datos (Exploratory Data Análisis), Análisis Inteligente de Datos (Intelligent Data Analysis), Descubrimiento de Conocimiento (Knowledge Discovery) y Minería de Datos (Data Mining).

Para acometer esta nueva problemática ha sido necesaria la fusión de distintas disciplinas dentro de las ciencias y las ingenierías con la finalidad de desarrollar las denominadas tecnologías de la información, estos desarrollos han tenido un impacto revolucionario en la industria y en el mundo de los negocios [11]. Actualmente existe una gran variedad de sistemas de software comerciales que se basan en las técnicas del *Análisis Inteligente de Datos* para llevar a cabo tareas como: planeación económica, vigilancia e inteligencia empresarial, análisis financiero, análisis de mercados y análisis de perfiles de clientes; entre muchas otras aplicaciones.

En esta tesis se abordará la problemática del análisis de grandes conjuntos de datos multidimensionales, de los cuales no se tiene información previa acerca de las estructuras subyacentes. La principal herramienta a considerar para el procesamiento y el análisis son las denominadas *Redes Neuronales Artificiales*. En particular, se utiliza el modelo propuesto por T. Kohonen [57] denominado *"Self-Organizing Maps"* (SOM).

En el primer capítulo se introduce el campo multidisciplinario del *Análisis Inteligente de Datos*. En la primera sección se presentan las formas de representar los distintos tipos de datos y de establecer relaciones de similitud. Después se examina la naturaleza multidisciplinaria de las herramientas de análisis de datos. Posteriormente se revisan las principales técnicas desarrolladas para el *Análisis Exploratorio de Datos*, entre las que sobresalen: *Visualización de Información*, *Proyección*, *Clustering* y *Vector de Cuantización*.

Otra de las disciplinas que se involucran fuertemente es la denominada *Aprendizaje de Máquina*, esta disciplina está especialmente enfocada en el desarrollo de modelos computacionales que permiten realizar un procesamiento masivo de información y brindan resultados que son utilizados en el análisis de grandes conjuntos de datos. Dentro de los distintos paradigmas para el *aprendizaje de máquina* se pondrá especial atención en las denominadas *Redes Neuronales Artificiales*.

En el segundo capítulo se describe con detalle el modelo de *Red Neuronal* propuesto por Kohonen. Esta descripción parte de las nociones fundamentales que dan origen a este modelo, desde la idea del *aprendizaje no supervisado*, pasando por el concepto de *auto-organización* y la representación del conocimiento por medio de *mapas*; después se construye el algoritmo *SOM básico*, que es el motor matemático de la *red neuronal*; posteriormente se menciona la utilidad

que tiene el uso del SOM en el *Análisis Inteligente de Datos*, principalmente en la exploración de las estructuras subyacentes; es decir, en aquellas estructuras que se establecen a partir de la similitud entre los distintos elementos del conjuntos de datos. Por último, se describen algunas variantes del algoritmo SOM y se discuten algunos aspectos teóricos.

Por último, en el tercer capítulo se describe el proceso de *Descubrimiento de Conocimiento en Bases de Datos* y se pone especial énfasis en la etapa de *Minería de Datos*, ya que en ésta se aplican las herramientas de análisis inteligente de datos. Posteriormente, se describe el funcionamiento de sistema de software *Viscovery SOMine*, el cual tiene como principal motor al SOM y está especialmente diseñado para llevar a cabo *minería de datos* y *visualización de información* de grandes conjuntos de datos. Finalmente, se reporta una aplicación en el análisis de grandes bases de datos con información científica.

La presentación de los distintos temas está encaminada a brindar una idea del gran potencial que tienen estas técnicas en las aplicaciones y un panorama general de las construcciones y la modelación matemática involucradas.

2. Análisis Inteligente de Datos

Gracias a los avances tecnológicos en el área de la computación muchas actividades de la vida cotidiana se han facilitado enormemente; ejemplos visibles son los microprocesadores que controlan los cambios de luces en los semáforos o los hornos de microondas; otros ejemplos bastante más relevantes e interesantes son los procesadores de texto y la gran cantidad de aplicaciones computacionales. Sin embargo, uno de los aspectos más significativos y no tan visible de estos avances tecnológicos, se encuentra en las nuevas formas de interactuar con información subyacente en grandes bases de datos.

En la actualidad, existen una gran cantidad de dominios (campos de investigación o aplicación) donde es posible contar con grandes bases de datos y sistemas que permitan un rápido acceso a estos; al mismo tiempo, el poder de procesamiento y la capacidad de almacenamiento de datos son cada vez son más baratos y eficientes. Dentro de este contexto es de esperar que el análisis de datos adquiera nuevas dimensiones [67].

En general, el término datos se refiere a un conjunto de valores numéricos que representan registros de magnitudes [8]. Estos registros pueden ser obtenidos a partir de un experimento, un muestreo o un censo. Sin embargo, en la actualidad es común considerar otras "formas de datos" como son: texto, imagen, mp3, video, etc. En este trabajo siempre se va a suponer que es posible representar a un conjunto de datos por medio de una matriz o tabla de datos. Esta representación es útil a la hora de implementar transformaciones matemáticas y aplicarlas sobre conjuntos de datos.

Tradicionalmente, el análisis de datos consiste en la aplicación de métodos matemáticos, principalmente estadísticos, con la finalidad de obtener información útil para el mejor entendimiento de por ejemplo: una población determinada o un fenómeno natural. En muchas ocaciones es posible conocer de antemano información *apriori* –distribuciones de densidad, taxonomías de clasificación, relaciones entre variables, etc.– que puede ser utilizada en el proceso de análisis de datos. Sin embargo, cuando se cuenta con una gran cantidad de datos y se conoce poco o nada acerca de la forma o estructuras subyacentes en los datos, en un principio pueden utilizarce herramientas analíticas con la finalidad de explorar el conjunto de datos y averiguar:

- Si los datos tienen una estructura particular.
- Si existen datos que no se ajusten al comportamiento de la mayoría.
- Si los datos presentan algún tipo de agrupamiento.
- Si es posible establecer similitudes o diferencias entre dos o más grupos de datos.

Cuando el análisis de datos está dirigido a responder estas cuestiones se conoce como *Análisis Exploratorio de Datos [92]*. En la respuesta a estas pregunatas, no es suficiente la aplicación de las herramientas analíticas, la pre-

sentación de los resultados arrojados por el análisis de datos es de gran importancia, ya que en muchas ocasiones es necesaria la interpretación de los resulatados. Por lo tanto, es deseable que la información obtenida a partir del análisis pueda ser presentada de manera resumida y ordenada. Con la finalidad de cumplir estos requerimientos han sido desarrolladas una serie de técnicas computacionales para la *Visualización de Información*.

Otro circunstancia interesente de la tecnología actual, es la gran variedad de paradigmas computacionales que se han desarrollado desde mediados del siglo pasado. Con principios distintos a los utilizados por modelos secuensiales (basados en el modelo de Von Newmann), algunos de estos paradigmas parten de la idea de construir modelos computacionales capaces de adaptarse a distintas situaciones de una manera no predeterminada. Estos modelos se funcdamentan en el supuesto de que es posible construir modelos computacionales en los cuales: "las capacidades del sistema se incrementen a medida de que el sistema es sometido a un proceso de entrenamiento". En el desarrollo de estos modelos computacionales surge la disciplina denominada *Aprendizaje de Máquina [73]*.

Estos modelos computacionales han sido exitosamente implementados en computadoras con arquitecturas secuenciales –por ejemplo las computadoras personales– y utilizados con la finalidad de extraer información valiosa subyacente en grandes conjuntos de datos [77]. De la combinación de métodos tradicionales para el análisis de datos, herramientas desarrolladas en el *aprendizaje de máquina* y métodos más específicos; surge el campo denominado **Análisis Inteligente de Datos**.

2.1. Naturaleza y Representación de los Datos

El análisis inteligente de datos involucra la aplicación iterada de algoritmos matemáticos. La aplicación de dichos algoritmos sobre conjuntos de datos normalmente es necesario que los datos estén representados de una manera adecuada. Por lo tanto, dicha representación dependerá del tipo de dato que se esté considerando y del algoritmo.

En esta sección se expondrá la forma general en la cual es posible representar un conjunto de datos; de manera que dicha representación sea apropiada para la aplicación de las técnicas de análisis inteligente de datos; algunas de estas técnicas serán expuestas en las secciones restantes de este capítulo; esta representación será tal que permitirá distintas formas para realizar comparaciones entre los distintos objetos que se puedan definir; así como las relaciones que se establecen a partir de estas comparaciones; por último se definirán algunas funciones de distancia que serán utiles en el establecimiento de relaciones de similitud.

Dado que se pretende que los métodos sean útiles para explorar las estructuras de similitud entre los datos, también es importante tener la capacidad de comparar distintos elementos del conjunto de datos, ya que en muchos casos dichas comparaciones son utilizadas para medir el desempeño de los métodos a la hora de establecer relaciones de similitud dentro del conjunto de datos. Estas relaciones de similitud serán representadas de modo que el analista tenga

la capacidad de interpretarlas y obtener concluciones utiles en el entendimiento del problema correspondiente. El cumplimiento de muchos de los abjetivos en el proceso de análisis de datos depende de lo adecuada que son las representaciones martemáticas de los datos y de las relaciones de similitud entre ellos.

2.1.1. Representación Matemática de Datos

Como ya se ha mencionado, los datos representan mediciones realizadas a partir de la observación de un fenómeno. Cada dato representa la medición de distintas características observadas. Al conjunto $\mathbb{U}$ de todas las observaciones posibles se le denomina **conjunto universo** o **espacio muestral**. En este trabajo se supone que cada elemento del universo puede ser representado como un vector, de manera que las entradas de dicho vector representen las distintas mediciones que se pueden extraer a partir del dato. Por lo tanto, para determinar todos los posibles elementos del universo es necesario conocer los posibles valores que se pueden presentar en cada entrada. Por lo tanto, se considera que cada entrada es una **variable** v con un **rango** R_v. De acuerdo con la naturaleza –topología– de R_v se pueden establecer distintos tipos de variables. Los tipos más importantes son:

- **Cualitativas o Categóricas:** describen características específicas de los objetos. Si los objetos de la población son seres humanos, estas características pueden ser: genero, color de ojos, ocupación, etc. Como caso especial tenemos a las variables *binarias* en donde $R_v = \{0, 1\}$, comúnmente estas variables son usadas para indicar cuando un objeto cumple una propiedad (cierto, 1) o no la cumple (falso, 0).

- **Cuantitativas:** en este caso R_v es un subconjunto de los números reales o incluso $R_v = \mathbb{R}$. En el caso en donde todos los rangos sean de este tipo tenemos que $\mathbb{U} = \mathbb{R}^n$. Estas variables son utiles para representar mediciones de magnitudes tales como: distancia, tiempo, intencidad, probabilidad, entre muchas otras. Cabe señalar que en este trabajo serán consideradas esencialmente este tipo de variables.

- **Cíclicas**: En este caso el rango de la variable es de la forma $R_v = \mathbb{Z}/n\mathbb{Z}$, es decir, el grupo cíclico de orden n. Estas varialbles son ideales para representar situaciones que tienen un comportamiento cíclico y dicho comportamiento está limitado a un conjunto finito de posibles estados. Los ejemplos más simples son la representación del tiempo en minutos dentro de horas, la representación de los días como elementos de la semana o la segmentación del año en estaciones.

Una vez determinado un conjunto de variables $V = \{v_1, v_2, ..., v_n\}$ y sus respectivos rangos R_{v_i}, es posible definir al **conjunto universo** como un subconjunto del producto cartesiano entre estos rangos es decir:

$$\mathbb{U} \subseteq \prod_{i=1}^{n} R_{v_i}$$

tal que toda posible observación está contenida en U. Para este trabajo es de especial interés el caso donde la dimensión del universo n es un espacio multidimensional ($n > 3$). De esta manera, un **conjunto de datos** X es un subconjunto de U, si $X = \{x_1, x_2, ..., x_k\}$ el elemento $x_i \in X$ se representará de manera vectorial como:

$$x_i = (x_{i1}, x_{i2}, ..., x_{in})$$

donde x_{ij} representa el valor de la variable v_j observado en el dato x_i.

En lo que resta de este trabajo se considerará al conjunto X como *conjunto de datos de entrada, datos input* o *conjunto de datos*. Las variables serán llamadas *componentes, atributos* o *características*. Un elemento $x \in \mathbf{U}$ será llamado *objeto, vector de entrada, dato de entrada* o simplemente *dato*; de acuerdo al contexto y a la conveniencia.

Cabe señalar que en las aplicaciones prácticas la selección y el preprocesamiento de los datos puede ser lo más importante ya que en estas etapas es donde precisamente se implementa la representación matemática de los datos; y la primera condición para poder aplicar cualquier método de análisis es que la representación matemática del conjunto de datos sea adecuada. Por ejemplo, cambios en las escalas relativas de las características tienen un efecto drástico en los resultados de la mayoría de los métodos que serán expuestos a lo largo de este trabajo: entre más grande sea la escala de una componente con relación a las demás, más afectará el resultado. "En general, es muy difícil dar una línea o estrategia para realizar el preprocesamiento"[8]. Sin embargo, esta tarea es de suma importancia ya que la utidilidad de los resultados del proceso de análisis dependen más de la representación de los datos que de la técnica o herramienta analítica que se implemente.

2.1.2. Relaciones de Similitud

En está sección se presentará la manera en que matemáticamente se pueden representar las relaciones de similitud entre los elementos del conjunto U. Para comenzar es necesario manejar la noción general de relación binaria entre los elementos de X como un subconjunto de $X \times X$. De todas las posibles relaciones, el primer tipo que vale la pena mencionar es el de relación de equivalencia.

Definición 1 *Una relación binaria* $r \subseteq X \times X$ *es una* ***relación de equivalencia*** *si cumple:*

i) $(x, x) \in r$, *para todo* $x \in X$ ***(reflexividad)***

ii) $(x, y) \in r \Leftrightarrow (y, x) \in r$, *para todos* $x, y \in X$ ***(simetría)***

iii) Si $(x, y) \in r \wedge (y, z) \in r \Longrightarrow (x, z) \in r$, *para todos* $x, y, z \in X$ ***(transitividad)***

Definición 2 *Sea* r *una relación de equivalencia en* X *y* $x \in X$. ***La clase de equivalencia*** *de* x *se define como:*

$$\overline{x} = \{y \in X \mid (x, y) \in r\}.$$

En los problemas de *clasificación* y *clustering* (ver sección 2.3.4) se busca encontrar una forma de dividir al conjunto X en partes disjuntas, para formalizar estas ideas son útiles las siguientes definiciones:

Definición 3 *Sea X un conjunto, el conjunto potencia de X se define como:*

$$P(X) = \{A \mid A \subseteq X\};$$

es decir, el conjunto potencia de X es el conjunto de todos los subconjuntos de X.

Definición 4 *Un partición $\mathcal{C} = \{C_1, C_2, ..., C_K\}$ de X es un subconjunto finito de $P(X)$, tal que:*

i) $X = \bigcup_{a=1}^{K} C_a$.

ii) $C_a \bigcap C_b = \emptyset$ *siempre que* $a \neq b$.

Teorema 5 *El conjunto $\mathcal{C} = \{\overline{x} \mid x \in X\}$ es una partición de X.*

De tal manera que para cualquier $x \in X$ existe un único $C_a \in \mathcal{C}$ tal que $x \in C_a$. A los conjuntos C_a se les llama **clases**. Si $x \in X$ la clase $C \in \mathcal{C}$ tal que $x \in C$ será denotada por $\overline{x}$, de tal manera que $\overline{x} = C$.

Dado un conjunto X con k elementos y $K < k$, existe una gran cantidad de posibles particiones de la forma $C = \{C_1, C_2, ..., C_K\}$ al número total de posibles particiones lo denotaremos por $\left\{ {k \atop K} \right\}$ y para calcular este número, se puede utilizar la siguiente proposición:

Proposición 6 *El número $\left\{ {k \atop K} \right\}$ cumple con la siguiente regla de recurrencia:* $\left\{ {k \atop K} \right\} = n\left\{ {k-1 \atop K} \right\} + \left\{ {k \atop K-1} \right\}$

Una relación de equivalencia es un concepto demasiado general y la partición que induce no necesariamente hace referencia a la similitud entre los objetos que están en una misma clase. Por tal motivo es necesario definir relaciones más particulares que sí hagan referencia a la similitud entre los objetos. Para definir relaciones de equivalencia que consideren la similitud entre los objetos, primero es necesario contar con una forma de medir dicha similitud.

La similitud entre dos elementos $x, y \in X$ estará dada por una función definida como sigue

Definición 7 (Índice de Similitud) *Una función $s : X \times X \longrightarrow [s_{\text{mín}}, s_{\text{máx}}]$ es un índice de similitud si cumple con las siguientes propiedades:*

i) $s(x, x) = s_{\text{máx}}$ *para toda* $x \in X$.

ii) $s(x, y) = s(y, x)$ *para toda* $x, y \in X$.

Donde $s_{\text{mín}}$ y $s_{\text{máx}}$ son la mínima y la máxima similitud entre los elementos de X.

Por ejemplo, cuando $s_{\text{mín}} = 0$ y $s_{\text{máx}} = 1$ en este caso el índice de similitud será llamado función de *dicotomía*. Si se tiene que $x_i, x_j \in X$ y llamamos $s(x_i, x_j) = s_{ij}$ para todo $i, j \in \{1, 2, ..., n\}$, la $S(X)$ matriz de $n \times n$ con componentes s_{ij} será llamada **matriz de similitud**. Una vez definido el índice de similitud se puede definir la siguiente relación

Definición 8 *Sea* $s : X \times X \longrightarrow [s_{\text{mín}}, s_{\text{máx}}]$ *un índice de similitud para* X *y* $\gamma \in [s_{\text{mín}}, s_{\text{máx}}]$. *La relación binaria* $\rho \subseteq X \times X$ *definida por:*

$$(x, y) \in \rho \Longleftrightarrow s(x, y) \geq \gamma$$

ρ es la ***relación de similitud*** *inducida por s con umbral de similitud γ.*

Es fácil observar que la relación binaria ρ, es una relación reflexiva y simétrica, pero no necesariamente transitiva. Por tal motivo, las relaciones de similitud no necesariamente implican una *partición*. La transición entre una relación de similitud y una relación de equivalencia es una de las tareas más importantes a resolver. Dada una relación de similitud ρ, una relación de equivalencia r, es importante poder medir la desviación entre ρ y r. Una manera natural de medir esta desviación se basa en la siguiente definición:

Definición 9 *Sea p una expresión Booleana, es decir aquella que puede se puede determinar se es falsa o verdadera, el* ***símbolo de inversión de Kroneker*** *$[.]$ para p está determinado de la siguiente manera:*

$$\begin{aligned} [p] &= 1 \text{ si } p \text{ es verdadera} \\ [p] &= 0 \text{ si } p \text{ es falsa} \end{aligned}$$

nótese que $[\,]^2 = [\,]$.

Una vez definida la inversión de Kronecker, la desviación entre la relación de similitud ρ y la relación de equivalencia r puede ser medida como el resultado de la siguiente suma:

$$\sum_{\substack{x,y \in X \\ x \neq y}} [[(x, y) \in \rho] \neq [(x, y) \in r]]$$

en un gran número de las aplicaciones es muy conveniente e intuitivo medir la similitud entre los objetos a partir de la distancia en lugar de definir una función de similitud. A continuación se define el concepto de función de distancia.

Definición 10 *Una función* $d : \mathbb{U} \times \mathbb{U} \longrightarrow [d_{\text{mín}}, d_{\text{máx}}]$ *es llamada* ***función de distancia*** *si satisface:*

- $d(x, x) = d_{\text{mín}}$ *para todo* $x \in \mathbb{U}$.

- $d(x,y) = d(y,x)$ *para todo* $x, y \in \mathbb{U}$

donde $d_{\text{mín}}$ representa la distancia mínima y $d_{\text{máx}}$ la máxima.

Definición 11 *Una función de distancia* $d : \mathbb{U} \times \mathbb{U} \longrightarrow [d_{\text{mín}}, d_{\text{máx}}]$ *es* ***métrica*** *si satisface:*

- $d_{\text{mín}} = 0$ *y* $d_{\text{máx}} = \infty$.
- $d(x,z) \leq d(x,y) + d(y,z)$ *para todo* $x, y, z \in \mathbb{U}$. *(desigualdad del triángulo)*
- $d(x,y) = 0 \Rightarrow x = y$ *para todo* $x, y \in \mathbb{U}$.

Como en el caso de las funciones de similitud, si se tiene que $x_i, x_j \in X$ y llamamos $d(x_i, x_j) = s_{ij}$ para todo $i, j \in \{1, 2, ..., n\}$, la matriz $U(X)$ de $n \times n$ con componentes d_{ij} será una matríz simétrica cuyos componentes en la diagonal serán $d_{\text{mín}}$ los cuales serán 0 en el caso de tener una función métrica. Esta matriz es conocida como **matriz de distancias**.

A continuación se introduce un concepto que será de gran importancia a lo largo de este trabajo.

Definición 12 $\mathbb{U}$ *es un* ***espacio métrico,*** *si* $\mathbb{U}$ *es un conjunto que tiene definida una función métrica de distancia* $d : \mathbb{U} \times \mathbb{U} \rightarrow [0, \infty]$.

A partir de este momento se considerará al espacio objeto $\mathbb{U}$ como espacio métrico a menos que se especifique lo contrario. Para el caso de variables cuantitativas, es decir cuando $\mathbb{U} = \mathbb{R}^n$, existe una gran variedad de funciones de distancia. La más conocida es la *distancia Euclidiana* la cual esta definida para elementos del espacio $\mathbb{R}^n$ como sigue:

$$d(x,y) = \|x - y\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1}$$

Sin embargo, la *distancia Euclidiana* es un caso especial de una familia de distancias métricas llamadas *normas* $- L_r$ que son funciones de la forma:

$$d(x,y) = \|x - y\|_r = \left(\sum_{i=1}^{n} |x_i - y_i|^r \right)^{\frac{1}{r}}$$

donde r es un número real positivo. Para el caso $r = 1$ la métrica es llamada *métrica Manhatan* o *taxista* por razones obvias.

Para variables que son medidas en unidades que no son comparables en términos de escala, la siguiente función de distancia es especialmente apropiada

$$d(x,y) = \sqrt{(x-y)^t B^{-1} (x-y)}$$

donde B es una matriz de $n \times n$ invertible definida positiva, esta función es conocida como la *distancia de Mahalanobis.* Nótese que la distancia de Mahalanobis generaliza a la norma Euclidiana ya que esta última se obtiene cuando $B = I$ donde I es la matriz identidad en $\mathbb{R}^n$.

Existen varios ejemplos útiles en donde se pueden relacionar funciones de distancia con funciones de similitud de tal manera que dada una función de distancia d y un rango de similitud $[s_{\text{mín}}, s_{\text{máx}}]$ se pueden determinar una gran cantidad de funciones de similitud. La siguiente proposición hace más clara esta noción.

Proposición 13 *Sea $d(x,y)$ una función de distancia y sea $D : [0,\infty) \longrightarrow [s_{\text{mín}}, s_{\text{máx}}]$ una función monótona decreciente tal que $\lim_{z\longrightarrow\infty} D(z) = 0$ entonces $D(d(x,y)) : X \times X \longrightarrow [s_{\text{mín}}, s_{\text{máx}}]$, es un índice de similitud*

Existen varios métodos para realizar estas transiciones, dada $d(x,y)$ una función de distancia, un ejemplo sencillo es definir $s_{\text{mín}} := -d_{\text{máx}}$, $s_{\text{máx}} := -d_{\text{mín}}$ y la función de similitud de la siguiente manera:

$$s(x,y) = D(d(x,y)) = -d(x,y)$$

Otra forma común de hacerlo cuando se tiene que es utilizando la función de *distancia Euclidiana* en $\mathbf{R}$ dada por $d(x,y) = |x-y|$ que tiene valores, $d_{\text{mín}} = 0$ y $d_{\text{máx}} := \infty$, teniendo $s_{\text{mín}} := 0$, $s_{\text{máx}} := 1$ y usando la función exponencial de la siguiente manera

$$s(x,y) = e^{\frac{-|x-y|}{\Delta}}$$

donde $\Delta \in \mathbf{R}$ es una constante positiva. Cabe señalar que esta función de similitud será retomada en el capítulo 3 en donde será implementada en un algoritmo que es útil para la determinación automática clases de similitud dentro del conjunto X. Una vez establecida la representación matemática del conjunto X y las relaciones de similitud entre sus elementos, se puede considerar la aplicación de métodos matemáticos que sean útiles para el establecimiento de **clases de similitud** dentro del conjunto de datos.

2.2. Herramientas Matemáticas de Análisis

El **análisis inteligente de datos** (IDA) es el estudio interdisciplinario concerniente al análisis de datos efectivo y eficiente. En el desarrollo de las herramientas para el IDA, las disciplinas que juegan un papel fundamental son: la *estadística* y el *aprendizaje de máquina*. Por un lado, la estadística tiene sus raíces en las ciencias matemáticas; su principal actividad consiste en el establecimiento de modelos que describan de manera general las características de la distribución del conjunto de datos dentro del universo. Estos modelos deben ser sometidos rigurosas demostraciones matemáticas antes de ser puestos en práctica. Por otro lado, el estudio del *aprendizaje de máquina* tiene su origen en prácticas computacionales, el sentido de esta disciplina es netamente práctico y tiene la disposición de abordar problemas sin esperar a que existan pruebas formales para evaluarlos [8].

En el resto de la sección se presentará un panorama general de cada una de estas disciplinas poniendo especial énfasis en su relación con el análisis inteligente de datos, posteriormente se abordará el papel de las matemáticas y de los matemáticos dentro de este nuevo contexto científico y tecnológico.

2.2.1. *Aprendizaje de Máquina*

En la actualidad existen una gran cantidad de proyectos multidisciplinarios de investigación que tienen como principal objetivo el desarrollo y la aplicación de métodos para el *aprendizaje de máquina*, la investigación se enfoca tanto en la teoría del aprendizaje, como en el mejoramiento de los resultados en la solución de problemas en dominios diversos.

Los investigadores parten de la idea de que el aprendizaje es un prerrequisito para cualquier forma de inteligencia verdadera, por esta razón el *aprendizaje de máquina* tiene un papel muy importante en el desarrollo de la denominada inteligencia *artificial*. A pesar de lo controversial de los términos, lo cierto es que las actuales capacidades de las computadoras sorprenderían a cualquiera de los grandes genios de épocas pasadas. El debate filosófico ha sido arduo (para una visión general de las dos posturas revizar [13] y [12]). Sin embargo, en la práctica las máquinas han demostrado ser capaces de realizar tareas que de ser realizadas por humanos requieren el uso de cierta inteligencia. Aún cuando en este punto existe un debate no resuelto, en este trabajo se partirá de una noción operativa del aprendizaje, es decir, un aprendizaje encaminado a adquirir la habilidad de realizar tareas específicas. De manera que por **aprendizaje** se entiende un proceso por medio del cual una máquina adquiere la capacidad de realizar tareas que antes no podía realizar o realizar de mejor manera tareas resueltas previamente. Cabe señalar que este aprendizaje es definido por medio de modelos matemáticos; esencialmente *sistemas dinámicos discretos no lineales*.

En el desarrollo del *aprendizaje de máquina* existen distintos paradigmas que surgen o se basan en diversas ramas de las ciencias, cada uno de estos emplean sus propios métodos y criterios para evaluar el éxito o fracaso del proceso. Sin embargo, todos comparten la meta de construir máquinas que puedan aprender a realizar tareas complejas de un modo eficiente. A continuación se describen brevemente las características generales de los cuatro paradigmas vigentes en la investigación y el desarrollo de estos métodos [73], [16]:

- **Inductivo:** este enfoque tiene como meta el inducir una descripción general de un concepto a partir de una secuencia de instancias y contraejemplos del concepto. Dado un conjunto de ejemplos, la tarea es construir un conjunto de descripciones entendibles de conceptos que clasifiquen correctamente nuevas instancias.

- **Analítico:** estos métodos son deductivos en lugar de inductivos ya que utilizan la experiencia de los problemas resueltos para resolver nuevos problemas. Las instancias corresponden a porciones de la solución de un problema y el aprendizaje utiliza estas instancias junto con el conocimiento previo para deducir la solución a los problemas.

- **Genético:** Este paradigma está basado en los métodos evolutivos. Variantes en la descripción de un concepto corresponden a individuos de especies. Los cambios inducidos y recombinaciones de estos conceptos son probadas contra una función objetivo (criterio de selección natural) para determinar que variantes prevalecen.

- **Conexionista:** Estos sistemas, también llamados *redes neuronales artificiales* (RNA) o *sistemas de procesamiento paralelo y distribuído*, estos sistemas están tienen como principal modelos computacional el cerebro humano. De manera que estos modelos emulan la forma en la cual las neuronas del cerebro procesan información a lo largo de una red de neuronas, en el caso de las RNA las neuronas son sistemas dinámicos en los cuales las variables de estado están en función de la interacción con otras neuronas. Este paradigma será abordado con mayor detalle en la sección 2.4.

Dado el enfoque esencialmente práctico de esta disciplina, la mayoría de estos métodos tienen la propiedad de operar con algoritmos computacionalmente poco complejos; sin embargo, en muchos casos no existen pruebas formales que garanticen la validez de sus resultados. "En el análisis inteligente de datos lo ideal es establecer resultados mediante la experiencia adquirida con la práctica y al mismo tiempo trabajar en la demostración matemática y la validación de los resultados" [8]. Desafortunadamente esta potencial combinación no ha sido lo suficientemente reconocida y explotada en el desarrollo actual de esta disciplina.

2.2.2. El Enfoque Estadístico

Por otro lado, la estadística es la matemática de la recolección, la organización y la interpretación de datos numéricos; especialmente enfocadas en el análisis de poblaciones donde se pretende inferir alguna característica de una población a partir del análisis de una muestra [8]. Dadas las características de la estadística es de esperarse que a consecuencia de la revolución informática, la estadística haya sufrido cambios importantes; tanto en el tipo de problemas que trata; como en la manera de aplicar sus técnicas.

A mediados de la década de los sesentas la comunidad de estadística se refería despectivamente a la libre exploración de datos como: pescar o rastrear datos. Los especialistas en los métodos clásicos de análisis -enamorados de la elegancia en las soluciones matemáticas a los problemas de inferencia- argumentaban que los métodos exploratorios -que tenían como punto de partida el mirar en los datos- carecían de formalidad matemática. En este enfoque clásico de la estadística normalmente se supone la existencia de un **modelo** que describe de una manera simple el comportamiento de los datos. Estos modelos representan estructuras a gran escala que resumen las relaciones entre todos los datos. Actualmente, los modernos sistemas de computación, tanto en hardware como en software, han liberado a los especialistas en estadística de la búsqueda minuciosa de modelos. De manera que, los nuevos analistas de datos pueden estar más interesados en el entendimiento de las estructuras subyacentes en grandes conjuntos de datos; que en el análisis de la fidelidad de un modelo predeterminado.

Como oposición a la concepción clásica del análisis de datos y por la necesidad de contar con métodos para la exploración de grandes conjuntos multidimensionales, surge la escuela del **análisis exploratorio de datos** (EDA). Este

planteamiento, con origen en el trabajo de Tukey [92], fue más propositivo que su contraparte tradicional, su principal fundamento es que tanto la teoría como la aplicación de la estadística debe apegarse al método científico antes que a otra cosa; por lo tanto, el primer paso en el análisis de datos debía ser la observación y la generación de hipótesis acerca de propiedades subyacentes en el conjunto de datos.

La práctica del análisis de datos cada vez está más enfocada en la clasificación y el reconocimiento de patrones; que en ajustar un conjunto de datos a un modelo predeterminado. Es decir, cada vez son más útiles y urgentes los métodos que ayuden a generar buenas hipótesis acerca de los comportamientos locales en las relaciones de similitud entre los datos; y basándose en estas relaciones, establecer subconjuntos de datos que compartan propiedades específicas.

Por lo tanto, muchos de estos métodos están enfocados al reconocimiento de **patrones**; es decir, estructuras locales o combinación de eventos que se presentan en un subconjunto de los datos, la cual ocurre más de lo esperado y se mantiene en datos nuevos.

Por otro lado, dentro del contexto actual del análisis de datos, son comunes las aplicaciones que involucran un gran número de variables y un vasto número de datos. Por tal motivo, la eficiencia computacional y la escalabilidad son temas de gran importancia para la aplicación de estos. Desafortunadamente, las técnicas desarrolladas dentro del enfoque estadístico, normalmente involucran operaciones computacionalmente costosas. En consecuencia, la aplicación directa de estos métodos sobre grandes conjuntos de datos, en muchos casos resulta inoperante. En el siguiente capítulo se detallan algunas de las técnicas más utilizadas en el análisis exploratorio de datos.

2.3. Análisis Exploratorio de Datos

En esta sección serán expuestos métodos clásicos útiles para ilustrar estructuras o relaciones multivariadas entre los elementos de conjuntos de datos multidimensionales. En la mayoría de estos métodos se supone que el conjunto de datos pertenece al espacio métrico $\mathbb{R}^n$ y en general no se sabe nada acerca de la distribución y las relaciones de similitud entre los datos. También se asume que no se conoce ninguna forma explícita de clasificación del conjunto de datos.

Los resultados de estos métodos dependerán únicamente de las estructuras subyacentes en el conjunto de datos; y no de presuposiciones acerca de alguna estructura de clasificación o algún modelo de distribución. En el caso de saber o suponer como se clasifican los datos; este conocimiento normalmente se utiliza después de haber aplicado los métodos exploratorios y de esta manera se enriquece la interpretación de los resultados. En lo que resta del capítulo se considera al conjunto de datos de entrada como:

$$X = \{x_1, x_2, ..., x_k\} \text{ donde } x_i \in \mathbb{U} = \mathbb{R}^n \text{ para toda } i = 1, ..., k$$

el principal objetivo de estos métodos es ilustrar estructuras dentro de X. En los casos en los que la dimensión del conjunto es tal que $n < 4$, el problema

de ilustrar las estructuras se reduce a un problema de graficación. Sin embargo, se está suponiendo que $n \geq 4$, por los tanto los métodos de visualización de los resultados deben ser capaces de representar visualmente estructuras multidimensionales. En general, al aplicar alguno de estos métodos las siguientes preguntas jugarán un papel central : ¿Qué clase de estructuras es posible extraer a partir de un método dado? ¿Es útil y viable la aplicación de un método para reducir la dimensionalidad del conjunto? ¿Es posible reducir el número de datos a ser considerados por medio del establecimiento de clases de equivalencia?. En las respuestas de estas preguntas se encontrará gran parte de la estrategia a seguir para el procesamiento del conjunto de datos.

2.3.1. Visualización de Información

La exploración visual de información en espacios complejos es uno de los temas de mayor interés dentro de la investigación de graficación por computadora. Existe una gran variedad de distintos paradigmas y distintos métodos han sido desarrollados en los últimos años.

La visualización de información puede ser entendida como un proceso asistido por la computadora, en el cual busca revelar señales de un fenómeno abstracto al transformar datos en formas visuales [15]. La intención de la visualización de información es optimizar el uso de nuestra percepción y la habilidad de nuestro pensamiento para tratar con fenómenos que por si solos no pueden ser representados visualmente en un espacio bidimensional [10]. El uso de métodos adecuados para la visualización de grandes conjunto de datos puede tiene una importancia estratégica para los métodos de análisis inteligente de datos .

El proceso de visualización comienza con información en forma de datos. Existen muchas formas en las que los datos pueden aparecer y para cada tipo de dato deben ser diseñadas transformaciones de preprocesamiento específicas que permitan la aplicación de algún algoritmo de visualización adecuado, de manera que efectivamente se representen visualmente el tipo de estructuras que se está buscando [17].

Un problema elemental de la visualización es el de representar visualmente elementos de un espacio multidimensional. Para la solución de este problema muchos medios gráficos han sido propuestos para visualizar directamente vectores de dimensión alta [47]. La idea central de estos métodos es la de asignar a cada componente del vector algún aspecto de la visualización. De esta manera los valores de las componentes son integrados en una figura o ícono. Estos métodos pueden ser usados para visualizar cualquier tipo de vector de datos y proveen una representación visual de elementos provenientes de espacios multidimensionales.

Tal vez el método más simple para visualizar un vector de datos sea graficar un **perfil** o **histograma**, es decir, una gráfica bidimensional en la cual cada una de las componentes están enumeradas en el eje las abscisas y los correspondientes valores en el eje ordenadas. Un método muy similar es el de la **curva de Andrews** [3], en él una curva es asignada a cada dato; esta asignación es por medio de los valores en las componentes de los vectores de datos que son

utilizados como coeficientes de una senoidal, de manera que estas curvas son puestas juntas unidas por un punto intermedio. Otro ejemplo es dejando que las componentes representen la longitudes de los rayos emanantes del centro de una **estrella**.

El más famoso de estos despliegues visuales son las **caras de Chernoff** [18]. Cada dimensión determina el tamaño, localización o forma de algún componente de una caricatura facial. Por ejemplo, una componente es asociada con la boca, otra con la separación de los ojos, otra con el tamaño de la nariz, etc. En la figura 1 se puede observar la aplicación de cada uno de estos métodos a un vector de datos de dimensión 10. Una vez elegido el método de representación visual,

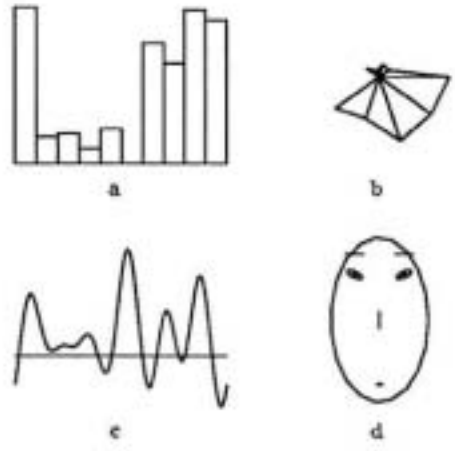

Figura 1: Técnicas de Visualización de Datos: (a) Histograma o Perfil, (b) Estrella, (c) Curva de Andrews, (d) Cara de Chernoff.

cada dato quedará representado por un ícono. Para la visualización de todo el conjunto de datos una alternativa es la utilización de una gráfica bidimensional de dispersión. En esta gráfica se deja que dos componentes de los vectores de datos sean asignados a la localización de un ícono en el plano cartesiano y el resto de las dimensiones serán asignadas a propiedades del ícono. En la figura 2 se puede apreciar la aplicación de este método utilizando caras de Chernoff.

La mayor desventaja de estos métodos es que no reducen la cantidad de datos. Si el conjunto de datos es muy grande el despliegue consistente de las representaciones todos los datos presentados por separado será muy difícil de interpretar o incluso incomprensible. Sin embargo, los métodos pueden ser útiles para ilustrar algunos sumarios producidos a partir del conjunto de datos. Entre los principales problemas a los que se enfrentan los desarrolladores de métodos para la visualización de la información se encuentran [17]:

- **Reducir la información y encontrar estructuras:** la exploración

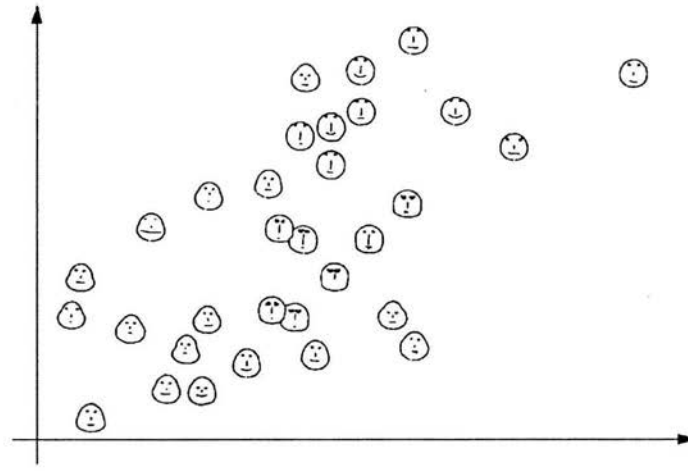

Figura 2: Diagrama de Dispersión usando caras de Chernoff

de grandes espacios de información sin estructura requiere de preprocesamiento para lograr reducir el tamaño de los datos activos. Este preprocesamiento puede implicar filtrar campos poco interesentes o agrupar datos similares en grupos homogéneos. Además métricas o medidas de similitud apropiadas para el tipo de datos, deben ser implementadas para poder representar las estructuras que se están buscando.

- **Visualización de conjuntos de información:** el éxito de la visualización depende en gran medida de su habilidad para soportar una gran variedad de tareas de exploración, por ejemplo: revisiones globales, acercamientos en temas específicos, etc. Es común que distintos métodos de visualización se requieran para revelar estructuras y contenidos.

- **Visualización del marco de referencia:** para lograr exploraciones efectivas dentro de espacios referenciados se requiere la combinación de un despliegue adecuado de los marcos de referencia espaciales junto con la visualización de estructuras complejas.

A la fecha, las técnicas de visualización de información aún no han sido suficientemente explotadas y desarrollo se encuentra en una etapa inicial, una de las principales dificultades es la gran variedad de tipos de datos y la falta de algoritmos de preprocesamiento adecuados [65]. En las secciones restantes se expondrán algunas técnicas clásicas que pueden ayudar a resolver algunos problemas de los problemas planteados: reducir la dimensión de los vectores (métodos de proyección), encontrar grupos homogéneos en los datos (clustering) y determinar representantes para regiones del espacio $\mathbb{U}$ (vector de cuantización).

2.3.2. Proyección y Reducción de Dimensión

Existen una gran variedad de métodos que pueden ser usados para reducir la dimensionalidad de los vectores de datos. La idea central de estos métodos es la de encontrar subespacios o simples subconjuntos $\mathcal{M} \subseteq \mathbb{U}$ (preferentemente visibles) en las cuales tenga sentido proyectar el conjunto de datos; una **proyección** entre $\mathbb{U}$ y una variedad $\mathcal{M}$ queda definida por una función inyectiva entre el universo y la variedad. A este tipo de herramientas se le conoce como métodos son llamados métodos de proyección.

El objetivo de la proyección es representar los vectores de datos de entrada en un espacio de dimensión baja. Es decir, la proyección debe ser tal que preserve aspectos interesantes e inherentes a la forma en la que los datos se distribuyen en el espacio multidimensional; de tal modo que las propiedades características de la estructura del conjunto de datos se preserven lo más nítido que sea posible. Además, la proyección puede ser usada para visualizar el conjunto de datos si el espacio codominio es de una dimensión baja.

Dada la generalidad del problema que se plantea en los métodos de proyección, existen una gran cantidad de enfoques sobre los cuales se han construido una gran variedad de métodos; por tal razón es difícil dar una clasificación precisa y desglosada de los mismos. Sin embargo, la simplicidad del leguaje matemático permite identificar dos tipos de métodos: los lineales y los no-lineales. A continuación se detalla más acerca de estos dos tipos de proyecciones.

Métodos Lineales

Los métodos de proyección lineales son aquellos en donde se parte del supuesto de que el conjunto universo U es un espacio vectorial, El objetivo de estos métodos consiste en encontrar un subespacio lineal de dimensión baja en el cual sea proyectado el conjunto de datos por medio de una transformación lineal entre el espacio universo y el subespacio objetivo.

El más famoso de estos métosdos es el **análisis por componentes principales** (ACP), el cual tiene su origen en un trabajo de Karl Pearson en 1901 y no fue sino hasta 1933 que Harold Hotelling [45] lo desarrolló con todo el rigor matemático. Se trata de una técnica descriptiva, por lo que no se requiere usar un modelo estadístico predeterminado, en particular no se hace ninguna suposición acerca de la distribución de probabilidad de las variables originales, aunque se puede dar más significado a las componentes en el caso donde se supone que las observaciones provienen de una normal multivariada.

La idea básica del método es transformar un conjunto de variables multivariadas (datos multidimensionales) altamente correlacionadas en un conjunto menor de variables no correlacionadas llamadas **componentes principales**.

Dichas componentes son una combinación lineal particular de las variables de origen, las nuevas variables se ordenan de forma decreciente en función de su importancia; es decir, la primer componente principal aporta la mayor variabilidad posible con respecto a los datos originales, la segunda componente principal aporta menos variabilidad que la primera, pero más que la tercera componente; así, la última componente principal, aportará la mínima cantidad de variabilidad con respecto a las demás componentes.

Entre los objetivos más importantes del ACP se menciona la reducción de la dimensionalidad de los datos. Partiendo del hecho de que los datos son multidimensionales, se desea averiguar si en realidad dichos datos poseen información tan significativa que merezca ocupar las n dimensiones. Si un ACP revela que la dimensión de los datos es menor que n, entonces se reemplazan las variables originales por las componentes principales y la dimensión se reducirá descartando y por consiguiente desechando las componentes menos importantes, quedando solo las primeras, que conservan la mayor parte de la variabilidad (respecto a los datos de origen), lo cual se traduce en información. Otro de los propósitos del ACP es hallar grupos en los datos, esto se logra al graficar las calificaciones de las componentes principales.

Las principales desventajas de los métodos lineales radican en el supuesto de que el espacio universo es un espacio vectorial y en su limitante a proyectar el conjunto de datos sobre subespacios lineales. De manera que son poco adecuados en los casos donde la estructura global del conjunto de datos es sumamente no-lineal. Además, en el caso del método de análisis por componentes principales, se requiere el cómputo de determinantes y la inversión de matrices de $n \times n$. Estas operaciones suelen tener una alta complejidad computacional, lo que implica que es muy difícil (o tal vez imposible) aplicarlas sobre conjuntos cuya dimensionalidad rebasa el orden de las centenas.

Métodos no Lineales

El análisis de componentes principales no puede tomar en cuenta estructuras no lineales; por ejemplo, estructuras que constan de variedades curvas; ya que describen los datos en términos de un subespacio lineal. Distintos enfoques han sido propuestos para reproducir estructuras no-lineales de dimensión alta reproduciéndolas en un espacio de dimensión menor. Los métodos más comunes determinan una representación de cada dato en un espacio de dimensión menor y tratan de optimizar estas representaciones de tal manera que las distancias entre los puntos sean lo más similar posible a las distancias correspondientes a los vectores de referencia de los datos originales. Los métodos difieren en como las distancias distintas son ponderadas y en como son optimizadas las representaciones.

Uno de los ejemplos más famosos es el denominado **escalamiento multidimensional** (MDS) este tipo de técnicas se refiere a todo un conjunto de métodos, cuyo objetivo no consiste únicamente de crear un espacio en el cual se puedan representar fielmente las relaciones entre los datos; sino también reducir la dimensionalidad del conjunto de datos a una suficientemente pequeña que permita la inspección visual.

Existe una multitud de variantes del MDS con algunas diferencias en las metas y algoritmos de optimización. Sin embargo, se pueden clasificas en dos tipos básicos: *métricos* y *no-métricos*.

En el **MDS métrico** original [96] la matriz de distancias entre los vectores de datos están dadas (matriz de distancias) y se busca una configuración de puntos en una dimensión menor que la original que representen a los datos y distancias originales. Dados $X = \{x_1, x_2, ..., x_k\} \subset \mathbb{R}^n$ con $n > 2$ a cada $x_i \in X$

se le asocia con un vector $x_i' \in \mathbb{R}^2$. Sean $x_i, x_j \in X$ con $d(x_i, x_j) = \|x_i - x_j\|$ y los respectivos $x_i', x_j' \in \mathbb{R}^2$ con $d(x_i, x_j) = \|x_i' - x_j'\|$. El MDS original se trata de aproximar la distancia $\|x_k - x_l\|$ mediante $\|x_k' - x_l'\|$. Si el error cuadrado es usado para la **función de costo** a minimizar esta se escribe como:

$$E_M = \sum_{1 \leq i < j \leq k} [d(x_i, x_j) - d(x_i', x_j')]^2$$

Una reproducción perfecta de las distancias Euclidianas no siempre será la mejor, especialmente si las componentes de los datos son expresan un orden de escala ordinal. En este caso, el orden en el rango de las distancias entre vectores es significativo y no los valores exactos.

Otro método interesante del de **análisis de curvas principales** (PC) [39] que puede ser visto como una generalización del método de ACP, ya que en este método se buscan **variedades no lineales**; en lugar de únicamente subespacios lineales. Mientras en PCA una buena proyección del conjunto de datos sobre una variedad lineal es construida, el objetivo de construir la curva principal es proyectar el conjunto en una *variedad no-lineal* o *hipersuperficie curva*.

Las curvas principales son curvas suaves que son definidas con la propiedad de que cada punto de la curva es el promedio de todos los puntos de datos que son proyectados en él; es decir, para el cual ese punto es el punto más cercano en la curva. Hablando intuitivamente, las curvas pasan a través del centro del conjunto de datos.

Las curvas principales son generalizaciones de los componentes principales que son extraídos usando PCA en el sentido de que una *curva lineal principal* es un componente principal. Las conexiones entre los dos métodos son delineadas más cuidadosamente en el artículo [39]. A pesar de que la estructuras extraídas son llamadas *curvas principales* la generalización a superficies parece relativamente sencilla, sin embargo, los algoritmos resultantes se vuelven computacionalmente más intensos.

2.3.3. Clustering

El objetivo del clustering es reducir la cantidad de datos mediante la caracterización o agrupamiento de datos con características similares. Esta agrupación es acorde con los procesos humanos de información y una de las motivaciones para usar algoritmos clustering es proveer herramientas automáticas que ayuden a la construcción de taxonomías. Los métodos pueden también ser usados para minimizar los efectos de los factores humanos que afectan el proceso de clasificación.

Como se puede observar, el problema de clustering consiste en encontrar una partición óptima en el sentido que los elementos de la partición se entiendan como clases o tipos de datos. De acuerdo con [78] "los **algoritmos de clustering** son métodos para dividir un conjunto X de k observaciones en K grupos de tal manera que miembros del mismo grupo son más parecidos que miembros de distintos grupos". Para dejar claro el concepto de clustering se dan las siguientes definiciones.

Definición 14 (Clustering) *Un **clustering** de X es un conjunto $\mathcal{C} = \{C_1, C_2, ...C_K\}$ tal que $\mathcal{C}$ es una partición de X que se obtiene por medio de la aplicación de un **algoritmo de clustering**. A cada elemento $C \in \mathcal{C}$ del clustering se le denomina **cluster**.*

Dado que un *clustering* es una partición, esta determina una relación de equivalencia entre los elementos de X. Dado un *cluster* $C \in \mathcal{C}$, existe mucha información útil que se puede extraer, ejemplos sencillos son los siguientes:

Definición 15 (Objeto de referencia) *Un objeto $x_C \in \mathbb{U}$ es **objeto de referencia** del cluster C si este es, de alguna manera, representativo de los elementos de C.*

Nótese que un *objeto de referencia* no tiene porque ser elemento del *cluster*, *sin embargo, al ser elemento de* $\mathbb{U}$ el *objeto de referencia* puede ser comparado con los elementos de X. Cuando los objetos son vectores, como en este caso, el *objeto de referencia* también es llamado *vector de referencia*. Este término será utilizado en otros contextos con significados similares. En el caso de los *clusterings* un objeto de referencia bastante utilizado es el siguiente:

Definición 16 *El **centro de gravedad de un cluster** C es un vector $g(C) \in \mathbb{R}^n$ obtenido de la siguiente forma*

$$g(C) = \frac{1}{\#C} \sum_{x \in C} x.$$

En general, dado un conjunto A, la notación $\#A$ representará la cardinalidad de dicho conjunto.

Otra información valiosa que se puede extraer a partir de un clustering es la distribución de probabilidad que implícitamente queda determinada en términos de las cardinalidades de las clases:

Definición 17 (Probabilidad a priori) *Sea $\mathcal{C}$ un clustering de X, la función de **probabilidad a-priori** $\mathbb{P}_C$ para $\mathcal{C}$ se define como:*

$$\begin{aligned} \mathbb{P}_{\mathcal{C}} : \mathcal{C} &\longrightarrow [0,1] \\ C &\longrightarrow \frac{\#C}{\#X} \end{aligned}$$

En muchas situaciones prácticas será de gran utilidad contar con formas de medir la similitud o la **distancia entre un elemento $x \in X$ y un cluster $C \in \mathcal{C}$**. Una forma sencilla de comparar un objeto con un cluster es a partir de la función de distancia en el espacio $\mathbb{U}$; y calculando la distancia entre el objeto x y un *objeto de referencia* x_C del cluster. Si se considera como distancia a la norma Euclidiana y como *objeto de referencia* x_C al centro de gravedad $g(C)$, tenemos que la distancia entre un objeto x y un cluster C puede ser definida como:

$$d(x, C) = \|x - g(C)\|$$

Así como se define la similitud de un cluster C con un objeto x, también es útil medir la **homogeneidad intra cluster**, es decir medir que tan similares son los elementos dentro de un cluster. Existen varias formas de hacer esta medición. Cuando se tiene un índice de similitud s con rango $[s_{\text{mín}}, s_{\text{máx}}]$ para los objetos en X, la homogeneidad se puede definir como una función h tal que

$$h : P(X) \longrightarrow \mathbb{R}$$

$$h(C) = \frac{1}{\binom{\#C}{2}} \sum_{x,y \in C, x \neq y} s(x,y) \tag{2}$$

donde $\binom{\#C}{2}$ es el número de elementos en $C \times C$, de tal forma que (2) es un promedio de las similitudes entre elementos de C.

En ocasiones es conveniente no considerar el factor de escalamiento $\frac{1}{\binom{\#C}{2}}$ de manera que también se puede encontrar

$$h(C) = \sum_{x,y \in C, x \neq y} s(x,y),$$

otra forma de medir la homogeneidad es:

$$h(C) = \min_{x,y \in C, x \neq y} s(x,y);$$

una mejoramiento en la complejidad computacional de la estimación de $h(C)$ puede obtenerse si se considera un *objeto de referencia* x_C para el cluster C. Entonces, en lugar de comparar todos los pares de elementos solo se tienen que comparar todos los elementos de un cluster con el *objeto de referencia*

$$h(C) = \sum_{x \in C} s(x, x_C)$$

un ejemplo bastante común es usar como objeto de referencia al *centro de gravedad* $g(C)$ del cluster.

En los ejemplos expuestos se espera que el valor de h sea cercano a $s_{\text{máx}}$. Sin embargo, cuando se usa una función de distancia en lugar de un índice de similitud, se espera que el valor de h sea mínimo. Un ejemplo de esta forma de definir la homogeneidad es la *inercia intra-clase* que también hace uso del centro de gravedad y se define de la siguiente manera:

$$h(C) := \sum_{x \in C} \|x - g(C)\|.$$

Otro aspecto importante a considerar es el que se refiere a la **comparación entre dos clusters distintos**. Si se tiene un índice de similitud s para comparar los objetos en X este puede ser extendido para comparar clusters en el clustering $\mathcal{C}$ por medio de una función

$$S : \mathcal{C} \times \mathcal{C} \longrightarrow \mathbb{R}$$

Una forma de hacerlo es utilizando una vez más la noción de promedio de tal forma que dados $C, D \in \mathcal{C}$ la similitud entre estos dos clusters se puede ser medida de la siguiente manera:

$$S(C,D) = \frac{1}{\#C\#D} \sum_{x \in C} \sum_{y \in C} s(x,y)$$

también puede ser conveniente eliminar el factor de escalamiento y obtener

$$S(C,D) = \sum_{x \in C} \sum_{y \in C} s(x,y)$$

Otro modo de hacerlo es simplemente tomando los valores mínimo o máximo. Si consideramos el mínimo tenemos que:

$$S(C,D) = \min_{x \in X, y \in Y} s(x,y)$$

esta medida es conocida como *conexión simple* o *vecindad más cercana*.

En el caso de considerar el valor máximo tenemos que:

$$S(C,D) = \max_{x \in X, y \in Y} s(x,y)$$

esta otra medida es conocida como *conexión completa*.

Finalmente, el centro de gravedad también puede ser utilizado de tal forma que

$$S(C,D) = \|g(C) - g(D)\|$$

Dadas las funciones de similitud y distancia para las distintas clases de objetos a considerar, se observa que existe una gran cantidad de particiones que se pueden derivar de las relaciones inducidas por estas funciones. Por tal motivo, es muy importante contar con una función que sea capaz de evaluar a un clustering dado.

Definición 18 (Criterio) *Una función que a cada clustering asigna un valor real, es llamada* ***criterio de cluster*** *y se denotará por*

$$\mathrm{c} : \{\mathcal{C} \mid \mathcal{C} \text{ es un } \textbf{\textit{clustering}}\} \longrightarrow \mathbb{R}$$

Una condición previa para un *criterio* es que éste modele la aplicación de manera adecuada. En general se espera que un criterio evalúe como bueno a un clustering que tenga clusters con una alta homogeneidad en su interior y una alta separabilidad de su exterior. Existe una gran cantidad de ejemplos en donde se construyen criterios a partir de funciones de similitud o distancias. Los **métodos de clustering** pueden ser divididos en dos tipos básicos: jerárquicos y particionales [37]. Con cada tipo de clustering existe una variedad de subtipos y diversos algoritmos para encontrar los clusters.

Un método **clustering jerárquico** procede asociando clusters pequeños dentro de otros más grandes. Los métodos de clustering difieren en la regla

por la cual se decide como asociar clusters pequeños para que pertenezcan al mismo cluster grande. El resultado final del algoritmo es un árbol de clusters llamado dendograma, cada nivel del arbol representa un *clustering*, de manera que cortando el dendograma en un nivel deseado un clustering es obtenido. El conjunto de clusterings contenidos en una dendograma es llamado **jerarquía**.

En los métodos de **clustering particional** se trata de descomponer directamente los datos en conjuntos de clusters disjuntos. La función criterio que el algoritmo de clustering trata de minimizar suele enfatizar las estructura locales de similitud entre los datos, así como asignar clusters que aproximen alguna función de densidad de la probabilidad predeterminada o alguna estructura global conocida. Muchas de estas técnicas comienzan suponiendo la existencia de K clusterings y lo que hacen es encontrar la partición optima con este número de elementos. Algunas de estas técnicas también suponen valores para los centroides de los clusters.

En el caso de los métodos particionales, el principal problema es que la elección del número de clusters puede ser crítica ya que diferencias sensibles entre clusters pueden surgir cuando K es cambiado. Además, una buena iniciación para los centroides de cluster puede ser crucial; algunos cluster inclusive podrían quedar vacíos si sus centroides quedan inicialmente lejos de la distribución de los datos.

En general, el principal problema con los métodos de clustering es la interpretación de los clusters. La mayoría de los algoritmos de clustering prefieren cierto tipo de cluster y los algoritmos siempre asignarán los datos a estos tipos de cluster aún cuando los datos dentro del cluster no tengan la similitud esperada. Por esta razón, si el objetivo no es únicamente comprimir el conjunto de datos sino también hacer inferencias acerca de la estructura de cluster, es esencial analizar cuando el conjunto de datos exhibe la tendencia inducida por la partición. En este sentido, los sistemas de visualización de información, que permiten explorar las estructuras inducidas por distintos métodos de clustering, son de gran utilidad.

2.3.4. Vector de Cuantización

Los métodos de **vector de cuantización** están basados en métodos clásicos de aproximación a la distribución de señales cuantizadas a un conjunto finito de **vectores de referencia**. La principal idea de este método cosiste en comprimir la información al representar a todos los elementos de $\mathbf{U}$ por un conjunto finito $\mathcal{N}$ de **vectores codificadores**. Para realizar esta representación se busca encontrar un conjunto óptimo de vectores $\mathcal{N} = \{\eta_1, \eta_2, ..., \eta_K\} \subseteq \mathbf{U}$, el cual se denomina **conjunto de codificadores**.

La representación de un elemento del conjunto universo se determina de la siguiente manera: dado $x \in \mathbf{U}$ el vector $\eta_{c(x)}$ es tal que:

$$d(x, \eta_{c(x)}) = \min \{d(x - \eta_i) \mid i = 1, ..., k\}, \tag{3}$$

en caso de existir más de un vector en $\mathcal{N}$ que cumpla con la condición 3 se elige uno de estos aleatoriamente de tal manera para que todo $x \in X$ exista un único

$\eta_c \in \mathcal{N}$ tal que η_c es la **representación** de x.

En los problemas de vector de cuantización establece a una función de manera que cada elemento del conjunto X es asociado a un elemento en $\mathcal{N}$; es decir, cada elemento de $x \in X$ queda representado por el vector codificador $\eta_{c(x)} \in \mathcal{N}$. Esta asociación se denotará de la siguiente manera:

$$x \sim \eta \Longleftrightarrow \eta = \eta_{c(x)}$$

Dado un elemento $\eta \in \mathcal{N}$ este define un subconjunto de X el cual consta de todos aquellos elementos del conjunto de datos que tienen como vector codificador al elemento η. Del mismo modo se puede extender la definición para que se concideren a todos los elementos del universo de manera que cada $\eta \in \mathcal{N}$ determina una región de $\mathbf{U}$. Estas nociones se formalizan en el siguiente concepto:

Definición 19 *Dado η un vector codificador, el **conjunto de Voronoi** asociado a η es:*

$$V_\eta = \{x \in X \mid x \sim \eta\},$$

en caso de hacer referencia al vector η_i, se utilizará la notación V_i en lugar de V_{η_i}.

Definición 20 *Dado η un vector codificador, el **región de Voronoi** determinada por η es:*

$$\hat{V}_\eta = \{x \in \mathbf{U} \mid x \sim \eta\},$$

En el caso de que $\mathbf{U} \subseteq \mathbf{R}^n$ normalmente se utiliza la métrica Euclidiana como función de distancia, las **fronteras** entre las regiones $\hat{V}_i$ y $\hat{V}_j$, están determinadas por $\hat{V}_i \cap \hat{V}_j$ y corresponden a subconjuntos de traslaciones de subespacios lineales. En la figura 3 se puede apreciar un conjunto de datos en $\mathbf{R}^2$ (puntos negros pequeños), que están asociados a un conjunto de codificadores (puntos grises grandes); en este caso las fronteras entre las regiones corresponden a segmentos de líneas.

Dado un conjunto de datos $X \subseteq \mathbf{U}$ y un conjunto $\mathcal{N}$ de vectores codificadores para este conjunto, se pueden definir medidas para determinar qué tan bien se representa el conjunto de datos por los vectores codificadores; una de estas medidas es la conocida como **error de cuantización** que se expresa de la siguiente manera:

$$E(X, \mathcal{N}^*) = \sum_{\eta \in \mathcal{N}^*} \sum_{x \in V_\eta} \frac{d(x, \eta_i)}{\#(V_\eta)} \tag{4}$$

donde $\mathcal{N}^* = \{\eta \in \mathcal{N} \mid V_\eta \neq \emptyset\}$.

El principal problema a resolver en estos métodos es determinar valores apropiados para los vectores del conjunto $\mathcal{N}$. En general, el problema puede ser planteado como un problema de optimización, una alternativa es determinar el valores óptimos de los vectores utilizando como función objetivo a minimizar el **error de cuantización** 4, con la restricción $\mathcal{N}^* = \mathcal{N}$. Como se puede apreciar, este es un problema de optimización que involucra un total de nK variables;

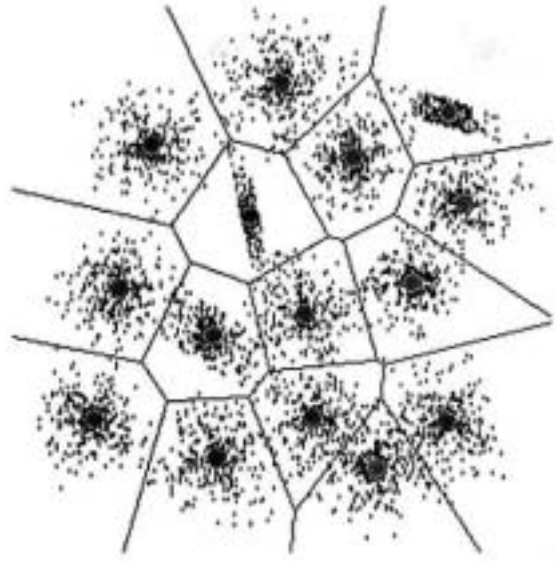

Figura 3: Vector de Cuantización

ya que cada entrada de cada vector codificador es una variable. Cabe señalar que no siempre está claro cuanto debe valer K; es decir, no siempre se sabe en cuantas regiones debe segmentarse el espacio. Estas circunstancias hacen que sea difícil establecer un método general y factible para resolver el problema de de *vector de cuantización*. En la actualidad este problema es abordado mediante el uso distintas herramientas, entre las que sobresalen las *redes neuronales*.

En general, las técnicas presentadas en esta sección tienen la gran limitante de poseer una alta *complejidad computacional [64]*, esta cualidad hace que aplicación de los métodos en grandes conjuntos de datos resulte poco factible, principalmente por el tiempo que le tomaría a una computadora llevar a cabo el procesamiento de los datos. Esta es una de las principales razones por la que el uso de modelos computacionales, con mayores capacidades de procesamiento, resulten una alternativa interesante para el procesamiento de grandes conjuntos de datos. En la siguiente sección se expondrá un ejemplo de una familia de estos modelos computacionales.

2.4. Redes Neuronales Artificiales

El ser humano lleva mucho tiempo preguntándose acerca del origen de la mente o del funcionamiento del cerebro. Desde la época de Aristóteles, el cual sentó las bases de la lógica que hoy en día sigue vigente; al dia de hoy, el tema de las funciones mentales y su relación con el funcionamiento fisiológico del cerebro es un campo de investigación fascinante; este gran proyecto se puede equiparar

a los proyectos de exploración del universo.

Por un lado se encuentra el estudio de los aspectos fisiológicos. Desde este enfoque se analiza el funcionamiento del cerebro como un sistema biológico con una rica dinámica. En la cual se involucran desde sus células constitutivas: las neuronas (ver figura 4). Estas células se organizan en redes intercambio de información, que permite que el cerebro esté conectado con los órganos de los sentidos y cada una de las partes del cuerpo humano.

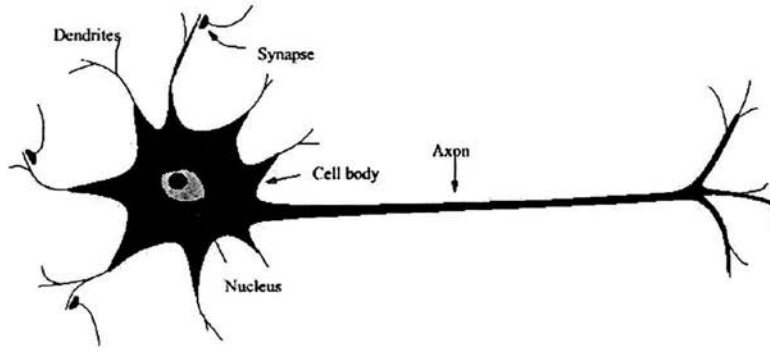

Figura 4: Esquema de neurona biológica

Por otro lado, los avances en el campo de la lógica matemática, han permitido establecer estructuras abstractas que permiten entender la forma de los razonamientos y de esta manera construir teorías y generar conocimiento.

A pesar de que aún no está clara la relación mente-cerebro. Lo cierto es que el cerebro tiene capacidades sorprendentes desde el punto de vista computacional. El número de operaciones y el tipo de tareas que el cerebro humano puede resolver, rebasan por mucho las capacidades y habilidades de cualquier máquina existente.

Las *redes neuronales artificiales* son modelos computacionales, los cuales tienen al cerebro humano como modelo ideal. Es decir, parten de un conjunto de procesadores denominados neuronas *artificiales* los cuales están interconectados de acuerdo a una arquitectura previamente definida. De manera tal que la información fluye en paralelo y distribuyéndose por varias neuronas; estas características permiten que las *redes neuronales artificiales* tengan un gran poder computacional, ya que son capaces de procesar grandes cantidades de datos y analizarlos simultáneamente desde distintas perspectivas. A continuación se intenta resumir la historia del desarrollo de este paradigma del *aprendizaje de máquina.*

2.4.1. Perspectiva Histórica

Por un corto periodo de tiempo, las *computadoras analógicas* compitieron con las *computadoras digitales*. Las computadoras analógicas podían ser usadas para modelar fenómenos naturales. Sin embargo, no podían ser usadas para realizar operaciones matemáticas exactas útiles en entre otras cosas para la contabilidad de negocios y manejo de inventarios. Aquí las computadoras digitales probaron ser superiores. Cada vez fueron más los problemas que se situaron dentro de los terrenos de las *computadoras digitales*, hoy en día computadora lleva implícito digital.

Una historia similar ocurrió con el desarrollo de modelos de *aprendizaje de máquina*. A finales de los años 50's y principios de los 60's, había dos escuelas principales de pensamiento. Una de ellas, la conexionista, partía del objetivo de construir un modelo computacional basándose en la arquitectura y los atributos clave que eran conocidos acerca del cerebro humano. La otra escuela sintió que la inteligencia podía ser producida en las máquinas a partir de sus capacidades para la manipulación de símbolos [11]. Los dos enfoques estaban fuertemente acoplados a las posiciones filosóficas prevalecientes respecto a la naturaleza de la inteligencia; esto permitió que el debate se llevara a cabo en la arena de la computación y el desarrollo teórico.

El antecedente inmediato al desarrollo del aprendizaje de máquina data de 1937 con el trabajo de Allan Turing [93], en el que se desarrolla el modelo de la "maquina de turing"el cual es un dispositivo que puede leer instrucciones de una cinta y simular cualquier otra máquina de cómputo. De hecho el mismo Alan Turing en su artículo de 1950 [94] habla de la universalidad de las *computadoras digitales* y del *aprendizaje de máquina*. Sin embargo, este modelo no se basa en el funcionamiento del cerebro; es un modelo abstracto que tiene su fundamento en la lógica matemática.

El comienzo de la modelación neuronal se dio con el trabajo de McCulloch, Warren y Pitts en 1943, estos autores introdujeron en [70] el primer ejemplo de modelo computacional de una neurona (ver figura 5), estas neuronas funcionaban como compuertas lógicas; y se planteron el cálculo de proposiciones lógicas por medio de *redes neuronales*. En este clasico artículo [70] titulado ."Logical Calculus of the Ideas Inmmanent in Nervous Activity"se muestra el modelo de una neurona *artificial* que funciona como una compuerta lógica capaz de integrar distintas entradas produciendo una salida.

En este trabajo pionero, los autores usaron al cerebro humano como modelo computacional y formularon redes que eran capaces de computar funciones booleanas. Las repercusiones de este trabajo van más allá del campo conexionista. De las memorias de John Von Neuman de 1958: "siguiendo a W. Pitts y W.S. McCulloch, estamos ignorando los aspectos más complicados del funcionamiento de una neurona... es facil ver que estas funciones simplicadas de las neuronas pueden ser imitadas por retransmisiones telegráficas y tubos de vacio", en esta cita [74] se puede observar la fuerte influencia que el trabajo de McCulloch y Pitts tiene en lo que hoy conocemos como arquitectura de computadoras.

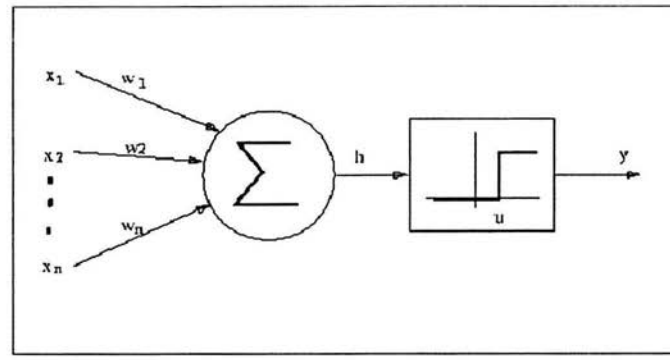

Figura 5: Modelo de Neurona de McCulloch y Pitts

El siguiente hecho importante en esta historia fue el trabajo de Rosenblatt con la introducción del perceptron en 1958 [81]. El perceptron era una máquina que tenía una malla de cuatrocientas foto celdas, parecida a una retina, las cuales eran conectadas aleatoriamente a 512 neuronas *artificiales*. Cuando un patrón era mostrado a las unidades sensoriales, los sensores activaban un conjunto de neuronas que determinaban la categoría del patrón. Esta máquina era capaz de reconocer todas las letras del alfabeto.

En 1960 Windrow y Hoff [95] presentan la aplicación del algoritmo de mínima media de cuadrados (LMS) para la corrección del error en su perceptron llamado Adaline (Adaptive Linear Element).

Una publicación importante de esta época fue la del libro de *aprendizaje de máquina* escrito por Nilsson en 1965 [75], en el cual se expone el caso de los patrones linealmente separables en hipersuperficies.

Como consecuencia de este debate en 1969 Minsky y sus colaboradores del MIT Marvin, Papert y Seymour publicaron su libro "Perceptrons: An introduction to Computational Geometry"[71], en el cual dieron un análisis matemático de las limitaciones de los perceptrones. Los autores fueron muy críticos con el movimiento conexionista, Minsky y Papert escribieron: "los perceptrones han sido ampliamente difundidos como reconocedores de patrones o máquinas que aprenden... y también se han discutido en un gran número de libros, artículos y voluminosos reportes. La mayoría de estos escritos...no tienen valor científico"[72]. La prueba más contundente era que los perceptrones resultaban ser fundamentalmente incapaces de resolver problemas tan simples como la operación lógica OR-Exclusiva. Las matemáticas de este libro eran irrefutables y por tal motivo una gran parte del movimiento conexionista se sumergió en una hibernación científica.

A pesar de que el ataque contra los perceptrones se dio dentro de los terrenos del debate científico, hubo otras motivaciones detrás de la curiosidad intelectual: agencias gubernamentales comenzaban a soltar fondos para la investigación en inteligencia *artificial* y los investigadores competían por estos fondos. En este

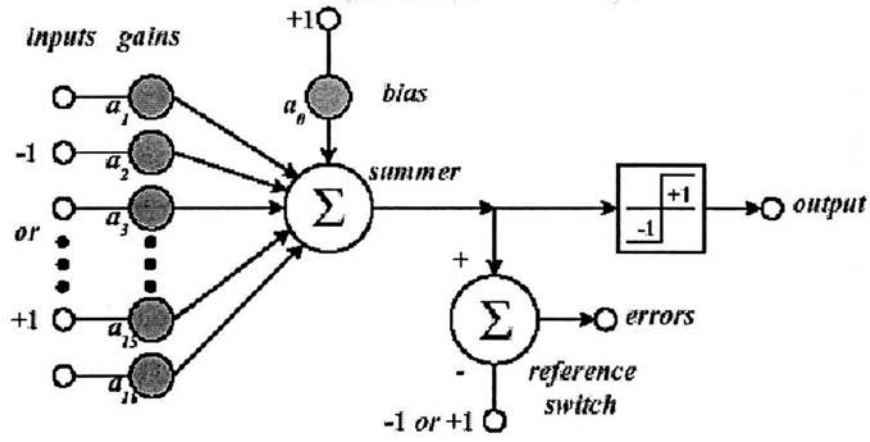

Figura 6: Modelo Adeline Básico

suceso histórico se puede observar una marcado divorcio entre dos paradigmas del *aprendizaje de máquina*: analítico y conexionista [2].

No obstante, la gran decepción ocasionada por el fracaso de los perceptrones, el trabajo en red neuronales no fue completamente abandonado. Durante los años 70's un número importante de investigadores continuaron en el desarrollo de modelos de *redes neuronales*. Mediante la exploración de distintas ramas las matemáticas, fue posible desarrollar modelos especializados para relizar diversas tareas. Dos temas importantes se gestaron durante esta época: memorias asociativas y redes auto-organizantes de aprendizaje competitivo.

En 1973, Grossberg publica: Çountour Enhancement, Short-Term Memory, and Constancies in Reverberating Neural Networks", en este artículo se expone la caracterización matemática de varias clases de redes competitivas. Esta línea de investigación mantuvo un desarrollo sostenido y tuvo un momento cúspide en 1984 con la publicación del libro "self-organization and associative memory"escrito por Kohonen [52], en este libro se introduce el algoritmo conocido como "self-organizing maps (SOM)"el cual hoy en día tiene una gran cantidad de aplicaciones en diversos campos de la ciencia e ingeniería; y será el tema central del siguiente capítulo.

Un trabajo de gran importancia que pasó casi desapercibido fue el de Paul Werbos en 1974, en su tesis doctoral "Beyond regresión: New tools for prediction and analysis in behavioral sciences", se introduce por primera vez el ahora popular modelo de Back Propagation; el cual puede ser visto como un perceptron con varias capas. Posteriormente este modelo es redescubierto de manera independiente por David Parker en 1982 .

A principios de la década de los 80´s un número importante de publicaciones apareció, estas cambiarían el curso de la historia de la investigación en *redes neuronales*. Una de las publicaciones más influyentes fue la de Hopfield en 1982. En el trabajo titulado: "Neural networks and physical systems with emergent properties"[43] se dieron las primeras razones formales para justificar su funcionamiento de las *redes neuronales* y por lo tanto dio legitimidad a su uso. En este trabajo se introduce la idea de función de energía de la física estadística para formular un nuevo camino en el entendimiento de la computación de redes recurrentes con conexiones sinápticas simétricas. Esta formulación hace explícito el principio de guardar información por medio de atractores dinámicamente estables (ver figura).

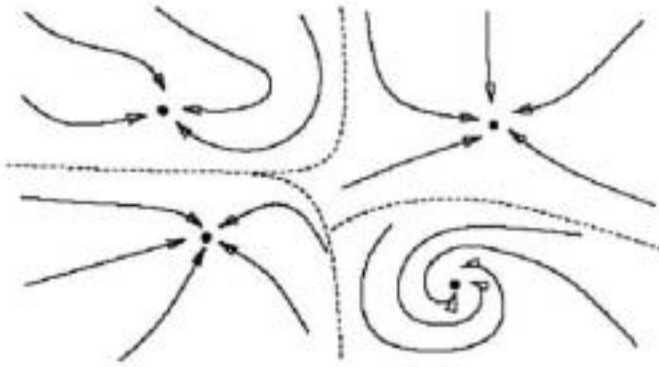

Figura 7: Esquema de los atractores, las cuencas de atracción y las orbitas.

Otro de los resultados que dieron validez a las redes neuronales es que problemas combinatorios de optimización, como el problema del vendedor viajero, fueron formulados y resueltos en términos de una función de energía de una red; la cual se minimiza cuando la red alcanza un estado estable.

De las características más importantes en la dinámica de una *red neuronal* son las llamadas propiedades emergentes. En palabras de Hopfield: "Ellas (las propiedades emergentes) nacen de la interacción de un muy grande número de elementos, entonces son la consecuencia de relaciones microscópicas que parecen tener vida propia". Sin embargo, ensamblar una colección de neuronas no es suficiente para producir pensamiento.

En 1986 McClelland, Rumelhart y el grupo de investigación "Parallel Distributed Processing-el paralelo es porque todas las neuronas actúan al mismo tiempo; el procesamiento porque el sistema no es solo un dispositivo de memoria, también usa información para a partir de un input particular generar un output; y distribuido porque ninguna neurona por si sola es responsable de alguna

función en particular, la actividad se distribuye entre muchas neuronas- publicaron este libro de dos volúmenes titulado "Parallel Distributed Processing: explorations in the microstructures of cognitions"[82]; en este libro se muestran programas básicos de *redes neuronales* y se da información concreta acerca de su modelación; es considerado por muchos como la Biblia del conexionismo.

McClelland encontró frustración en el viejo modelo de la mente: "el enfoque dominante era que la mente funcionaba como un dispositivo de cómputo secuencial de estados discretos", él dice: "trataba de usar esta visión para entender cómo el contexto influye a la percepción, pero no podía encontrar respuestas".

En el mismo libro, Rumelhart, Hilton y Williams reportaron el desarrollo del perceptron multicapas para resolver una gran variedad de problemas de reconocimiento de patrones. Rumelhart muestra que el modelo de back propagation se encuentra dentro de la categoría de perceptron multi-capas.

En la actualidad muchas de los modelos mencionados pueden ser simuladas en software o implementadas en hardware. Un número importante de paquetes compuaracionales, tanto públicos como comerciales, están disponibles; y cada vez más y más distintos investigadores han reconocido la importancia de las implementaciones en hardware, ya que esta es probablemente la única manera de aprovechar al máximo todas la capacidades de las RNA´s. Al mismo tiempo, miles de investigadores de diversos campos, tales como: neurociencias, psicología, medicina, matemáticas, física e ingeniería; han resuelto un número importante de problemas del mundo real utilizando *redes neuronales*. Estas actividades continúan creciendo como resultado del éxito obtenido en las aplicaciones y la inagotable producción de nuevos modelos [11].

2.4.2. Arquitecturas

Como ya se ha mencionado una *red neuronal artificial* RNA es un ensamble de neuronas *artificiales*. Por tal motivo, una RNA puede ser representada por medio de una gráfica dirigida; es decir, un conjunto de vértices unidos por flechas. El conjunto de vertices $\mathcal{N}$ es un conjunto de neuronas y al conjunto de conectividades entre las neuronas se le denomina: **arquitectura de la red** [46]. Basándose en el patrón de conectividad de la gráfica, las arquitecturas de las RNA pueden ser divididas en dos grandes categorías: las redes **"hacia delante"** (feedforward) en las cuales no existen ciclos; y las redes de **retroalimentación** (feedback) en las cuales si existen ciclos y por lo tanto la red se retroalimenta. Los ejemplos más típicos de las redes hacia-adelante son redes con varias capas en las que solo existe conexión entre capas consecutivas. En la figura 8 se muestran las arquitecturas de redes en cada categoría.

Generalmente, distintas conectividades resultan en comportamientos distintos al interior-exterior de las redes. En general, las *redes hacia-adelante* son **redes estáticas**; es decir, dado un input, estas producen un solo conjunto de valores de salida y no una secuencia de estos. Además, las redes *hacia-adelante* no tienen memoria ya que la respuesta de una de estas redes a un *dato de entrada* dado, es independiente de los *estados previos* de la red [80].

Las redes de *retroalimentación* son **sistemas dinámicos**, cada vez que se

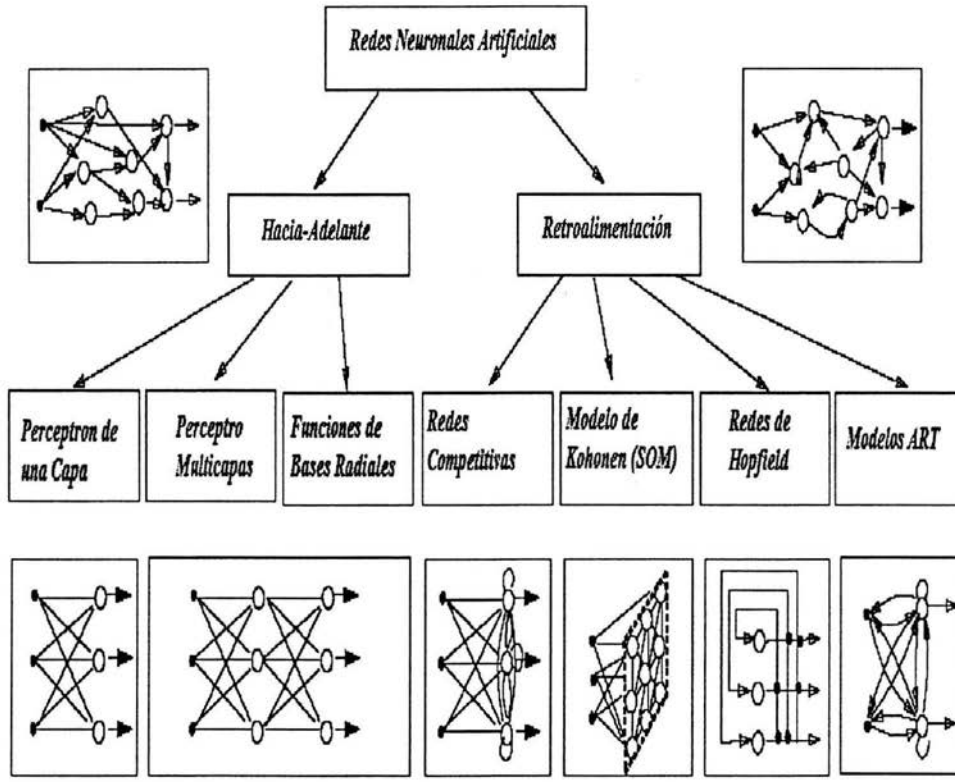

Figura 8: Arquitecturas de *Redes Neuronales.*

presenta un *dato de entrada*, las respuestas de las neuronas son computadas por medio de las conexiones de *retroalimentación*, de manera que los *vectores de pesos* de las neuronas son modificados. De esta manera, la red se encuentra en un *nuevo estado*. Este proceso se repite hasta que se alcance algún tipo de *equilibrio* o *convergencia [80]*.

Distintos *herramientas matemáticos* son empleadas para tratar con estos dos tipos de *redes neuronales*. Por ejemplo, en el caso de las redes de *retroalimentación*, los sistemas dinámicos son normalmente representados por ecuaciones diferenciales o por una función de dominio discreto en de manera que el sistema dinámico queda definido a partir de iteraciones de dicha función.

2.4.3. El Proceso de Aprendizaje

Para definir una red neuronal, no solo basta tener una arquitectura, también es necesaria la determinación de un proceso de aprendizaje. El **proceso de aprendizaje** en una *red neuronal* puede ser visto como un problema de *actualización* de los *pesos sinápticos* de las neuronas de manera tal que la *red neuronal* mejore la forma en que realiza cierta tarea específica. Estas mejoras en el desempeño de una *red neuronal* son logradas a lo largo del proceso de entranamiento y mediante la actualización iterativa del los pesos sinápticos, los cuales están representados por un vector denominado **vector de pesos**. En este proceso, se modifican los *vectores de pesos*, de manera que el comportamiento de la red neuronal se *adapte* y alcance un desempeño *óptimo*. Por esta razón, se dice que las *redes neuronales* son sistemas adaptables. En general, este aprendizaje de las se alcanza de dos maneras. Por lo tanto, *redes neuronales* son sistemas dinámicos auto adaptables, ya cambian para adaptarse a nuevas condiciones y realizar de mejor manera una tarea específica.

En ocasiones, las *redes neuronales* utilizarán información previamente establecida; y basandose en dicha información se lleva a cabo la actualización de los pesos. En otras ocasiones las *redes neuronales* deben aprender a establecer los pesos correctos a partir del conjunto de datos. Por lo tanto, existen tres tipos de aprendizaje: supervisado, no supervisado e híbrido.

En el aprendizaje supervisado, la *red neuronal* es provista de las respuestas correctas para cada elemento del conjunto de entrenamiento. Los pesos son determinados de manera que la red pueda producir respuestas tan cercanas como sea posible a las respuestas correctas.

En contraste el aprendizaje no supervisado no requiere de ninguna respuesta correcta; la *red neuronal* explora las estructuras subyacentes en los datos o las correlaciones entre patrones, y organiza los patrones en categorías basándose en estas correlaciones. Este trabajo está principalmente interesado en las *redes neuronales* de aprendizaje no supervisado, particularmente en el modelo de Kohonen el cual será expuesto con detalle en el siguiente capítulo.

Por último, el aprendizaje híbrido combina el aprendizaje supervisado y el no supervisado. Típicamente, una porción de los datos es actualizada utilizando aprendizaje supervisado, mientras que los pesos restantes son obtenidos de manera no supervisada.

La habilidad de las *redes neuronales* de automáticamente aprender de los ejemplos hace que las *redes neuronales* sean tan atractivas. En lugar de tener previamente especificado un conjunto de reglas, las RNA aprenden de una colección de ejemplos representativos.

Con el objetivo de entender o diseñar un proceso de entrenamiento, primero se debe tener un modelo del ambiente en el que las redes operarán; es decir, contar con un conjunto de entrenamiento. En segundo lugar, se debe entender la manera en que los pesos son actualizados; es decir, cuales son las reglas de aprendizaje que gobiernan el proceso de actualización.

2.4.4. Aplicación de las Redes Neuronales

El éxito de las *redes neuronales* reside más en su aplicación que en el desarrollo teórico. A continuación se expone brevemente los distintos problemas para los cuales las *redes neuronales* han sido exitosas (ver [11], [42], [46] y [80]).

- **Clasificación de Patrones**: esta es la tarea de asignar a un patrón de entrada representado por un vector de características, una clase predeterminada. Funciones discriminantes o fronteras de decisión son construidas a partir de un conjunto de patrones de entrenamiento con los niveles conocidos para separar patrones de distintas clases. Las aplicaciones más conocidas son reconocimiento de caracteres, reconocimiento de voz, clasificación de células sanguíneas, entre otras.

- **Clustering**: también conocido como clasificación no supervisada de patrones, ya que en el problema de *clustering* se supone que no existen clases conocidas para los datos de entrenamiento. Como ya se há mencionado, un algoritmo de clustering explora las similitudes entre los patrones y sitúa patrones similares en un mismo cluster; en el caso de las redes neuronales de entrenamiento no supervisado, el entrenamiento tendrá como el que los vectores de pesos representen a cierta subconjunto del conjunto de datos. En este sentido, las redes neuronales que se utilizan para hacer clustering pueden ser vistas como un método de vector de cuantización. El número de cluster no siempre es conocido a priori, por lo tanto, el problema de clustering es más difícil. Las aplicaciones más conocidas de estas redes neuronales son: minería de datos, compresión de datos y análisis exploratorio de datos.

- **Aproximación de una Función**: dado un conjunto de m patrones de entrenamiento etiquetados (pares input-output), $\{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$, generado a partir de una función desconocida $f(x)$, la tarea de la aproximación es la de encontrar una estimación, f^* de la función f. En la literatura estadística, este problema es comúnmente referido a la regresión. La estimación de una función f^* puede ser hecha fijando los datos de entrenamiento con una fidelidad arbitraria ajustando su complejidad. La clasificación de patrones también puede ser tratada como un problema

de aproximación. Existen una gran cantidad de problemas científicos y de ingeniería que requieren aproximar una función.

- **Predicción**: dado un conjunto de m muestras en una secuencia de tiempo, $\{y(t_1), y(t_2), ..., y(tn), t_1 < t_2 < ... < t_m\}$, la tarea es encontrar la muestra $y(t_{m+1})$ para algún tiempo futuro t_{n+1}. La predicción tiene un impacto muy significativo en la toma de decisiones dentro de los negocios, la ciencia y la ingeniería, así como en nuestra vida diaria. La predicción del stock del mercado y el pronóstico del tiempo son aplicaciones típicas de estas técnicas.

- **Optimización**: una amplia variedad de problemas en matemáticas, estadística, ingeniería, ciencias, medicina y economía, pueden ser tratados como un problema de optimización. Estos problemas usualmente contienen los siguientes componentes: (i) un conjunto de variables de estado o parámetros que se refieren a estados del proceso; (ii) una función objetivo o función de costo a optimizar; y (iii) un conjunto de restricciones para el dominio de la función objetivo. El objetivo de un algoritmo de optimización es encontrar el estado que satisfaga todas las restricciones al mismo tiempo que la función objetivo es maximizada o minimizada. Un problema de optimización combinatoria se refiere al problema en el cual todas las variables de estado son discretas y tienen un número finito de posibles valores. Un problema clásico de este tipo es el problema del vendedor viajero el cual es un problema NP-completo.

- **Memoria Asociativa**: en la arquitectura de computadora de Von Neuman, un registro en memoria es accesible solo a través de su dirección la cual no tiene un significado físico en términos del contenido en la memoria. Más aún, un pequeño error es hecho al calcular la dirección, un registro totalmente distinto será obtenido. Las memorias asociativas pueden ser accesadas a través de su contenido, este contenido puede ser llamado aún con un contenido distorsionado. La memoria asociativa es extremadamente deseable en la construcción de bases de datos con información multimedia.

- **Control**: Considera la dinámica de un sistema definido por $\{u(t), y(t)\}$, donde $u(t)$ es el input de control y $y(t)$ es el output resultante del sistema en el tiempo t. Para lograr esto se parte de un modelo de referencia de control adaptable, la meta de esta adaptación es la de generar un input $y(t)$ tal que el sistema siga la trayectoria deseada determinada por el modelo de referencia.

Un gran número de enfoques han sido propuestos para resolver estos problemas. Sin embargo, es común de que a pesar de que existen aplicaciones exitosas para problemas bien determinados, no existen soluciones suficientemente flexibles que permitan un buen desempeño fuera del dominio inicial de aplicación. El campo de las *redes neuronales artificiales* ha provisto de enfoques alternativos para la solución de estos problemas.

2.4.5. Ventajas de las Redes Neuronales

La principal razón del uso de las *redes neuronales* radica en el gran número de aplicaciones exitosos. El éxito en las aplicaciones se debe principalmente a las ventajas que las *redes neuronales* tienen sobre otro tipo de modelos computacionales. A continuación se mencionan algunas de estas ventajas.

Aprendizaje Adaptable

La capacidad de adaptable es una de las características más atractivas de las *redes neuronales*. Esto es, aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos. Como las *redes neuronales* pueden aprender a diferenciar patrones mediante ejemplos y entrenamiento, no es necesarios que elaboremos modelos a priori ni necesitamos especificar funciones de distribución de probabilidad.

Tolerancia a fallos

Las *redes neuronales* son los primeros métodos computacionales con la capacidad inherente de tolerancia a fallos. Comparados con los sistemas computacionales tradicionales, los cuales pierden su funcionalidad en cuanto sufren un pequeño error de memoria, en las *redes neuronales*, si se produce un fallo en un pequeño número de neuronas , aunque el comportamiento del sistema se ve influenciado, no sufre una caída repentina. Hay dos aspectos distintos respecto a la tolerancia a fallos: primero, las redes pueden aprender a reconocer los patrones con ruido, distorsionados o incompletos, esta es una tolerancia a fallos respecto a los datos. Segundo, pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red.

La razón por la que las *redes neuronales* son tolerantes a fallos es que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en este tipo de almacenamiento. La mayoría de los ordenadores algorítmicos y sistemas de recuperación de datos almacenan cada pieza de información en un espacio único, localizable y direccionable. Las *redes neuronales* almacenan información no localizada. Por tanto, la mayoría de las interconexiones entre los nodos de la red tendrán unos valores en función de los estímulos recibidos, y se generará un patrón de salida que represente la información almacenada.

Operación en tiempo real

Una de las mayores prioridades de la mayoría de las áreas de aplicación, es la necesidad de realizar grandes proceso con datos de forma muy rápida. Las *redes neuronales* se adaptan bien a esto debido a su implementación paralela. Para que la mayoría de las *redes neuronales* puedan operar en un entorno de tiempo real, la necesidad de cambio de los pesos de las conexiones o entrenamiento es mínima. Por tanto, las *redes neuronales* son un excelente alternativa para el reconocimiento y clasificación de patrones en tiempo real.

Fácil inserción dentro de la tecnología existente

Debido a que una red puede ser rápidamente entrenada, comprobada, verificada y trasladada a una implementación hardware de bajo costo, es fácil insertar *redes neuronales* para aplicaciones específicas dentro de sistemas existentes [11].

3. Redes de Kohonen

En este capítulo se presenta el modelo de *red neuronal* propuesta por T. Kohonen [53] denominado: Self-Organizing Maps (SOM). La presentación del modelo se realizará de la siguiente manera: en la introducción se presentan los aspectos "filosóficos"que envuelven al planteamiento matemático de la *red neuronal*; posteriormente se presentará el modelo matemático; luego se expondrán las principales razones del por qué este modelo representa una poderosa herramienta para el análisis inteligente de datos. Por último, se describirán algunas variantes del algoritmo básico y se mencionarán los principales aspectos teóricos que han sido abordados dentro del desarrollo de la investigación matemática.

3.1. Introducción

El SOM (Self-Organizing Map) es un algoritmo nuevo y eficiente para llevar a cabo la visualización de grandes conjuntos de datos multidimensionales. El algoritmo define una función del espacio de entrada a una *red de neuronas* en el plano. A su vez, ésta función define una proyección del conjunto de datos multidimensionales (invisible) a un espacio visible (normalmente bidimensional). La visualización del conjunto de datos permite que las relaciones de similitud que se presentan entre los datos dentro del espacio multidimensional puedan ser observadas en un despliegue bidimensional denominado "mapa".

Este algoritmo se ubica dentro del contexto de los algoritmos de entrenamiento para *redes neuronales* de aprendizaje no supervisado, lo cual implica que ninguna intervención humana es necesaria durante el proceso de entrenamiento; por lo tanto la visualización del conjunto de datos es generada de forma automática.

El SOM fue presentado en 1982 por **T. Kohonen** [50], desde entonces se han producido miles de artículos de investigación (una gran lista artículos se puede cosultar en [63]) y ha sido aplicado en una amplia variedad de campos de investigación [58].

La principal razón de la popularidad del SOM es su capacidad de presentar de manera automática un mapa en el cual se puede observar una descripción intuitiva de la similitud entre los datos; el despliegue bidimensional tiene la propiedad de presentar la información contenida en los datos de manera ordenada y resaltando las relaciones mencionadas. A continuación se exponen algunos conceptos generales relativos a la naturaleza y utilidad del algoritmo SOM.

3.1.1. Aprendizaje no Supervisado

Una problemática frecuente en el análisis de datos es que por un lado se cuenta con grandes cantidades de datos multidimensionales y por otro lado no se cuenta con información acerca de las relaciones y las estructuras subyacentes del conjunto de los datos; mucho menos se cuenta con una función de distribución o modelo matemático que describa estas estructuras; lo único con lo que se cuenta

es con un gran volumen de datos multidimensionales y con una forma de *medir la similitud* entre ellos.

Dado el gran volumen de datos y su alta dimensionalidad, resulta poco factible la aplicación de técnicas clásicas para la exploración de datos: como son el análisis por componentes principales o los métodos de *clustering* (ver sección 2.3).

En la sección 2.4 se mencionó que las *redes neuronales* poseen la capacidad de procesar grandes cantidades de datos multidimensionales; sin embargo, algunos de los modelos que se mencionan suponen que se cuenta con información que es utilizada para "corregir" el desempeño de la red por medio de un proceso de entrenamiento supervisado. Por lo tanto también resulta poco factible abordar esta problemática utilizando un *red neuronal* de este tipo.

Una alternativa para la solución a esta problemática es la utilización de *redes neuronales* de **aprendizaje no supervisado**. Estas *redes neuronales* son capaces de encontrar y descubrir, de manera automática, patrones de similitud dentro del conjunto de datos de entrenamiento [11]; y agrupar a los elementos de este conjunto en *clusters*, de manera que datos similares se agrupen dentro del mismo *cluster*. Estos descubrimientos pueden realizarse sin ningún tipo de retroalimentación con el medio externo y sin la utilización de información *a priori*.

Dentro del contexto de los procesos cognitivos del cerebro, la forma de situar al *aprendizaje no supervisado* es considerándolo semejante a los procesos inconscientes, en los cuales ciertas neuronas del cerebro aprenden a responder a un conjunto específico y recurrente de estímulos provenientes del medio externo, de esta manera se construyen los llamados "mapas sensoriales" en el cerebro.

Desde hace tiempo es sabido que varias áreas del cerebro, especialmente la corteza cerebral, están organizadas de acuerdo a distintas modalidades sensitivas: hay áreas que se especializan en algunas tareas específicas (ver Figura 9), ejemplos de estas tareas son: control del habla y análisis de señales sensoriales (visual, auditivo, somatosensorial, etc.) [59]. Distintas regiones de un mapa sensorial aprenden a reconocer estímulos específicos del medio ambiente. Como consecuencia, la información que cada cúmulo de neuronas reconoce se ubican dentro de cierta categoría dentro de los estímulos que se reciben del exterior.

La manifestación más clara del sentido fisiológico en el aprendizaje no supervisado de las *redes neuronales artificiales* es que "el aprendizaje puede suceder únicamente cuando hay redundancia en la presentación de los datos" [4]. En la práctica esta redundancia se obtiene mediante la utilización iterada (reciclaje) de un mismo conjunto de datos, a lo largo de todo el proceso de entrenamiento [60]. "Sin una retroalimentación con el exterior solo la redundancia puede proveer de información útil acerca de las propiedades del *espacio de entrada*" [80].

En resumen, una *red neuronal* no supervisada evoluciona durante su entrenamiento, de tal manera que cada unidad de salida será sensible a reconocer y organizar porciones específicas en el espacio de entrada.

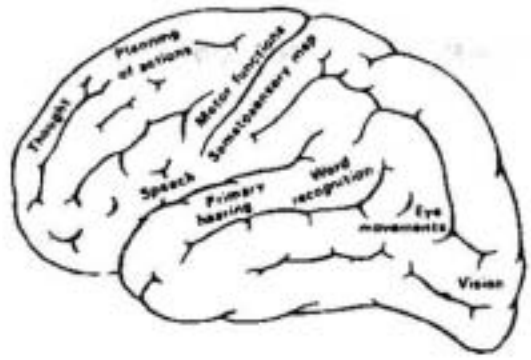

Figura 9: Areas Cerebrales

3.1.2. Entrenamiento Competitivo

Dentro del *aprendizaje no supervisado* existen dos filosofías principales: el **aprendizaje Hebiano** y el **entrenamiento competitivo**. Las redes neuronales correspondientes al primer caso están orientadas a medir la similitud o a proyectar al conjunto "input"en sus componentes principales, mientras que en el aprendizaje competitivo cada neurona de la red es entrenada para identificar y representar porciones específicas del espacio de entrada.

En las *redes neuronales artificiales* de *aprendizaje competitivo* las células reciben de manera idéntica la información de entrada sobre la cual compiten (ver Figura 10). Esta competencia consiste en determinar cual de las neuronas es la que mejor representa a un estímulo de entrada dado. Como resultado de esta competencia solo una neurona es activada en cada momento [61].

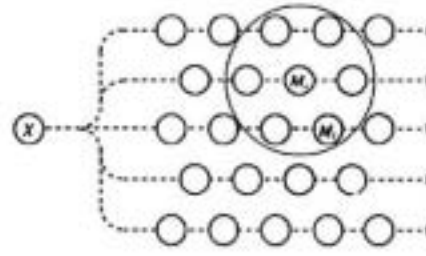

Figura 10: La neurona ganadora de un dato en X es M_c y la neurona M_i es una de las unidades vecinas

De esta manera, el proceso de entrenamiento competitivo de una *red neuronal*, determina un sistema dinámico discreto, en el cual cada iteración consiste

en la determinación de la neurona ganadora para cada elemento del conjunto de datos. "El proceso de entrenamiento competitivo de una *red neuronal* es estable, si después de un número finito de iteraciones, ningún patrón en el conjunto de aprendizaje cambia de representante"[80].

Una forma de lograr la estabilidad es forzando a un parámetro denominado **factor de aprendizaje** decrecer y eventualmente converjer a cero. De esta manera la red dejará de aprender y por lo tanto se mantendrá estable. Sin embargo, este congelamiento *artificial* del aprendizaje ocasiona que se pierda la plasticidad de la red, es decir la habilidad de adaptarse a nuevos datos. El dilema entre forzar la estabilidad y mantener la plasticidad durante el proceso de entrenamiento en una *red neuronal* es conocido como: "dilema de estabilidad-plasticidad"de Groosberg.

Una de las ventajas más significativas en las aplicaciones es que generalmente los modelos de *redes neuronales* basados en aprendizaje competitivo tienen arquitecturas muy simples y cuentan con algoritmos de entrenamiento más rápidos que otras *redes neuronales* como los perceptrones multi capas [80].

La más famosa de las *redes neuronales* no supervisadas competitivas es la propuesta por Kohonen (SOM). El resultado del *aprendizaje competitivo* en el caso del SOM es una partición del conjunto de datos de entrada inducida por la distribución de los datos en las neuronas. Ésta partición se realiza de manera que datos similares son agrupados por la red y representados por una sola neurona. Dicha neurona es la unidad ganadora para cada uno de los datos asociados durante la última iteración en el proceso de entrenamiento. Por lo tanto, la agrupación de los datos es realizada de manera automática, basándose en la similitud entre los datos y en la distribución de las respectivas neuronas ganadoras localizadas a lo largo y ancho de una retícula bidimensional.

3.1.3. Redes Neuronales y Auto-organización

La emergencia de comportamientos complejos en un sistema de elementos que interactúan es uno de los fenómenos más fascinantes observados en la naturaleza. Ejemplos pueden ser observados en casi cualquier campo de interés científico, desde la formación de patrones en los sistemas físicos y químicos, el movimiento de enjambres de animales en biología hasta el comportamiento de grupos sociales. Estas investigaciones conducen a la hipótesis de que existe una forma común por medio de la cual describir la formación de estas estructuras.

A pesar de que no existe una definición de auto-organización comúnmente aceptada, en este trabajo entenderemos que:

"La **auto-organización** es el proceso por medio del cual en un sistema de unidades individuales, por medio de interacciones cooperativas, emergen nuevas propiedades en el sistema que trascienden a las propiedades de sus partes constitutivas "[86].

El cerebro no escapa a la presencia de autoorganización, ésta puede observarse durante el establecimiento de las conexiones entre las neuronas en el cerebro y el sistema nervioso. Una idea fundamental que se deriva a partir de la presencia de *auto-organización* en el cerebro es que la información no es-

tá concentrada en una simple neurona, reside distribuida en distintas áreas, la memoria de un hecho corresponderá a la activación de una familia específica de neuronas [42]. Por lo anterior, algunos investigadores manejan la hipótesis de que el conocimiento es representado por el cerebro a partir de la emergencia de organización en las conexiones neuronales.

En el caso de las *redes neuronales artificiales de entrenamiento competitivo*, la ausencia de información previa hace necesario contar con algún *mecanismo auto-organizante*. Este mecanismo debe estar basado en algún criterio de similitud para que así, la organización de los datos corresponda a grupos de datos semejantes entre sí. De esta manera la evolución de la *red neuronal*, durante el proceso de entrenamiento, estará dirigida a hacer *emerger* una representación de las relaciones derivadas a partir de la similitud entre los datos.

El mecanismo auto-organizante en estas redes consiste en que la neurona ganadora tiene el potencial de modificar el vector de referencia de las unidades vecinas, la magnitud de la modificación está en función de la distancia física entre la neurona ganadora y cada una de las neuronas vecinas. A partir de esta interacción (ganadora-vecinas) es posible que las neuronas cercanas a la unidad ganadora también aprendan del dato de entrada y modifiquen su vector de referencia con la finalidad de adecuarse al tipo específico de dato. De manera que si una neurona representa a un conjunto de datos sus vecinas representan datos similares; por lo tanto, las relaciones de similitud son representadas por medio de la cercanía entre las neuronas.

El mecanismo auto-organizante que se propone en el SOM consiste de una *red neuronal* que usa su capacidad de aprendizaje *adaptable* para representar la estructura geométrica (orden topológico) subyacente en el conjunto de datos de entrenamiento, la representación es posible gracias a la auto-organización topográfica de las neuronas de acuerdo a las relaciones de similitud entre los datos representadas por la cercanía entre las neuronas y los vectores de referencia correspondientes.

En este sentido, el SOM constituye un mecanismo que brinda la posibilidad de producir automáticamente una representación del conjunto de datos en una estructura bidimensional. De manera que en dicha representación se haga evidente la emergencia de propiedades que ayuden a entender el orden geométrico subyacente en el conjunto de datos.

3.1.4. Visualización de Información y Mapas del Conocimiento

Partiendo del marco conceptual y las nociones expuestas en las secciones anteriores se pueden deducir algunas consecuencias que corresponden a propiedades de los "mapas sensoriales" y en la forma en la información es organizada. Entre otras cosas se puede concluir que [42]:

- Es posible representar la organización de la información a través de las relaciones entre las neuronas de una red.
- Como se dispone de un número finito de neuronas, la representación de la información debe corresponder al orden natural –estructura subyacente

en el conjunto de datos– y hacerce de manera que se utilice eficientemente el número de neuronas.

- La pérdida de neuronas no implica que se pierda la representación de la información.

Dadas estas propiedades es pertinente plantearse la posibilidad de construir diagramas visuales que representen la organización natural de la información. "Los **mapas de conocimiento** son representaciones gráficas de las conexiones hechas por el cerebro en el proceso de entendimiento de los hechos"[38]. Dichos mapas constituyen un medio visual en el cual ideas complejas puedan ser expuestas de manera rápida y en un orden lógico. La representación proporcionada por los mapas resulta de gran utilidad en el descubrimiento de características presentes en el conjunto de datos, de las que no se tenía conocimiento previo [38].

Por otro lado, en diversos campos de aplicación e investigación, la visualización adecuada de la información es un medio para resolver problemas de toma de decisiones o para la confirmación de alguna hipótesis [76]. Por tal motivo, la exploración visual de información en espacios complejos es uno de los temas de mayor interés dentro de la investigación actual de graficación por computadora. La *visualización de información* es un proceso asistido por herramientas computacionales en el cual se busca la representación visual de señales emitidas por un fenómeno abstracto o no visible [15]. La intención de la visualización de información es optimizar el uso de nuestra percepción y la habilidad de nuestro pensamiento para tratar con fenómenos que por si solos no pueden ser representados visualmente dentro del espacio [29].

En una gran cantidad de aplicaciones los mapas topográficos que se producen a partir del SOM resultan ser poderosas herramientas de análisis; el algoritmo SOM tiene la capacidad de producir medios visuales que representen las relaciones y estructuras de similitud entre los datos. En consecuencia, el despliegue visual de las relaciones de similitud provee al analista de una visión que es imposible obtener al leer tablas de resultados o simples *sumarios de estadísticas*.

Por lo tanto, los mapas generados a partir del SOM resultan ser útiles para el descubrimiento de información previamente desconocida y relevante en la comprensión del fenómeno correspondiente al conjunto de datos. En este sentido, "el SOM representa una herramienta que puede ser utilizada para la generación automática de mapas del conocimiento". Esta utilidad es aprovechada por los desarrolladores de sistemas computacionales para el *Descubrimiento de Conocimiento en Bases de Datos* (ver sección 4.2.4). En el capítulo 4 mencionarán algunos de estos sistemas y se explorarán las capacidades de un ejemplo en particular, el sistema Viscovery SOMine. A continuación se muestra un ejemplo en el cual es utilizado el SOM para la generación de mapas de conocimiento.

3.1.5. Ejemplo de la Aplicación de Mapas de Conocimiento

En el siguiente figura (11) se muestra un mapa producido por el SOM a partir de una tabla de datos en la que se listan valores de diversos indicadores

de varios países, registrados en 1992.

Para cada país se consideran 39 indicadores que describen varios factores del "nivel de vida de la población", tales como son indicadores de: salud, nutrición, educación, servicios, etc.

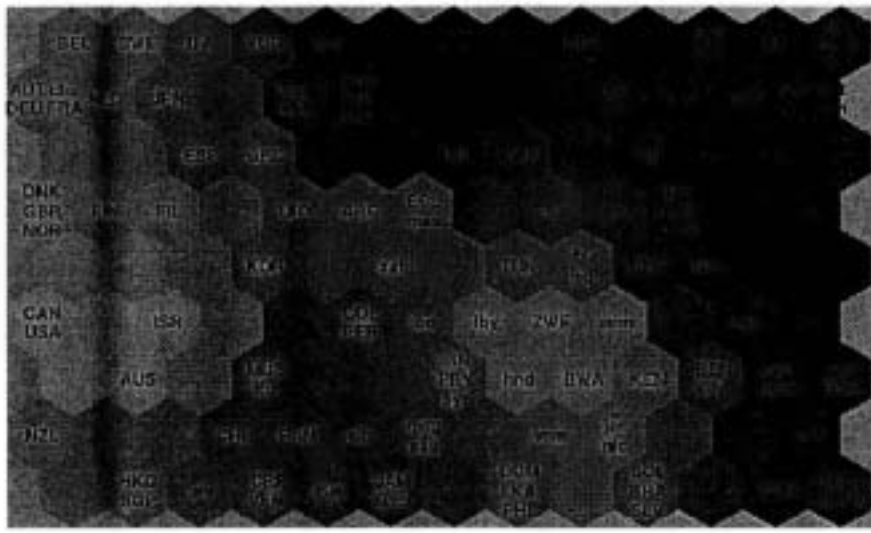

Figura 11: SOM en donde se representa la similitud entre los países de acuerdo a sus indicadores de nivel de vida.

Los distintos países se agrupan de acuerdo a la similitud que existe entre los datos. Los países que tienen indicadores similares, ocupan lugares cercanos en el mapa. Las clases de similitud o *clusters* son representadas por el mapa con distintas tonalidades de gris.

Este despliegue puede ser complementado por un sistema de información georeferenciada de tal forma que la tonalidad de este SOM puede ser usada en un planisferio (ver figura 12) como se puede observar la distribución geográfica de los países y la tonalidad tienen una alta correspondencia, lo cual no tendría necesariamente que suceder ya que para el entrenamiento del SOM tan solo se utilizaron indicadores de "nivel de vida". Estos mapas son una forma de representar conocimiento acerca de las relaciones entre los indicadores del nivel de vida y la distribución geográfica física de los países.

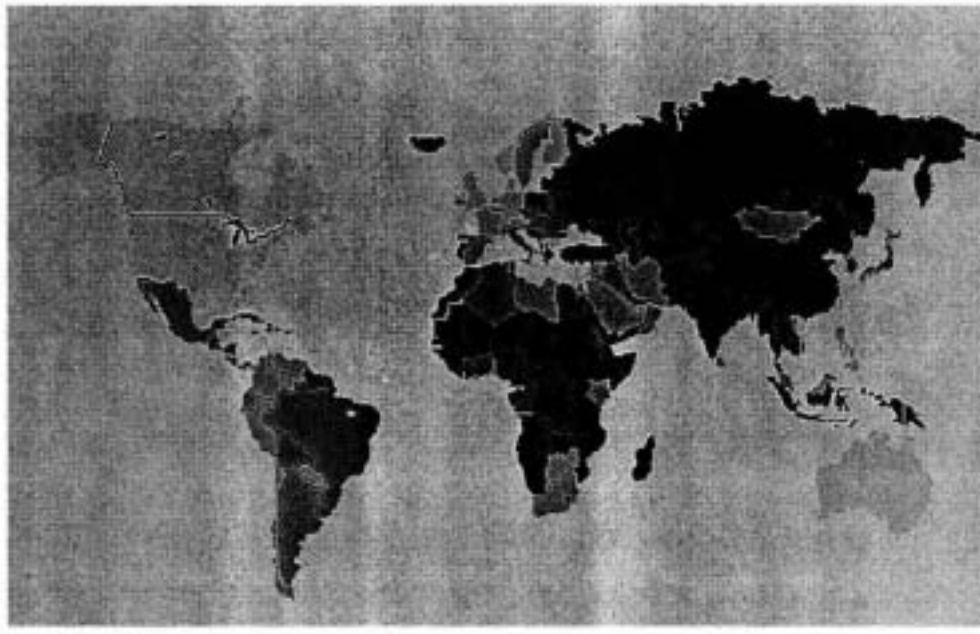

Figura 12: Planisferio en donde se representa la similitud entre los países de acuerdo a sus indicadores de nivel de vida.

3.2. El Algoritmo SOM

El SOM (Self-Organizing Maps) constituye un importante ejemplo dentro contexto del paradigma de las *redes neuronales*. En la sección 2.4 se estableció que para determinar una *red neuronal* era necesario definir: las neuronas, la arquitectura y el algoritmo de entrenamiento. Por esta razón, la presentación del algoritmo SOM se establece mediante la definición de estos aspectos.

3.2.1. Arquitectura

El punto de partida del SOM es un conjunto $\mathcal{N}=\{\eta_1,...,\eta_N\}$ de neuronas todas ellas con las mismas propiedades: se conectan de manera idéntica a la entrada $\bar{x} \in \mathbb{R}^n$ –normalmente se considera que $\mathbf{U} \subseteq \mathbb{R}$– e interactuan entre ellas por medio de relaciones laterales que se activan durante la actualización de los pesos. Estas relaciones responden a la relación (ver figura 13) de distancia física entre una neurona y sus vecinas.

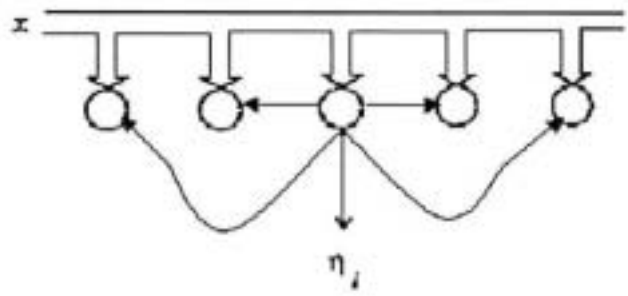

Figura 13: Representación de una neurona y sus conexiones con la entrada x y las neuronas vecinas.

Durante el proceso de entrenamiento competitivo, la entrada $\bar{x}$ se considera como una variable en función de t –donde t es la coordenada de tiempo discreto– que toma valores del *conjunto de datos de entrada* X, por tal motivo es necesario indexar a los elementos del conjunto X de la siguinete manera:

$$X = \{x(t) : t = 1, 2, ..., m\},$$

cuando el valor de t sobrepasa al número m, el conjunto X es reciclado y sus elementos son reindexados manteniendo el orden de la primera presentación.

Normalmente, la arquitectura de la red tiene las siguiente características:

- Las neuronas se distribuyen a lo largo de una retícula bidimensional.

- Cada neurona constituye a un nodo de la retícula.
- La configuración o tipo de retícula puede ser definida como rectangular, hexagonal o incluso irregular.
- La localización de la neurona sobre la retícula está representada por su **vector de localización** $r_i = (p_i, q_i) \in \mathbb{N}^2$.
- Cada neurona es asociada a un *vector de pesos* $w_i \in \mathbb{R}^n$. En el caso del SOM este vector es también llamado **vector de referencia**.

En la figura 14 se muestran las configuraciones o tipo de retícula más usados con los correspondientes $r_i = (p_i, q_i)$ en cada nodo.Cabe señalar que la

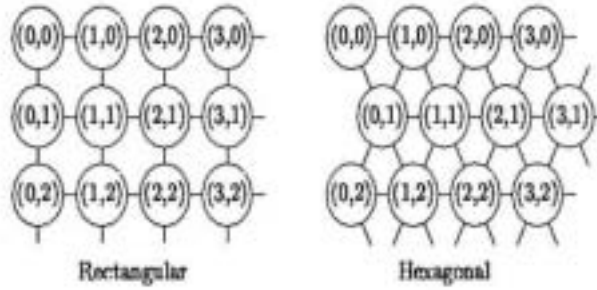

Figura 14: Configuraciones más comunes en la retícula del SOM.

configuración hexagonal es más conveniente para efectos de visualización.

En el algoritmo SOM básico, las relaciones topológicas entre los nodos (hexagonal o rectangular) y el número K de neuronas son fijados desde el principo. Normalmente se definen las distancias entre las unidades del mapa de acuerdo a la distancia *Euclidiana* entre los vectores de localización, sin embargo, en ocasiones es más práctico usar otras funciones de distancia.

3.2.2. Entrenamiento

El modelo de *red neuronal* del SOM pertenece al tipo de *redes neuronales* de aprendizaje no supervisado lo cual significa que ninguna intervención humana es necesaria durante el proceso de entrenamiento y que muy poco es necesario saber acerca de las características del conjunto de datos de entrada. El entrenamiento se lleva a cabo mediante un proceso de *aprendizaje competitivo* en el cual las neuronas se vuelven gradualmente sensibles a diferentes categorías de los datos de entrada.

En cada momento t del proceso de entranamiento, un vector de entrada $x(t) \in \mathbb{R}^n$ es conectado a todas las neuronas en paralelo vía los *vectores de*

referencia w_i de cada neurona. Las neuronas compiten para ver cual de ellas es capaz de representar de mejor manera al *dato de entrada* $x(t)$.

Dado cualquier $x \in X$ la competencia consiste en encontrar la neurona tal que su vector de referencia w_c cumpla con:

$$\|x - w_c\| = \min_{i=1}^{N} \{\|x - w_i\|\} \tag{5}$$

a la neurona ganadora η_c se le define como el **nodo que mejor representa** al dato x. Nótese que el subíndice c es función de x; para cada x existe un $\eta_{c(x)}$. En caso de que este indice no esté bien definido, es decir cuando para un dato x existan dos $\eta_e, \eta_d \in \mathcal{N}$ tal que:

$$d(x, \eta_e) = \min \{d(x - \eta_i) \mid i = 1, ..., k\} = d(x, \eta_d),$$

la selección de un único $c(x)$ debe hacerce de manera aleatoria. Por simplicidad se adoptará la siguiente notación:

$$x \sim \eta \iff \eta = \eta_{c(x)}.$$

Generalmente se utiliza la *distancia Euclidiana* para determinar el nodo que mejor representa a un dato en (5); sin embargo, se pueden usar otras *normas* (ver sección 2.1.2) si el problema lo requiere.

Para cada tiempo t se realiza la competencia (5) de manera que se puede definir $c = c(t)$ tal que $x(t) \sim \eta_{c(t)}$, aquellas neuronas que se encuentran dentro de una vecindad de $\eta_{c(t)}$ en el arreglo bidimensional (ver figura 15) aprenderán de la misma entrada $x(t)$.

La vecindad de $\eta_{c(t)}$ sobre la retícula se define a partir del vector de localización $r_{c(t)}$ de la siguiente manera:

$$N_c(t) = \left\{i \in \mathbb{N} \mid \left\|r_{c(t)} - r_i\right\| \leq \rho(t)\right\} \tag{6}$$

donde $\rho(t)$ es el radio de la vecindad en el tiempo t. Como se observa en 6, el radio de la vecindad varía en función de t. Para efectos de la convergencia del algoritmo, la variación del radio a través del tiempo debe cumplir las siguientes condiciones (ver figura 15):

1. Si $t_i \leq t_j \Longrightarrow \rho(t_i) \geq \rho(t_j)$
2. $\rho(t) \to 0$ cuando $t \to \infty$.

Debe tenerse cuidado al escoger el tamaño inicial de $\rho(0)$, si desde un comienzo la vecindad es muy pequeña, el mapa no se ordenará globalmente, lo cual implicará que el mapa generado se verá como un mosaico de parcelas entre las cuales el ordenamiento cambia discontinuamente. Para evitar este fenómeno $\rho(0)$ puede comenzar siendo más grande que la mitad del diámetro de la red.

Para iniciar el proceso de aprendizaje se utilizan valores aleatorios para los vectores de referencia $w_i(0)$. En las versiones más simples del SOM los valores

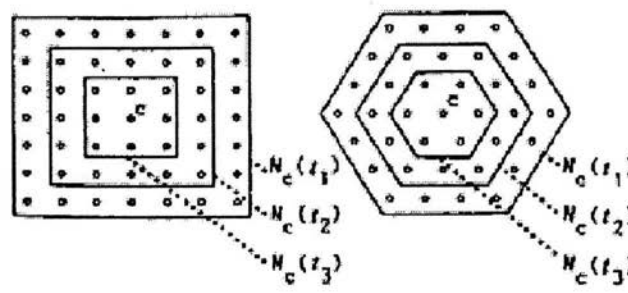

Figura 15: Variación en el tiempo del radio de la vecindad.

sucesivos para los *vectores de referencia* se determinan recursivamente por el siguiente mapeo de iteraciones:

$$w_i(t+1) = w_i(t) + h_{ci}(t)\,[x(t) - w_i(t)]$$

La función $h_{ci}(t)$ desempeña un papel fundamental en este proceso. A esta función se le conoce como **función vecindad**. En la literatura es común encontrar que esta función tenga la forma:

$$h_{ci}(t) = h(\left\|r_{c(t)} - r_i\right\|, t) \tag{7}$$

lo cual implica que el valor de la función depende de la distancia entre el neurona η_i y la neurona ganadora $\eta_{c(t)}$ en el tiempo t.

El ancho promedio $\rho(t)$ y forma de $h_{ci}(t)$ definen la *rigidez* del mapa que será asociada a los datos. Independientemente de cúal sea la forma explícita de la función 7, debe ser tal que $h_{ci}(t) \to 0$ mientras $\left\|r_{c(t)} - r_i\right\|$ se incrementa.

Una de las definiciones más simples que se encuentran de la función vecindad es la siguiente:

$$\begin{aligned} h_{ci}(t) &= \alpha(t) \quad \text{si } i \in N_c(t) \\ h_{ci}(t) &= 0 \quad\;\; \text{si } i \notin N_c(t) \end{aligned} \tag{8}$$

el valor de $\alpha(t)$ se define como **factor de aprendizaje** el cual cumple con la condición $0 < \alpha(t) < 1$ y usualmente $\alpha(t)$ es una función monótona decreciente.

Otra forma común de la función vecindad está dada en términos de la función Gausiana:

$$h_{ci}(t) = \alpha(t) \exp(\frac{\left\|r_c - r_i\right\|^2}{2\sigma^2(t)}) \tag{9}$$

donde $\alpha(t)$ es el *factor de aprendizaje* y el parámetro $\sigma(t)$ corresponde al ancho promedio de $N_c(t)$, en este caso $\rho(t) = \sigma(t)$. Tanto $\alpha(t)$ como $\sigma(t)$ son

funciones escalares decrecientes con respecto al tiempo. La definición de estas funciones debe tener como consecuencia del cumplimiento de básicamente dos etapas del proceso durante el proceso de entrenamiento: *ordenamiento global y refinamiento*.

3.2.3. Etapas del entrenamiento

- **Ordenamiento Global:** Segun lo reportado por T. Kohonen en [60], durante aproximadamente las primeras 1000 competencias se lleva acabo el ordenamiento de los datos a lo largo y ancho del mapa. Este ordenamiento consiste en establecer los pesos de cada neurona para que estas sean capaces de identificar cierto subconjunto característico dentro del conjunto de datos X y para que las relaciones de cercanía entre las distintas neuronas del mapa reflejen cercanía de los datos correspondientes en el espacio multidimensional del cual provienen. Si los valores iniciales de los pesos han sido seleccionados de manera aleatoria, durante estos primeros 1000 pasos los valores de $\alpha(t)$ deben comenzar siendo razonablemente grandes (cercanos a la unidad) e ir descendiendo hasta llegar a valores cercanos a 0,2. En general, la forma de $\alpha(t)$ no es importante, puede ser lineal, exponencial o inversamente proporcional a t. Es importante señalar que la selección óptima de estas funciones y sus parámetros solo pueden ser determinadas experimentalmente; ya que no existe algún resultado analítico que garantice dicha selección óptima.

- **Refinamiento:** Despues de la fase de ordenamiento los valores de $\alpha(t)$ deben ser pequeños y decrecer lineal o exponencialmente durante la fase fina. Dado que el aprendizaje es un proceso estocástico, la precisión final del mapa dependerá del número de pasos en esta etapa final de la convergencia, la cual debe ser razonablemente larga. El número de pasos debe ser del orden de 100000, sin embargo en ciertas aplicaciones, como el reconocimiento de voz, es de alrededor de 10000. Por otro lado, cabe señalar que la cardinalidad del conjunto X no es relevante para determinar este número de pasos. Nótece que el algoritmo es computacionalmente ligero y que el conjunto X puede ser reciclado para lograr tantos pasos como sea necesario [57].

Una vez concluido el proceso de entrenamiento, el SOM define una *regresión no-lineal* que proyecta un conjunto de datos de dimensión alta en un conjunto *vectores de referencia*, por lo que dicho conjunto sirve para obtener una representación del conjunto de datos en una *red adaptable* ("elástica") de dos dimensiones en la cual se pueden observar las relaciones de similitud y la distribución de los datos. De esta manera es posible construir una *representación bidimensional de un conjunto de datos multidimensional*. En las figuras 16, 17, 18 y 19, se observan las gráficas de los *vectores de referencia* durante el proceso de entrenamiento de un SOM con retícula rectangular, las adyacencias entre los *vectores de referencia* se refieren a las adyacencias entre neuronas vecinas en la red. El algoritmo implementado utiliza 9 como función vecindad. Las distintas

figuras corresponden a distintas distribuciones del conjunto de entrenamiento; en todas, la condición inicial de los vectores de referencia es aleatoria. Estas imagenes fueron realizadas por un programa que se está desarrollando en el *Laboratorio de Dinámica no-Lineal*, de la *Facultad de Ciencias* de la *U.N.A.M.*, en el cual el autor de esta tesis participa en el diseño e implementación de los algoritmos.

3.3. Ventajas en la exploración de datos

La virtud del algoritmo de aprendizaje del SOM es que forma una regresión no-lineal del conjunto ordenado de vectores de referencia dentro del espacio de entrada. Los vectores de referencia forman una red elástica de dos dimensiones que sigue a la distribución de los datos. A continuación se especifican las propiedades del SOM que lo hacen una herramienta útil y eficiente en el análisis de grandes conjuntos de datos multidimensionales.

3.3.1. Visualización del ordenamiento del conjunto de datos

El ordenamiento producido por la regresión permite el uso de los *mapas* como un despliegue de los datos. Cuando los datos son mapeados a aquellas unidades en el mapa que tienen los vectores de referencia más cercanos, las *neuronas vecinas* serán similares a los datos mapeados dentro de ellas. Este despliegue ordenado de los datos facilitará la comprensión de las estructuras subyacentes en el conjunto de datos.

El *mapa* puede ser usado como un campo de trabajo ordenado en el cual los datos originales pueden ser dispuestos en su orden natural. Estas disposiciones han sido discutidas en [48], las variables se aplanan localmente en el mapa, lo cual ayuda a penetrar en las distribuciones de los valores del conjunto de datos. Este mapa es mucho más ilustrativo que tablas de columnas con estadísticas linealmente organizadas. Estas características de los *mapas* generados por el SOM, permiten que el SOM sea útil para la generación de *mapas de conocimiento* los cuales son de gran utilidad en el proceso de *descubrimiento de conocimiento en bases de datos* (ver sección 4.2.2)

3.3.2. Visualización de *clusters*

El mapa resultante análisis del conjunto de datos puede ser usado para ilustrar la densidad de las acumulaciones en diferentes regiones en el espacio U en las cuales es posible observar relaciones de similitud. La densidad de los datos del conjunto de entrada X es representada por su acumulación en los *vectores de referencia*. En las áreas de acumulación los vectores de referencia serán cercanos y el espacio vacio entre ellos se hará cada vez más escaso. Por lo tanto, la estructura de *clusters* en el conjunto de datos puede vislumbrarse por la disposición de las distancias entre los vectores de referencia de las unidades vecinas. El diagrama de acumulación resultante es muy general en el sentido de que no se necesita asumir nada acerca del tipo clusters. Sin embargo, para lograr definir

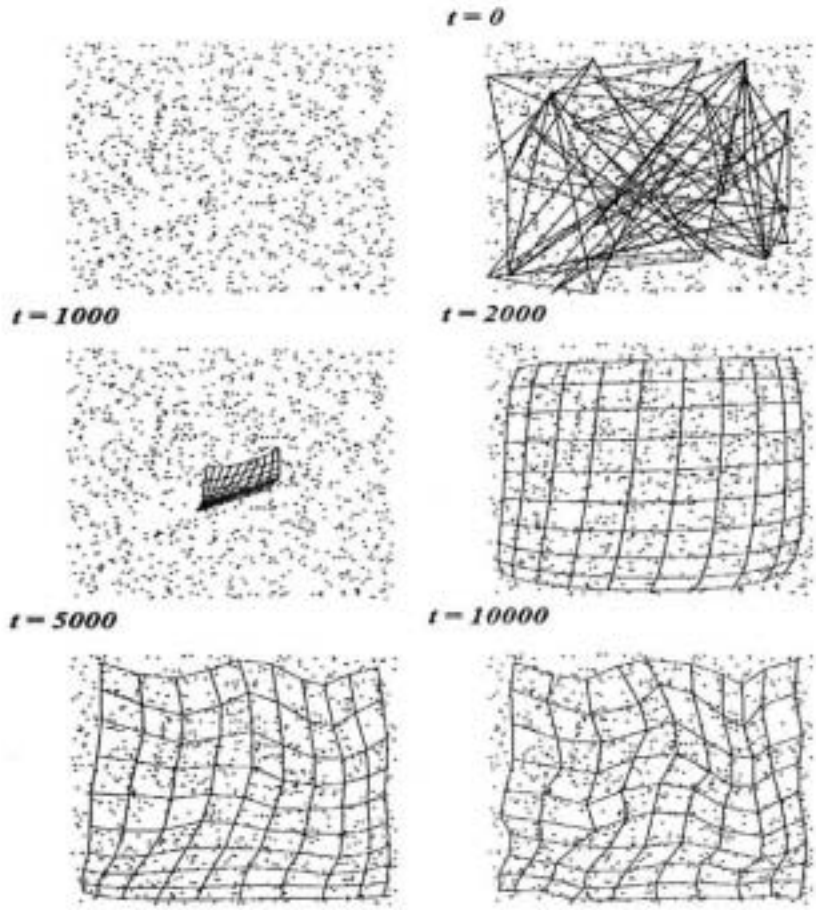

Figura 16: Evolución del entrenamiento de un SOM con 10×10 neuronas y un conjunto de 1000 puntos con distribución uniforme de sobre el plano.

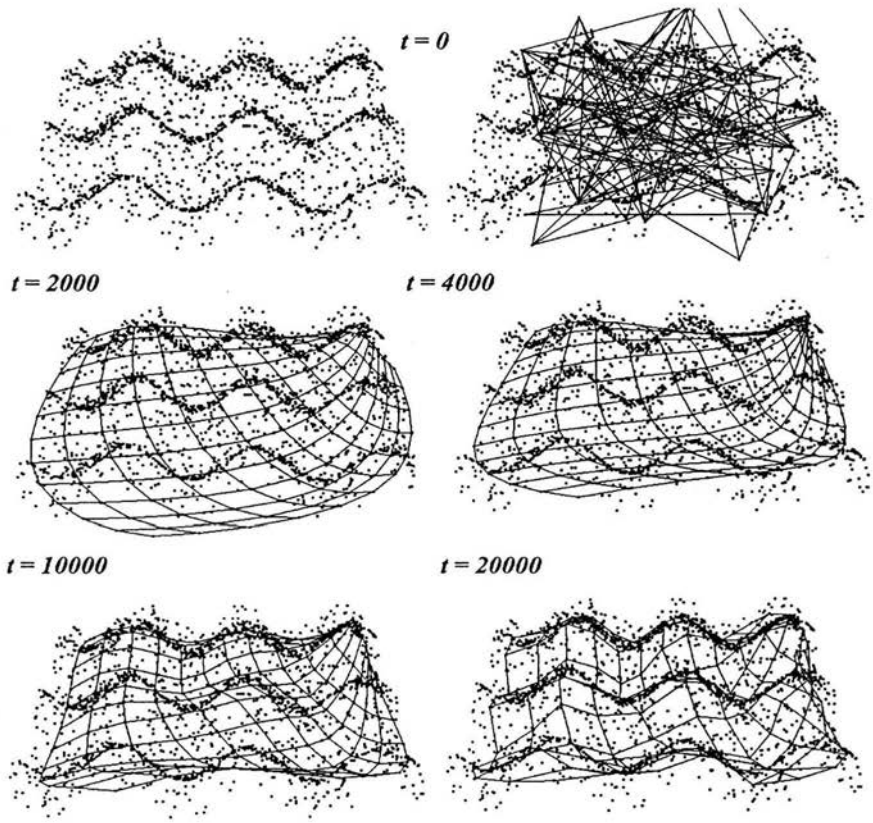

Figura 17: Evolución del entrenamiento de un SOM con 12×12 neuronas y un conjunto de 2000 puntos con distribución uniforme de sobre una superficie senoidal.

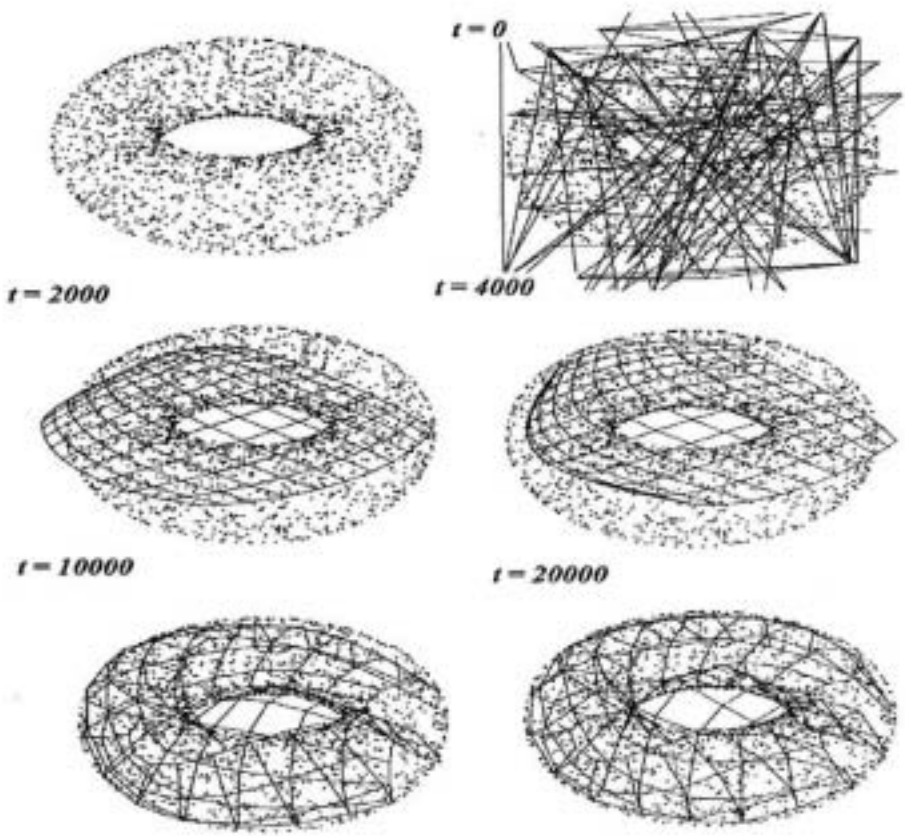

Figura 18: Evolución del entrenamiento de un SOM con 12×12 neuronas y un conjunto de 2000 puntos con distribución uniformesobre la superficie de un toro.

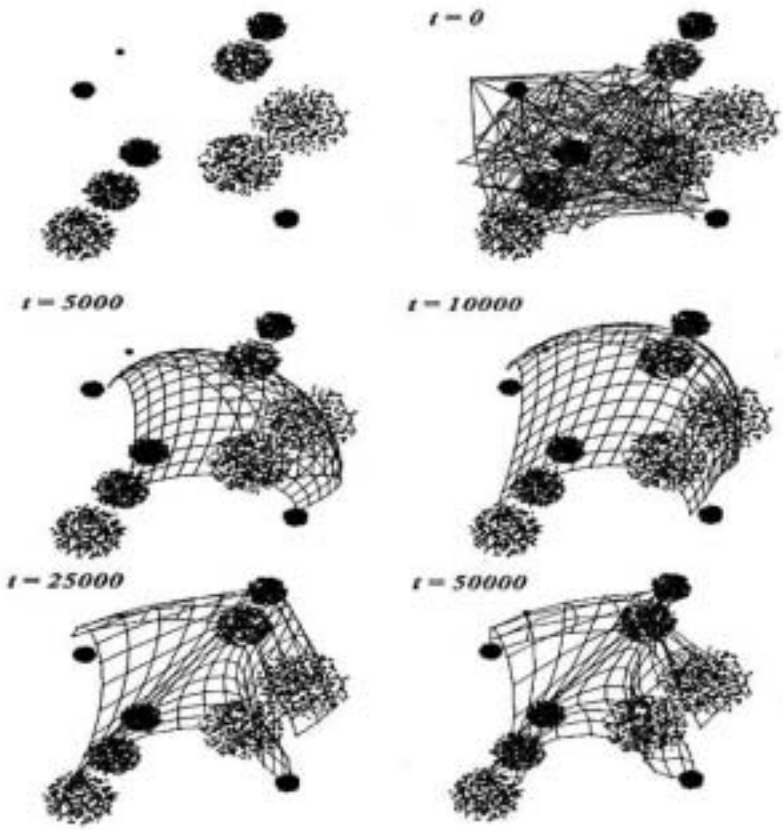

Figura 19: Evolución del entrenamiento de un SOM con 15×15 neuronas y un conjunto de 5000 puntos repartidos en el interior de 10 esferas.

los *clusters* es necesaria la aplicación de algún *algoritmo de clustering* sobre los vectores de referencia [33]. En la sección 4.4.2 se tratarán algunos ejemplos de la utilización del SOM como método clustering.

3.3.3. Datos faltantes

Un problema que ocurre frecuentemente al aplicar métodos estadísticos es el de los datos faltantes. Alguna de las componentes de los datos no está disponible o no es definible. Enfoques simples y complejos han sido propuestos para atacar este problema, el cual sufren parcialmente todos los métodos de clustering y de proyección.

En el caso del SOM el problema de datos faltantes puede ser tratado como sigue: cuando se escoge la unidad ganadora por (5), el vector de entrada x puede ser comparado con los vectores de referencia w_i usando solo aquellos componentes que están disponibles en x. Nótese que en los vectores de referencia no hay datos ausentes, de tal forma que si únicamente una pequeña porción de las componentes está ausente, el resultado de la comparación será estadísticamente completo. Cuando los vectores de referencia son adaptados solo las componentes que están disponibles en x serán modificadas.

En la artículo [85] se demuestra que se obtienen mejores resultados si se aplica el método antes descrito que si se opta por descartar los datos con componentes faltantes. Sin embargo, para datos en los cuales la mayoría de las componentes faltan, no es razonable asumir que la selección del ganador es adecuada. Un criterio razonables es el utilizado en [48] en donde los datos cuyas componentes ausentes exceden una porción determinada se descartan durante el proceso de aprendizaje. Sin embargo, las muestras descartadas pueden ser dispuestas en el mapa después de que ha sido organizado.

Nota: A pesar de que el SOM también puede ser usado para explorar conjuntos de datos incompletos, algunos métodos de preproceso pueden tener problemas con componentes ausentes en las referencias de los datos de entrada. Por ejemplo, la normalización de los vectores de datos no puede ser hecha de manera adecuada. La normalización de la varianza de cada componente por separado es una opción viable aún con datos incompletos.

3.3.4. Datos extremos

En la medición de los datos pueden existir datos extremos, que son datos ubicados muy lejos del cuerpo principal del conjunto de datos. Los datos extremos pueden resultar a partir de la medición de los errores o registrando los errores hechos mientras se insertan las estadísticas dentro de la base de datos. En estos casos es deseable que datos no afecten el resultado del análisis.

En el caso en el que el mapa producido por el algoritmo SOM: cada dato extremo afecta únicamente una unidad del mapa y su vecindad, mientras que el resto del mapa puede ser usado para inspeccionar el resto de los datos.

Más aún, los datos extremos pueden ser fácilmente detectados basándose en la distribución del *conjunto de entrada* X dentro del mapa. Si es deseado, los

datos extremos pueden ser descartados y el análisis puede continuar con el resto del conjunto de datos.

3.4. Variantes del SOM

Para crear una representación ordenada y organizada del conjunto de datos en una *red neuronal*, el principio esencial que utiliza el SOM es dejar que el aprendizaje de las neuronas dependa de la activación de las neuronas vecinas. Esta activación depende de la definición de la neurona que mejor representa a un dato de entrada, que a su vez está en función de la función de distancia que se esté utilizando. De manera tal que se pueden obtener variantes interesantes del algoritmo si en lugar de utilizar la métrica Euclidiana se utilizan otro tipo de funciones de distancia; sin embargo, estas variantes siguen siendo consideradas del SOM básico. Existen otros algoritmos basados en la filosofía del SOM que son más adecuados para resolver problemas particulares [57]. A continuación se exponen algunos de estos algoritmos.

3.4.1. La versión Batch Map

La versión del SOM presentada en la sección 3.2 es conocida como SOM **básico** o **secuencial** ya que en esta modalidad a cada dato corresponde una actualización. Por lo tanto los vectores de referencia $w_i(t)$ cambian después de la presentación de un dato $x(t)$, ya que la variable temporal t se incrementa con la presentación de cada dato $x(t)$. En la versión **Batch Map** del algoritmo SOM las actualizaciones se llevan a cabo después de que se han determinado las unidades ganadoras de todos los datos del conjunto $X = \{x_1, x_2, ..., x_m\}$. Por consiguiente, el incremento en la variable temporal t implicará la presentación de todo el conjunto de datos. A continuación se describe el algoritmo *Batch Map* considerando que se cuenta con una arquitectura igual a la utilizada en el algortimo SOM.

- **Primer Paso:** iniciar los N vectores de referencia $w_i(0)$ de manera "apropiada". De tal forma que para cada η_i exista $x \in X$ tal que $x \sim \eta_i$.
- **Segundo Paso:** para cada unidad η_i determinar el conjunto de Voroni en el tiempo t y calcular su centroide:

$$\begin{aligned} V_i(t) &= \{x \in X \mid x \sim \eta_i\}, \text{ donde } \eta_i \text{ tiene vector de referencia } w_i(t), \\ \widehat{x}_i(t) &= \frac{1}{(\#V_i(t))} \sum_{x \in V_i(t)} x. \end{aligned}$$

- **Tercer Paso:** los vectores de referencia son actualizados de la siguiente manera:

$$w_i(t+1) = \frac{\sum_{j=1}^{N} (\#V_j(t))(h_{j,i}(t+1))\widehat{x}_i(t)}{\sum_{j=1}^{N} (\#V_j(t))(h_{j,i}(t+1))}$$

- **Cuarto Paso:** repetir el proceso iterativamente a partir del *segundo paso*.

La forma de la función vecindad $h_{j,i}(t)$ se como en (7), de manera similar que en el SOM original. La diferencia principal diferencia es que en este algoritmo una iteración implica la presentación de todo el conjunto de datos.

Este algoritmo es especialmente efectivo si los valores iniciales de los vectores de referencia ya están ordenados de acuerdo a la estructura subyacente en el conjunto de datos, aún cuando no aproximen aún la distribución de la muestra. Cabe señalar que el presente algoritmo no considera un *factor de aprendizaje*, esto es porque no es necesario forzar la convergencia ya que los valores obtenidos para los vectores de referencia en cada iteración son mucho más estables [62].

3.4.2. Aprendizaje Supervisado

El algoritmo SOM se puede aplicar directamente a la visualización de un conjunto de datos. Sin embargo, en muchas aplicaciones se requiere que los algoritmos sean capaces de reconocer patrones previamente determinados, para estos casos es necesario contar con una *red neuronal* de aprendizaje supervisado. El SOM determina una aproximación de la *función de densidad de probabilidad* del conjunto de datos mediante la acumulación de los datos en los vectores de referencia. Sin embargo, las tareas de clasificación requieren aproximaciones de las *fronteras de decisión* optimas entre distintas clases.

Learning Vector Quantization (LVQ) es un grupo de algoritmos aplicable al reconocimiento estadístico de patrones, de acuerdo al cual las *clases* o *clusters* quedan representadas por un conjunto relativamente pequeño de **vectores codificadores** que corresponden a los *vectores de referencia* del algoritmo SOM. Como su nombre lo indica, el algoritmo de aprendizaje está orientado a obtener una *cuantización* del conjunto de datos [41]. Estos algoritmos se consideradan la versión supervisada del SOM ya que durante el entranamieto se presupone que ya se conocen las clases y la pertenencia tanto de los vectores de referencia como de los elementos del conjuntos X, a dichas clases. A continuación se exponen los algoritmos LVQ1, LVQ2 y LVQ3 [53].

A. El Algortitmo LVQ1

Asúmase que los valores de los *vectores codificadores* $w_i(0)$ son situados dentro del espacio de entrada de manera tal que aproxime distintas regiones del conjunto X. Generalmente varios vectores codificadores son asignados a cada clase y por lo tanto el vector de entrada $x(t)$ es clasificado en la misma clase a la cual pertenece $w_{c(t)}$, donde $\eta_{c(t)} \sim x(t)$. Los valores de $w_i(t)$ que minimizan los errores en la clasificación, cuando t se incrementa, pueden ser encontrados con la aplicación del siguiente proceso de aprendizaje:

$$
\begin{aligned}
\eta_{c(t)} &\sim x(t) \\
w_c(t+1) &= w_c(t) + \alpha(t)[x(t) - w_c(t)] \\
&\quad \text{si } x \text{ y } w_c \text{ pertenecen a la misma clase,} \\
w_c(t+1) &= w_c(t) - \alpha(t)[x(t) - w_c(t)] \\
&\quad \text{si } x \text{ y } w_c \text{ pertenecen a clases diferentes,} \\
w_i(t+1) &= w_i(t) \text{ para } i \neq c.
\end{aligned}
$$

donde $0 < \alpha(t) < 1$ y usualmente $\alpha(t)$ es una función decreciente. En la práctica se puede tener incluso que $\alpha(0) < 0{,}1$.

B. El Algortitmo LVQ2B.

La decisión de clasificación es idéntica a la de LVQ1. La diferencia es que en la actualización de los vectores de referencia correspondiente al elemento x se consideran dos elementos w_i y w_j. La propiedad que estos dos vectores deben cumplir es que uno de ellos debe pertenecer a la misma clase que el dato x y el otro a una clase distinta. Más aún, x debe pertenecer a una "ventana" la cual es definida alrededor del hiperplano intermedio entre w_i y w_j. Si d_i y d_j son las distancias Euclidianas desde x a w_i y w_j respectivamente, x está dentro de la ventana con ancho d si:

$$\min(\frac{d_i}{d_j}, \frac{d_j}{d_i}) < s, \text{ donde } s = \frac{1-d}{1+d}$$

una ventana con ancho $0{,}2 \leq d \leq 0{,}3$ es recomendada por Kohonen en [53].

El algoritmo de aprendizaje es definido de la siguiente manera:

$$
\begin{aligned}
w_i(t+1) &= w_i(t) - \alpha(t)[x(t) - w_i(t)] \\
w_j(t+1) &= w_j(t) + \alpha(t)[x(t) - w_j(t)]
\end{aligned}
$$

donde w_i y w_j son los dos vectores más cercanos a x, tal que w_j pertenece a la misma clase que x, mientras que w_i pertenece a una clase distinta. Además x cae dentro de la ventana con ancho d.

C. El algoritmo LVQ3

El algoritmo LVQ2 está basado en la idea de ir desplazando diferencialmente las fronteras de decisión basándose en las *fronteras de decisión Bayesianas* establecidos por la ventana. Sin embargo, no se toma en cuenta que pasa con la localización de los *vectores codificadores* a lo largo del proceso. Para asegurarse que los *vectores codificadores* continúen aproximándose a las distribuciones respectivas a sus clases se propone el siguiente algoritmo:

- Si se cumplen las condiciones de LVQ2:

$$
\begin{aligned}
w_i(t+1) &= w_i(t) - \alpha(t)[x(t) - w_i(t)] \\
w_j(t+1) &= w_j(t) + \alpha(t)[x(t) - w_j(t)]
\end{aligned}
$$

- Si x, w_i, w_j pertenecen a la misma clase, para $k \in \{i, j\}$:

$$w_k(t+1) = w_k(t) + \epsilon\alpha(t)[x(t) - w_k(t)]$$

Kohonen reporta en [53] que en una serie de experimentos se ha demostrado que valores de ϵ entre 0,1 y 0,5 son adecuados. Se ha observado que los valores óptimos para ϵ dependen del tamaño de la ventana; entre más angosta, más pequeño será el valor de ϵ.

3.4.3. Arreglos de SOM's

Con la finalidad de representar estructuras más complejas, conviene diseñar arquitecturas constituidas por la interconexión de varios SOM´s. Uno de los ejemplos más simples es el Multi-SOM desarrollado por el grupo encabezado por X. Polanco [76] en el *CNRS*. Este método comienza considerando el resultado del SOM básico original, como el nivel más bajo de una estructura piramidal de SOM´s, en cada nivel el número de nodos del SOM correspondiente se reduce, de manera que si en un nivel M el número de neuronas es $p \times q$, en el nivel $M+1$ el número de neuronas es $(p-1) \times (q-1)$ (ver figura 20).

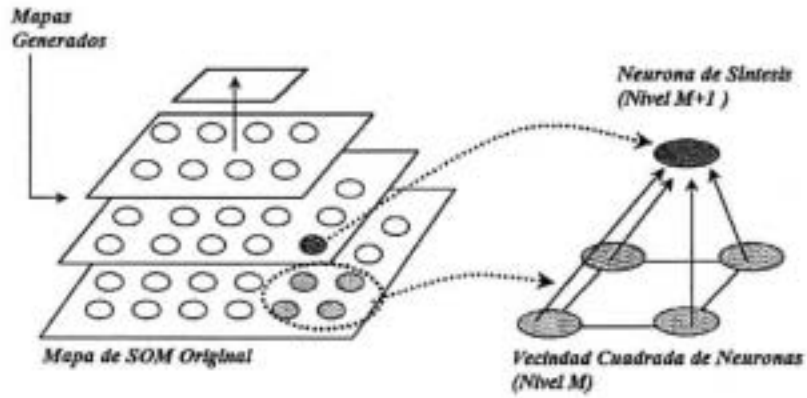

Figura 20: Esquema del modelo Multi-SOM.

La transición de un nivel a otro se hace asociando cada nodo del nuevo mapa con cuatro nodos del mapa en el nivel inferior (vecindad cuadrada de neuronas) de manera tal que el vector de referencia del nodo en el nivel superior es igual al promedio de los vectores de referencia de los nodos correspondientes en el nivel superior. Este procedimiento tiene la ventaja de preservar la estructura de vecindad del mapa original en los nuevos niveles generados y en cada nivel reducir el número de neuronas. De esta manera es posible ordenar el conjunto de datos dentro de una estructura jerárquicamente ordenada.

3.4.4. Retículas en Otros Espacios

El algoritmo SOM básico se construye con base en una retícula en un espacio bidimensional. Sin embargo es posible definir el mismo algoritmo de entrenamiento basándose en retículas definidas sobre otros espacios. La modificación principal en el algoritmo de entrenamiento para este tipo de modelos, tiene que ver con la definición de la función de distancia sobre la retícula.

El ejemplo más simple es el **SOM unidimensional**, en este modelo la retícula de neuronas consiste de un arreglo lineal de nodos (ver figura 21), al igual que en SOM básico cada nodo esta representado por un vector de referencia, que en este caso suele ser un vector de una dimensión.

Figura 21: Arreglo lineal de neuronas

Para el entrenamiento se usa la misma *regla de aprendizaje* que en el SOM básico. Definiendo la misma función vecindad y utilizando como norma el valor absoluto de las diferencias. Lo más relevante de esta variante del SOM es que permite definir la auto-organización en términos de alcanzar configuraciones ordenadas en los pesos (ver sección 3.5.1).

En lugar de reducir la dimensión del mapa, el **SOM multidimensional** se construye sobre una retícula dentro de un espacio con dimensión mayor a dos. La principal desventaja de esta variante es la imposibilidad de visualizarlo cuando la dimensión es mayor que tres.

En el la figura 22 se representan las neuronas de un SOM tridimencional. Los vectores de localización de dichas neuronas son de la forma:

$$(i, j, k) \in \mathbb{N} \text{ tal que } 1 \leq i \leq 4, 1 \leq j \leq 41 \leq k \leq 4;$$

por otro lado, en la figura se muestra la evolución de los vectores de referencia de las neuronas $\mathcal{N}$ durante el proceso de entrenamiento en el se utiliza un conjunto de 2000 puntos uniformemente distribuidos en el interior de un cubo. Cada imagen corresponde a el tiempo t que se señala.

Un caso especial de estas arquitecturas, es cuando sus nodos corresponden a puntos sobre un toro, este caso es útil cuando se quiere determinar si en el conjunto de datos visto como serie de tiempo presenta algún tipo de comportamiento periódico. Otro caso especial es cuando los nodos de la retícula son

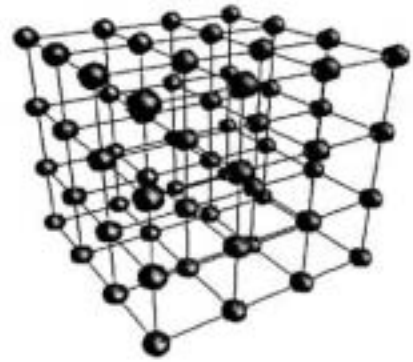

Figura 22: Retícula rectangular de un SOM tridimensional de $4 \times 4 \times 4$ neuronas.

vértices de un hipercubo $\{0,1\}^N$ de dimensión N. En esta situación la vecindad de cada nodo consistirá de él mismo y todos los nodos que compartan una arista con él.

3.5. Aspectos Teóricos

El algoritmo SOM tiene un una gran utilidad práctica. A pesar de que está originalmente concebido como una *red neuronal*, no es fácil ubicarlo dentro de alguna familia de algoritmos [20]. Como ya se há observado, una propiedad interesante en el SOM es el gran número de formas y variantes en su configuración, esta propiedad hace que el número y el tipo de aplicaciones sea muy extenso. Sin embargo, esta falta plasticidad hace que el análisis teórico sea muy difícil de realizar.

El mismo Kohonen opina que "uno podría pensar que la estructura del arreglo regular básico y su algoritmo son tan básicos simples que después de veinte años de investigación intensiva la teoría ya estaría establecida" [57].

M. Cottrell, una de las principales investigadoras de los aspectos téoricos del SOM opina al respecto: "El algoritmo SOM es muy raro. Por un lado, es muy simple de escribir y simular; sus propiedades prácticas son claras y fáciles de observar. Pero, por otro lado, sus propiedades teóricas permanecen sin pruebas en el caso general, a pesar de los grandes esfuerzos de varios autores" [20]... "El algoritmo de Kohonen es sorprendentemente resistente a un estudio matemático completo" [22].

A la fecha, los intentos por encontrar resultados teóricos generales para el algoritmo han fracasado y sólo se han podido encontrar algunos resultados parciales al restringir la forma o el dominio de alguno de sus operadores. Hasta donde se sabe, el único caso donde se han podido establecer pruebas completas es en el caso unidimensional, en el cual la entrada es de una sola dimensión y las neuronas son dispuestas a lo largo de un arreglo lineal.

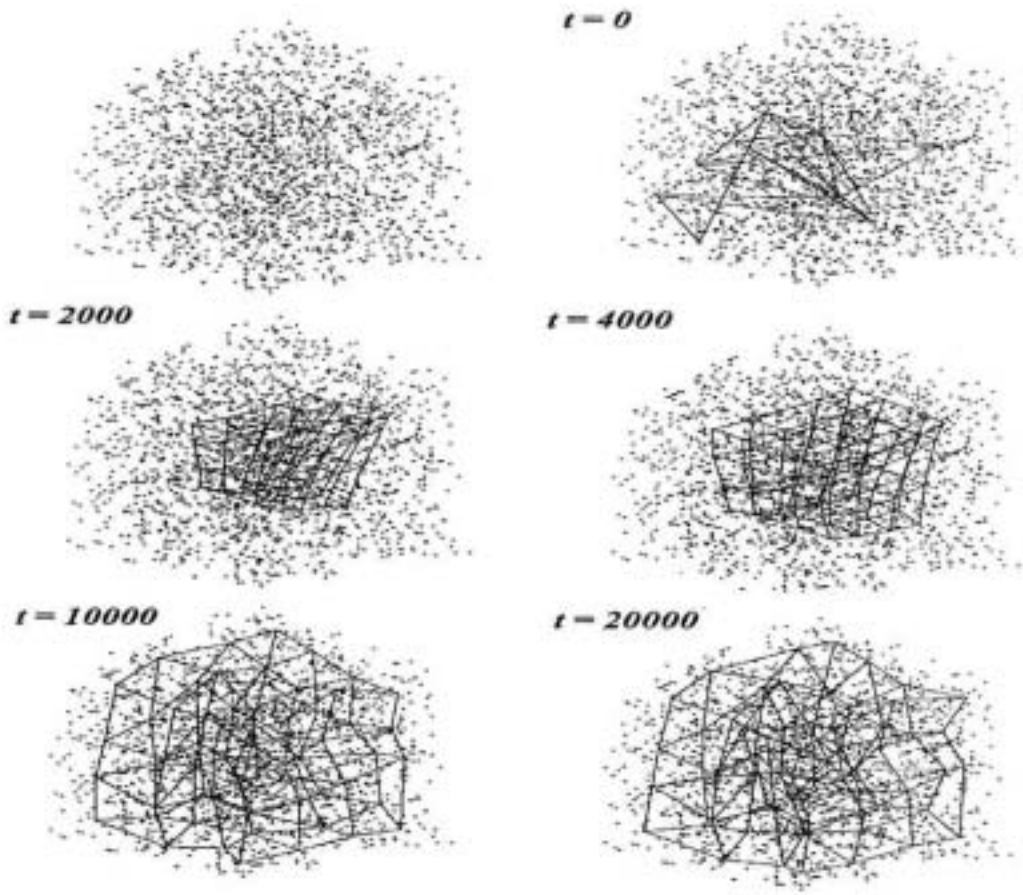

Figura 23: Evolución de los *vectores de referencia* de un SOM tridimensional.

Dentro de los aspectos teóricos que han llamado la atención a los investigadores, los más desarrollados son: convergencia y auto-organización, preservación de la topología y el enfoque de sistemas dinámicos. A continuación se comenta brevemente cada uno de estos aspectos.

3.5.1. Convergencia y Auto-organización

Los estudios de la convergencia del algoritmo de entrenamiento del SOM, parten de la noción de que es posible obtener una configuración ordenada de la retícula que corresponda a un orden en los vectores de pesos. La auto-organización se establece cuando durante el proceso de entrenamiento se alcanza dicha configuración ordenada. La idea central para conceptualizar lo que significa una *configuración ordenada* es partir de la definición de "orden topológico", pero ¿Qué es exactamente "orden topológico"?, esta pregunta tiene una respuesta trivial en el **caso unidimensional**, ya que una configuración ordenada de acuerdo al orden topológico corresponde al caso donde los vectores de referencia están ordenados linealmente.

Definición 21 *Considerese un SOM unidimensional con* $\mathcal{W}_t = \{w_i(t) \in \mathbb{R} \mid 0 \leq i \leq K\}$ *el conjunto de vectores de referencia en el tiempo de iteración* $t \in \mathbb{N}$. *Se dice que el SOM se* **auto-organiza** *si existe* $\tau \in \mathbb{N}$ *tal que:*

$$\begin{aligned} \mathcal{W}_\tau &= \{w_i(\tau) \in \mathbb{R} \,|w_1(\tau) \leq w_2(\tau) \leq ... \leq w_K(\tau)\} \text{ ó} \\ \mathcal{W}_\tau &= \{w_i(\tau) \in \mathbb{R} \,|w_K(\tau) \leq w_2(\tau) \leq ... \leq w_1(\tau)\} \end{aligned}$$

Por lo tanto, lo que uno esperaría es probar que el SOM siempre o casi siempre se auto-organiza. Un primer esbozo de prueba de la convergencia se establece en los artículos originales de Kohonen en 1982 [50], [51]; y más recientemente en su libro [57]. Sin embargo, la primera prueba completa para la convergencia y auto-organización, se establece en por Cottrell en 1987 [19], en este trabajo se considera una *distribución uniforme* de las entradas y una función de vecindad simple de salto, es decir que vale uno o cero. Bouton, Pagès y Fort en 1993 dan una prueba rigurosa para la *convergencia casi segura* a un estado único después de la auto-organización, bajo el supuesto que esta se alcanza [14] y considerando una clase muy general de *función vecindad* [35].

Posteriormente, Sadeghi en 1998 [84], [83] estudió la auto-organización para un tipo muy general de estímulo y función vecindad. Para probar la convergencia casi segura se considera al SOM como un algoritmo de tipo Robbins-Monroe y haciendo ver que el conjunto de configuraciones bien ordenadas es una clase absorbente del proceso estocástico asociado.

Recientemente sobresale el trabajo de Flanagan [30], [31], [32], en este último artículo del 2001 se presenta una prueba para la auto-organización en el caso de una dimensión para una *función vecindad* general y donde la *distribución de las entradas* puede ser *discontinua*. Cabe señalar que la herramienta analítica que se utiliza para establecer todos estos resultados es la de las *cadenas de Markov*. El caso unidimensional del SOM tiene muy poca importancia práctica.

Sin embargo, un profundo entendimiento de sus propiedades es un primer paso necesario para lograr entender las versiones de mayor dimensión.

En el caso de contar con una retícula en un espacio de dimensión mayor o igual a dos existe una dificultad natural para establecer el concepto de *configuración ordenada*. Después de una serie de fracasos actualmente se piensa que: "no es posible determinar que sería una configuración ordenada que fuera estable para el algoritmo y que pudiera ser una clase absorbente del proceso estocástico" [57].

3.5.2. Preservación Topológica

Otra línea de investigación es el establecimiento de medidas que determinen la eficiencia del SOM en lo que se refiere a preservación de la topología. Para esto se han expuesto varias propuestas para la cuantificación del error, la mayoría de éstas se basan en aplicar métodos lineales estadísticos, sin embargo, la mayoría de estos intentos fallan en el caso donde los datos se localicen sobre variedades no lineales. Un ejemplo de estos trabajos es el de Bauer [5] en donde se propone una medida para cuantificar la preservación de la vecidad. Otra propuesta interesante es la expuesta por Martinetz en [69] en la cual se propone una medida que toma en cuenta la no-linealidad de los datos, sin embargo, en esta propuesta se establecen cambios significativos en lo que se refiere a la determinación del nodo ganador y las funciones de distancia.

3.5.3. Sistemas Dinámicos

En la busqueda de otros enfóques analíticos. Una línea de investigación interesante es la que se deriva de la teoría general de *sistemas dinámicos* y que hace extensivo el uso de herramientas teóricas tales como las funciones de Lyapunov que son una generalización de las *funciones de energía* que aparecen en los sistemas mecánicos. Erwin, Obermayer y Shulten [25] prueban que el algoritmo original no puede ser derivado de funciones (potenciales) de energía. Por otro lado, Heskes [40] prueba que con un cambio en la definición de la unidad ganadora la regla de aprendizaje original puede ser vista como un *gradiente descendente* estocástico de una función de energía. Otro avance en esta dirección lo logra Tolat [91], estableciendo una función de energía para cada una de las neuronas.

Otra forma de investigar la auto-organización y la convergencia es estudiar *ecuación diferencial ordinaria* (ODE) asociada la cual describe el comportamiento promedio del algoritmo. "Sin embargo, actualmente el completo estudio asintótico de la ODE en el caso multidimensional parece ser intratable. Se tienen que verificar algunas suposiciones globales acerca de la función vecindad y los cálculos explícitos son muy difíciles y tal vez imposibles" [36].

4. Descubrimiento de Conocimiento con el SOM

En el capítulo 2 se presento el **análisis inteligente de datos** como un campo multidisciplinario en el cual convergen principalmente la estadística y el *aprendizaje de máquina*. Por un lado, se considera la aplicación de métodos estadísticos principalmente enfocados a la determinación de modelos; y por otro, la implementación de métodos computacionales capaces de evolucionar y mejorar su desempeño en la realización de tareas específicas. Posteriormente en el capítulo 3 se presentó el ejemplo del SOM como un método para el análisis inteligente de datos. Por medio del algoritmo SOM, una *red neuronal* aprende a representar visualmente un conjunto de datos y las relaciones entre sus variables, gracias a que la proyección preserva la topología en el conjunto de datos.

El desarrollo de sistemas de software que implementan este tipo de herramientas tiene distintos enfoques. Uno de los más interesantes es el de aplicaciones computacionales que tienen como objetivo el descubrimiento de conocimiento a partir del procesamiento masivo de información contenida en grandes bases de datos. Este objetivo se alcanza por medio de la concreción de un proceso conocido como **Descubrimiento Conocimiento en Bases de Datos** (KDD).

La denominada **Minería de Datos** (DM) es una etapa dentro del proceso KDD. En esta etapa se lleva a cabo la aplicación de distintas herramientas y métodos de análisis inteligente de datos, por medio de los cuales es posible obtener, de manera automática, información útil que una vez validada se acepta como conocimiento.

A continuación se expone de manera general el proceso KDD y se hace especial énfasis en la etapa de minería de datos; posteriormente se presentan sistemas de software para la minería de datos que tienen al SOM como una de sus principales herramientas de visualización de la información; Finalmente se discuten las capacidades de uno de estos sistemas, Viscovery SOMine. Para ilustrar algunas de las capacidades del Viscovery SOMine se reportará la aplicación de este sistema a un problema de Bibliometría.

4.1. Descubrimiento de Conocimiento en Bases de Datos

"La tecnología actual hace muy fácil colectar datos, sin embargo, el análisis tiende a ser lento y computacionalmente costoso"[92]. La naturaleza misma de la práctica computacional inevitablemente impone limitaciones sobre la dimensión y la cantidad de los datos que son analizados; además, no es raro que dentro del planteamiento teórico de los problemas, el espacio de patrones sea infinito. Bajo estas circunstancias, la aplicación de técnicas de análisis inteligente de datos se presenta como una alternativa viable.

Como ya se mencionó, en esta sección se tratará uno de los enfoques de mayor impacto en la aplicación de estas técnicas. Antes de iniciar la discusión de los problemas que involucra el *descubrimiento de conocimiento en bases de datos* conviene describir de manera general la situación actual en el desarrollo tecnológico de las denominadas **bases de datos** (DB).

4.1.1. Estado tecnológico actual y Bases de datos

Dentro del contexto tecnológico actual no se puede hablar de datos sin hacer referencia las estructuras que los almacenan dentro de los dispositivos de cómputo; dichas colecciones de datos almacenados en uno o más archivos, reciben el nombre de **bases de datos**. Las bases de datos están integradas de forma lógica y organizada de tal forma que se facilita el almacenamiento eficiente, la modificación y la consulta de información [7]. En la actualidad, las bases de datos con cientos de campos y tablas, millones de registros y un tamaño de varios gigabytes son bastante comunes. El desarrollo de métodos para tratar con grandes volúmenes de datos implica el diseño y uso de algoritmos más eficientes, aplicación de técnicas de muestreo, métodos de aproximación y procesamiento masivo paralelo. Por lo tanto, para lograr un análisis eficiente de estas grandes colecciones de datos suele ser indispensable el uso de técnicas para el análisis inteligente de datos.

Un problema que surge a partir de la proliferación de las computadoras es el gran número de bases de datos dispersas. Por un lado, la gran cantidad las bases de datos hace necesario contar con un sistema que permita el rápido acceso y el manejo de las mismas. Estos **sistemas de manejo de bases de datos** están especialmente desarrollados para el almacenamiento y la recuperación flexible de grandes masas de datos estructurados [7]. Por otro lado, a pesar de que el movimiento de los datos bajo un control centralizado es deseado, muchas de las bases de datos permanecen donde fueron originalmente formadas; sin embargo, cuando nuevas aplicaciones son desarrolladas este conjunto disperso de datos necesita ser consolidado en una raíz. Esta es la problemática da origen una de las ideas más arrolladoras en la arena del manejo de bases de datos: el **"Data Warehouse"**, que consiste en la combinación sistemática de tecnologías y componentes de hardware que tienen como objetivo la efectiva integración de bases de datos operacionales, en una atmósfera que permita el uso estratégico de los datos.

Estas tecnologías incluyen sistemas de administración de bases de datos relacionales y multidimensionales, arquitecturas cliente/servidor, modelación de metadatos (datos acerca de los datos) y depósitos, interfaces gráficas con el usuario y mucho más; en otras palabras el data warehouse combina: una o más herramientas para extraer campos de cualquier tipo de estructura en los datos, incluyendo datos externos.

Otro concepto importante dentro del desarrollo actual de las bases de datos concierne a las herramientas de análisis. Por sus siglas en ingles **OLAP** (on-line analytical processing) se refiere a sistemas que integran distintas técnicas computacionales usadas para poder realizar análisis de bases de datos on-line. Este análisis puede consistir en predecir tendencias, reconocer patrones y realizar vistas multidimensionales de los datos.

Dentro de este contexto tecnológico es pertinente pensar en la importancia estratégica que tiene el diseño y la aplicación de una métodología que logre integrar todos estos elementos con la finalidad de descubrir conocimiento. A continuación se expone un esfuerzo en esta dirección.

4.1.2. El Proceso de Descubrimiento de Conocimiento en Bases de Datos

El proceso de KDD está definido en términos del objetivo que su nombre indica, el cual se logra como consecuencia del cumplimiento de una serie de etapas. El proceso de KDD tiene como meta principal identificar patrones o establecer modelos; válidos, nuevos, útiles y entendibles; a partir del procesamiento de grandes bases de datos. Se trata de un proceso no-trivial ya que el cómputo involucrado no es simple y por lo tanto en muchas ocasiones se requiere la aplicación de técnicas computacionales de alto desempeño, principalmente desarrolladas dentro de los ámbitos del *aprendizaje de máquina*.

Varios autores [37], [7], [49], [34], [28], [68], [15] coinciden en la consideración de que el proceso KDD implica el cumplimiento de una serie de etapas en las cuales se involucra:

- El prepocesamiento y la preparación de los datos.
- la búsqueda de patrones o modelos.
- La evaluación del conocimiento.

Además se hace énfasis en la naturaleza interactiva e iterativa del proceso. Los patrones descubiertos deben ser válidos en *nuevos datos* con cierto grado de certeza, así mismo deben ser entendibles; si no inmediatamente, sí después de algún post-procesamiento.

En el contexto anterior se supone que se cuenta con una representación matemática adecuada de los datos para que sea válida la aplicación de métodos y técnicas del *análisis inteligente de datos*. En muchas ocasiones la base de datos originalmente considerada, no está dada en términos de una *representación matemática*. Obtener esta representación es una de las tareas a considerar en la etapa de preprocesamiento.

Además se supone que existen medidas cuantitativas para evaluar la validez de los patrones extraídos. Estas medidas se usan para integrar **criterios de evaluación**. En muchos casos es posible definir medidas de certeza, para estimar la fidelidad en la predicción de *nuevos datos*. También se ocupan medidas de utilidad: funciones que establecen un orden lineal dentro del conjunto de patrones, es decir que permitan determinar los patrones preferibles.

Por otro lado, es importante señalar que en este proceso se requiere de un experto que posea un buen dominio en el campo de aplicación, para que interprete y/o valide los resultados de los sistemas de KDD; y para que sea capaz de tomar decisiones en torno a subconjuntos de datos apropiados, clases confiables de patrones y pueda aplicar criterios adecuados para determinar patrones interesantes.

4.1.3. Las Etapas en el Proceso KDD

Los sistemas de KDD deben ser vistos como herramientas interactivas dentro de un proceso iterativo, no como sistemas de análisis automático, o como

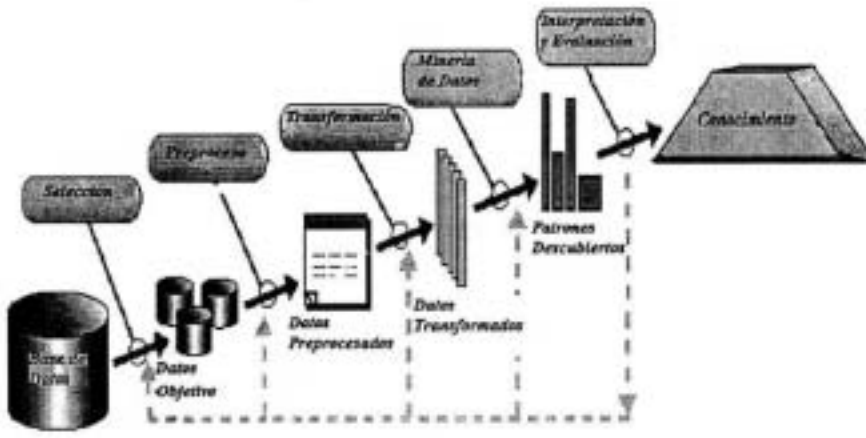

Figura 24: Representación esquemática del *proceso KDD*.

máquinas de producción de análisis en serie. Brachman & Anand [15] ofrecen un enfoque práctico de este proceso haciendo especial énfasis en la naturaleza interactiva del mismo. Antes de iniciar el proceso es necesario desarrollar un entendimiento del dominio de la aplicación, así como contar con un conocimiento relevante previo; de manera que sea posible identificar la meta del proceso KDD desde el punto de vista del beneficiario. Entender el dominio de los datos es un prerrequisito para extraer cualquier tipo de información útil. En la figura 24 se despliega de manera esquemática la sucesión de etapas que constituyen el proceso KDD y que a continuación se describen brevemente:

1. **Crear un conjunto de datos objetivo:** Es decir, seleccionar un conjunto de datos o enfocarse en un subconjunto de variables de las muestras, para llevar a cabo el análisis y el descubrimiento.

2. **Limpieza de datos y preprocesamiento:** Operaciones básicas como limpieza del ruido si es requerido; normalización de los campos que contiene cada dato; implementación del modelo de representación; decidir estrategias para el manejo de campos faltantes en los datos, entre otras tareas. Se ha considerado que esta etapa puede tomar más del 80 % del tiempo total del proceso [29].

3. **Transformaciones:** En los casos en los que es posible usar métodos de reducción de la dimensión o métodos de transformación para reducir el número de variables a considerar y encontrar características útiles para representar los datos dependiendo de la meta del proceso (ver sección

2.3).

4. **Selección del algoritmo de minería de datos:** Esto incluye decidir qué modelos y parámetros pueden ser apropiados (por ejemplo, los métodos para datos categóricos son distintos que los métodos para vectores de componentes reales) y empatar el método particular de minería de datos con el criterio global del proceso.

5. **Minería de datos**: Buscar patrones de interés en una forma representativa o un conjunto de estas representaciones. En algunos círculos esta etapa es considerada como todo el proceso. En la minería de datos se pueden usar muchas técnicas de la estadística y *aprendizaje de máquina*, tales como regla de aprendizaje, inducción por árbol de decisión, clustering, programación lógica inductiva, etc. El énfasis de la investigación de minería radica en el descubrimiento de patrones simples y entendibles.

6. **Interpretación de los patrones minados:** El proceso de KDD no se detiene cuando los patrones han sido descubiertos. El usuario debe entender que ha sido descubierto, para lograr esto se pueden llevar a cabo tareas tales como: la selección u ordenamiento de patrones, **la visualización de los patrones extraídos** o la visualización de los datos dados los modelos extraídos, etc. El proceso de KDD es necesariamente iterativo: los resultados de la minería pueden mostrar que algunos cambios deben ser hechos en la conformación del conjunto de datos y entonces será necesario volver al principio. Muchos enfoques del proceso KDD ponen mucho énfasis en la interpretación o post-procesamiento.

7. **Consolidación de conocimiento descubierto:** La incorporación en otro sistema del conocimiento obtenido para futuras acciones o simplemente documentar y reportar el conocimiento obtenido. Esto también incluye buscar o resolver conflictos entre el conocimiento previo y el extraído.

Como explica la figura , el proceso de KDD puede implicar varias iteraciones entre distintas etapas y contener ciclos. Esto dependerá de los criterios de evaluación del proceso y de los requerimientos específicos de cada aplicación.

4.2. Minería de Datos

El componente de minería de datos en el proceso de KDD comúnmente involucra la aplicación repetida e iterativa de métodos del *análisis inteligente de datos*. "En general, la mayoría de los métodos de minería de datos están basados en técnicas de calidad probada, provenientes de disciplinas como el *aprendizaje de máquina*, el reconocimiento de patrones y la estadística"[68].

4.2.1. Tareas y Algoritmos

Los métodos de minería de datos pueden ser clasificados de acuerdo a las tareas específicas que desempeñan. A continuación se listan las metas básicas y se describe brevemente en que consisten:

- **Clasificación:** Determinar una función que asigna a cada dato una o varias clases predeterminadas.
- **Regresión:** Determinar una función que representa el comportamiento de alguna porción del conjunto de datos o el descubrimiento de relaciones funcionales entre variables.
- **Sumarización:** Encontrar una descripción compacta para un subconjunto de datos, es decir, extraer un sumario o reglas de asociación y aplicar técnicas de visualización multivariada (ver sección 3.3.1).
- **Clustering:** identificar un conjunto finito de clases que describan los datos (ver sección 2.3.3).
- **Modelación de Dependencia:** Encontrar un modelo que describa dependencias significativas entre las variables.
- **Cambio y Detección de Desviación:** Descubrir los cambios más significativos en los datos a partir de medidas previas o valores normativos.

Se pueden identificar tres componentes primarios en cualquier algoritmo de minería de datos: modelo de representación, criterio de evaluación del modelo y método de búsqueda. Este reduccionismo no es completo, es una forma conveniente de expresar los conceptos clave en los algoritmos de minería de datos en una forma unificada y compacta [68]. A continuación se describen cada una de éstas componentes.

- **Modelo de Representación:** Es el *lenguaje* o *modelo matemático* usado para describir los patrones a ser considerados por el algoritmo. Esta representación es de gran importancia ya que, si es demasiado limitada, ninguna cantidad de tiempo de entrenamiento o ejemplos producirá una descripción del conjunto de datos. Además, es muy importante que el analista de datos entienda que las suposiciones con relación a la representación matemática ya que, debe cerciorarse que la representación matemática de los datos sea adecuada al método. Igualmente importante es que el diseñador del algoritmo deje claro cuales serán las suposiciones que serán hechas por un algoritmo en particular; de manera que estas especificaciones sean consideradas en el preprocesamiento.
- **Modelo o Criterio de Evaluación:** Son *medidas cuantitativas* (funciones adecuadas) de la validez de un patrón particular (o un modelo y sus parámetros). Por ejemplo, modelos predictivos son juzgados por la fidelidad empírica en la predicción sobre un conjunto de prueba. En algunos casos es posible definir medidas de certeza, es decir fidelidad estimada de la predicción de nuevos datos; o medidas de utilidad, es decir el establecimiento de funciones que establezcan un orden de preferencia dentro del conjunto de patrones. Un concepto interesante es el de *grado de interés* del patrón descubierto, que comúnmente es tomado como la suma total

del valor del patrón, combinando validez, novedad, utilidad y simplicidad. Las funciones de *grado de interés* pueden ser definidas de manera explícita o implícitamente en los casos donde el sistema de KDD produzca un ordenamiento en los patrones o modelos descubiertos.

- **Método de búsqueda:** Estos métodos normalmente consisten de dos componentes: estimación de parámetros. Una vez que el modelo de representación ha sido fijado, el problema de minería de datos ha sido reducido a un problema de optimización: encontrar los parámetros o modelos de la familia seleccionada en el modelo de representación los cuales optimizan el criterio de evaluación. En la búsqueda de parámetros el algoritmo debe buscar aquellos que optimicen la evaluación del modelo dado un conjunto de datos observados y un modelo fijo de representación. La búsqueda de modelo ocurre como un "loop"sobre el método de búsqueda de parámetros: el modelo de representación es cambiado para que una nueva familia de modelos sea considerada.

En la actualidad existe una gran variedad de algoritmos de minería de datos. Para una revisión concisa de los más populares ver [37]. Los modelos de representación de los patrones minados incluyen árboles y reglas de decisión, regresión no lineal, métodos de clasificación, métodos basados en ejemplos, modelos gráficos de dependencia probabilística (incluyendo redes Bayesianas), modelos de aprendizaje relacional (incluyendo programación lógica) y mapas de conocimiento (representaciones de la organización de la información).

Un punto importante es que cada técnica es típicamente más adecuada para algunos problemas en particular. No existe un método universal de minería de datos y escoger un algoritmo en particular para una aplicación es algo parecido a un arte. En la práctica, gran la mayor parte del esfuerzo se dirige más bien a la formulación adecuada del problema que a la optimización de los detalles del algoritmo.

El descubrimiento de estructuras inherentes en los datos se ha convertido en uno de los principales retos en las aplicaciones actuales de minería de datos. Esta tarea requiere de herramientas que sean al mismo tiempo estables como adaptables. Además, se busca que sean capaces de operar en espacios de una dimensión alta.

Recientemente las *redes neuronales artificiales* han adquirido una importancia significativa al ser utilizadas como algoritmos de minería de datos principalmente en problemas de clustering [7]. Sin embargo, a pesar de que las *redes neuronales* son poderosas maquinas de reconocimiento, aún no existe nada tan poderoso como la habilidad human para ver y reconocer patrones. Por tal motivo, las técnicas de visualización juegan un papel muy importante en el análisis de los resultados obtenidos como producto de la aplicación de algoritmos de minería de datos. De hecho, la visualización es comúnmente usada desde la etapa de prepocesamiemto de los datos como ayuda en la selección de variables a ser consideradas [48], [66]. A continuación se aborda con mayor profundidad el tema de la visualización de información en el proceso de descubrimiento de conocimiento.

4.2.2. Visualización de la información y KDD

Una vez aplicado algún algoritmo de minería de datos, una tarea importante para la interpretación de los resultados es la visualización de la información. De manera independiente al proceso KDD, la exploración visual de información en espacios adecuados es un tema estratégico. Existe una gran variedad de paradigmas y métodos de visualización. Gracias a los avances tecnológicos, una gran variedad de sistemas de software han sido desarrollados en los últimos años para facilitar la visualización y el análisis de información. Entre los principales problemas a los que se enfrentan los desarrolladores de métodos y medios para la visualización de la información se encuentran [17]:

- **Reducir el número de datos y encontrar estructuras:** La exploración de grandes espacios de información sin información previa acerca de su estructura interna, muchas veces requiere de preprocesamiento para lograr reducir el tamaño de los datos activos. Este preprocesamiento puede implicar "filtrar campos poco interesentes.º agrupar datos similares en grupos homogéneos (clustering). Métricas o medidas de similitud apropiadas deben ser aplicadas para obtener estructuras dentro de espacios multidimensionales. En este sentido se puede observar el paralelo que existe en el proceso de visualización de información con el de descubrimiento de conocimiento.

- **Visualización de conjuntos de información:** El éxito de la visualización depende en gran medida de su habilidad para entender una gran variedad de tareas de exploración, por ejemplo: revisiones globales, zoom en temas específicos, visualización simultanea en distintos escenarios, etc.. Es común que dentro de un mismo problema, distintos métodos de visualización se requieran para revelar estructuras y contenidos.

- **Visualización del marco de referencia:** Para lograr exploraciones efectivas dentro de espacios referenciados se requiere la combinación de un despliegue adecuado de los marcos de referencia espaciales junto con la visualización de estructuras complejas. Estas referencias pueden llevarse a cabo mediante en el despliegue de etiquetas.

Las técnicas de visualización de información aun no han sido suficientemente explotadas dentro del contexto del descubrimiento de conocimiento en bases de datos. Una de las causas es la gran variedad de distintos tipos de datos y la falta de algoritmos de preprocesamiento adecuados. A pesar de que existe una gran cantidad de sistemas para la minería de datos que hacer uso de técnicas de visualización de información, la mayoría de estos sistemas, o son muy genéricos o demasiado especializados en cierto tipo de datos concernientes a cierto tipo de problemas. Por otro lado, existen disciplinas en las cuales se realizan tareas propias de la visualización de información y que sin embargo, no han utilizado las técnicas desarrolladas por los especialistas; es decir, han desarrollado sus propias técnicas de análisis pero estas no han sido incorporadas en un sistema de

visualización de información. Un ejemplo de estas disciplinas es la cienciometría que busca la visualización de estructuras intelectuales, basada en literatura científica. Sin embargo, la interacción entre la cienciometría y los practicantes de la visualización de la información es incipiente. Un ejemplo de esta interacción es el trabajo realizado por el grupo de investigación del *Instituto Finlay* en la *Habana, Cuba*; en colaboración con un grupo en el *Laboratorio de Dinámica no-Lineal*, de la *Facultad de Ciencias*, de la *UNAM*. Entre los trabajos de este grupo en esta linea se encuentran [87], [89].

Dado este escenario, es de esperarse que en el desarrollo de los nuevos sistemas de minería de datos, el tema dc visualización de información tenga un papel cada vez más prioritario. Los recientes avances en las áreas de realidad virtual*, hacen que sea fácil imaginar la posibilidad de interactuar con espacios abstractos de información.

4.2.3. Retos en el Desarrollo Actual de Sistemas de Software

Como ya se ha mencionado, en los últimos años el desarrollado sistemas de minería de datos ha ido incrementándose y se espera que esta tendencia de investigación y desarrollo tenga un impacto cada vez mayor en distintos campos. El principal fundamento de esta estimación es que han sido colectadas grandes sumas de datos y casi en cualquier campo existe la necesidad del entendimiento y el buen uso de la información para la toma de decisiones.

Existe una gran variedad de temas que actualmente representan retos en el desarrollo de sistemas de software para la minería de datos. De entre estos temas se pueden mencionar: el diseño de lenguajes, el desarrollo de métodos y sistemas eficientes y efectivos, la construcción de atmósferas interactivas e integradas y la aplicación de herramientas para resolver una gran variedad de problemas. Estos temas representan tareas importantes para los investigadores y los desarrolladores de sistemas. Sin embargo, dentro del desarrollo de los sistemas las principales problemáticas se derivan de las diversas características de las bases de datos, las limitaciones computacionales y del dominio del conocimiento por parte del usuario. A continuación se exponen los principales retos del diseño y desarrollo de este tipo de sistemas de software. Esta lista no es exhaustiva, sin embargo, intenta brindar una aproximación al tipo de problemas que los practicantes de KDD tienen que afrontar:

- **Overfitting:** Tiene lugar cuando un algoritmo busca los mejores parámetros para un modelo y es usando un conjunto limitado de datos; bajo estas condiciones puede que no se encuentren patrones generales válidos en los datos, pero también puede pasar que no se distinga el ruido en ese conjunto, como resultado se obtendrá un desempeño pobre del modelo sobre nuevos datos de prueba. Entre las posibles estrategias para solucionar este problema se incluyen validación cruzada, regularización y otras estrategias sofisticadas de estadística.

- **Datos y conocimiento dinámicos:** La evolución de los datos en el tiempo puede hacer que los patrones previamente descubiertos sean inválidos

en el momento de ser evaluados. Adicionalmente, las variables determinadas en una base de datos para una aplicación dada, pueden ser modificadas, borradas o aumentadas con nuevos valores en el tiempo. Posibles soluciones incluyen métodos incrementales para actualizar los patrones y probar los cambios. Al usar estos métodos se abre la posibilidad de descubrir patrones de cambio.

- **Datos faltantes y ruidosos:** atributos importantes pueden estar faltando si la base datos no ha sido diseñada tomando en cuenta la aplicación de técnicas de descubrimiento de conocimiento. Entre las posibles soluciones incluyen estrategias estadísticas sofisticadas para identificar variables escondidas y dependencias.

- **Relaciones complejas entre campos:** Pueden encontrarse valores jerárquicamente estructurados, relaciones entre atributos y medios sofisticados para representar el conocimiento acerca de los contenidos en una base de datos requerirán algoritmos que puedan utilizar eficientemente esta información. Históricamente, los algoritmos de minería de datos han sido desarrollados para registros simples de atributos, sin embargo, nuevas técnicas para derivar relaciones entre las variables están siendo desarrolladas, para mayor detalle consultar [49] y [34].

- **Entendimiento de los patrones:** Como hemos visto, en muchas aplicaciones es necesario hacer a los patrones más entendibles para los seres humanos. Las posibles soluciones incluyen representaciones gráficas, reglas de estructura, generación de lenguaje natural y técnicas para la visualización de los datos y el conocimiento. Las estrategias de refinamiento pueden ser usadas para dar dirección a un problema relacionado con el conocimiento descubierto que puede ser implícita o explícitamente redundante.

- **Interacción con el usuario y conocimiento previo:** Muchos métodos y herramientas actuales para el KDD no son realmente interactivas y no es fácil incorporar el conocimiento previo acerca del problema. El uso del "dominio del conocimiento"es importante en todos los pasos del proceso de KDD. Enfoques Bayesianos usan las probabilidades previas sobre los datos y las distribuciones como una forma de codificar el conocimiento previo. Otros enfoques utilizan las capacidades deductivas de las bases de datos para descubrir conocimiento, el cual es usado como guía para la búsqueda en la minería de datos.

- **Integración con otros sistemas:** Un sistema de descubrimiento particular y solitario puede que no sea muy útil. La integración de sistemas otros sistemas de administración de bases de datos, de hojas de cálculo y con herramientas de visualización de información. Ejemplos de sistemas integrados para KDD son descritos en [28] y [68].

Dadas las propiedades del algoritmo SOM y las ventajas que ofrece, cuando es implementado en un sistema de minería de datos facilita la solución a muchos de los retos anteriormente planteados. Por tal motivo, actualmente este

algoritmo representa una herramienta fundamental en una gran variedad de sistemas para la minería de datos y la visualización de información. En la siguiente sección se mencionan algunos de estos sistemas.

4.2.4. Sistemas de Análisis y Visualización de Información Basados en el SOM

Las *redes neuronales* no supervisadas introducidas por Kohonen han sido utilizadas extensivamente en para realizar tareas de Minería de Datos y Visualización de la Información. El despliegue visual del conjunto de datos permite realizar la búsqueda y el descubrimiento de información valiosa a través de la exploración de mapas de conocimiento.

Diversos grupos de investigación se han apoyado en el algoritmo SOM para desarrollar sistemas de software, a partiendo de los datos, llevan a cabo el entrenamiento de una *red neuronal* y producen toda una gama de mapas de conocimiento. A continuación se presentan algunos ejemplos de sistemas para la minería de datos y la visualización de información que utilizan al SOM [33].

Kensington Discovery Edition

Sistema desarrollado por la compañía de software Inforsense que se origina del grupo de investigación en Minería de Datos, afiliado a "the Parallel Computing Centre" en el "Imperial College of Science, Technology and Medicine" de Inglaterra.

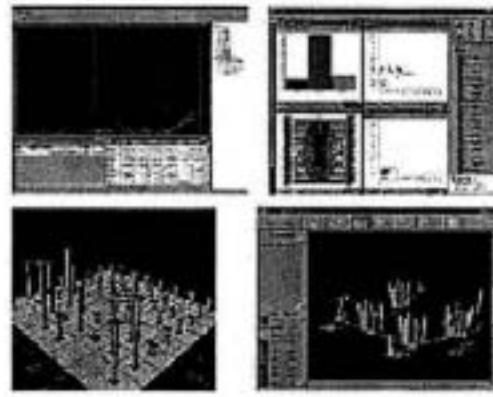

Figura 25: Pantalla del *Kensington Discovery Edition*.

Las principales aplicaciones de este sistema se encuentran en el proceso de descubrimiento de drogas a partir del estudio genómico. Los principales usuarios son la industria farmacéutica y grupos académicos de investigación.

Además del SOM, utiliza este sistema utiliza métodos de clasificación basados en árboles de decisión, clasificación Bayesiana y Redes Neuronales de Retro-Propagación para el descubrimiento de reglas de asociación. La herramienta

provee una visualización interactiva que corre simultáneamente sobre distintas bases de datos, hojas de cálculo, texto o documentos web. También es útil para la recuperación de información, el manejo de distintos contextos de conocimiento y el almacenamiento dinámico de datos.

Clementine

Desarrollado por la compañía SPSS, más famosa por sus paquetes de análisis estadístico, Clementine es un sistema para la Minería de Datos que permite un rápido desarrollo de modelos predictivos utilizando la información de expertos; y posteriormente permite incorporar estos modelos en las operaciones. El mismo sistema también puede ser utilizado en aplicaciones bioinformáticas y de minería de texto.

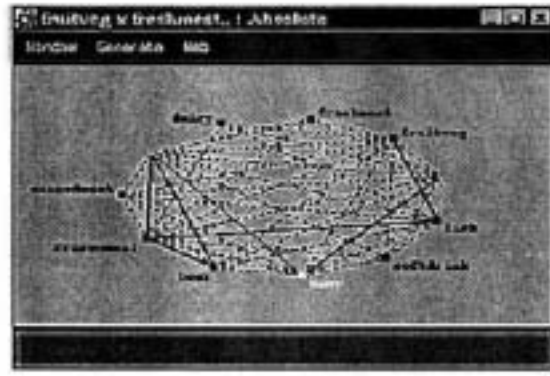

Figura 26: Pantalla del *Clementine*

Además del SOM, el sistema utiliza: métodos estadísticos de regresión, Retro-Propagación y el algoritmo de *aprendizaje de máquina* ID3/C5.0. Utiliza varios métodos de análisis de datos. También posibilita la importación y exportación de datos, la generación de reportes, el pre-procesamiento, la modelación y la programación visual.

IBM Intelligent Miner

Desarrollado por la compañía IBM, este sistema tiene el enfoque de la "Inteligencia Empresarial", una de las principales aplicaciones de este sistema es la minería de datos recopilados en puntos de venta.

Los métodos utilizados por el sistema son: árboles de decisión y varios métodos de regresión para la clasificación, asociación y el descubrimiento de secuencias.

DataEngine

Este sistema es una herramienta para el análisis inteligente de datos que integra métodos estadísticos con Redes Neuronales y tecnologías fuzzy. Diseñado e implementado por la compañía alemana M.I.T. (Management Intelligenter

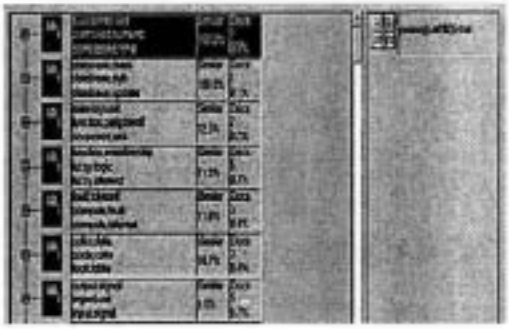

Figura 27: Pantalla del *IBM Intelligent Miner*

Technologien GmbH) la herramienta provee de módulos de adquisición y visualización de datos.

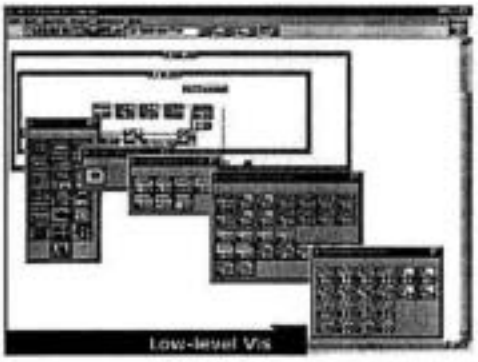

Figura 28: Pantalla del *DataEngine*

El sistema está especialmente diseñado para realizar y visualizar Clustering. Además, el sistema contiene una lista extensa de funciones matemáticas y métodos estadísticos, que pueden ser utilizados de manera simultanea.

NGO NeuroGenetic Optimizer

Desarrollado por BioComp Systems, Inc. es un sistema principalmente en *redes neuronales* como Retro-Propagación y una variante del algoritmo SOM básico llamada "SOM temporal", además incorpora técnicas de regresión generalizada y algoritmos genéticos.

El mayor atributo de la herramienta es su habilidad de realizar múltiples modelaciones de manera simultanea. Los algoritmos genéticos son utilizados para optimizar a las *redes neuronales*.

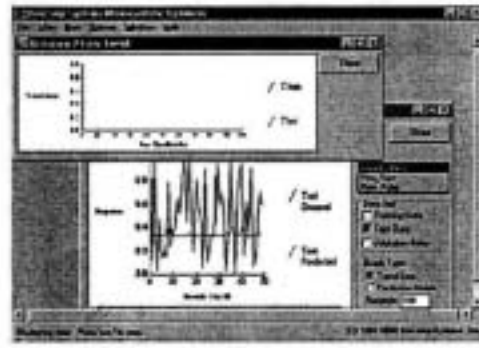

Figura 29: Pantalla del *NGO NeuroGenetic Optimizer*

Vispoint

Desarrollado por la compañía Vispoint este sistema está totalmente basado en el SOM básico.La herramienta utiliza como input cualquier hoja de cálculo o

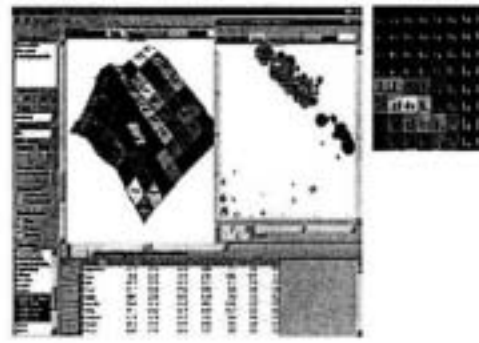

Figura 30: Pantalla del *Vispoint*

texto. Provee funciones para el pre-procesamiento y distintas representaciones de los datos para la visualización.

En la siguiente sección examinaremos con más detalle a uno de estos sistemas, "Viscovery SOMine".

4.3. Viscovery SOMine

El sistema de software Viscovery SOMine, desarrollado por la compañía austriaca de software Eudaptics, es una herramienta para el análisis avanzado y el monitoreo de conjuntos de datos numéricos; su motor principal es una variante del algoritmo Batch Map presentado en la sección 3.2.4..

Este sistema provee de medios poderosos para analizar conjuntos de datos con una estructura compleja, sin necesidad de contar con algún tipo de información estadística a priori. El usuario es guiado durante el proceso de entrenamiento por un ambiente de ventanas bien definidas. Una vez computados los mapas, la representación de información puede ser sistemáticamente transformada de tal forma que se pueden utilizar distintas técnicas de visualización y de clustering. Además el sistema permite que alguna información numérica pueda ser solicitada por el usuario en cualquier momento. A continuación se desglosan las capacidades del sistema durante las etapas del proceso KDD; comenzando por el preprocesamiento hasta llegar a la exploración de los mapas y el descubrimiento de conocimiento. El contenido con relación al uso del ViscoverySOMine está basado en [27].

4.3.1. Sumario de Componentes

El primer paso en la aplicación del Viscovery es la selección de la fuente de datos. Esta fuente puede estar en formato texto (*.txt) o Microsoft Excel Workbook (*.xls), en cualquiera de los dos casos, el archivo debe ser una tabla de datos numéricos en la que se especifique el nombre de cada componente. Una vez seleccionada la fuente de datos, el sistema despliega un **sumario de componentes** (ver figura 31).

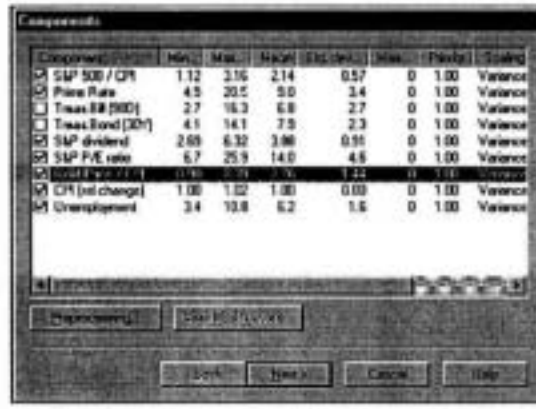

Figura 31: Ventana del sumario de componentes

En esta ventana se puede determinar cuales componentes serán usadas durante el proceso de entrenamiento para la creación del mapa. En ella se especifican los siguientes parámetros:

- **Componente:** especifica el nombre de la componente.
- **Mínimo:** especifica el valor mínimo de la componente en el conjunto de datos.

- **Máximo:** especifica el valor máximo de la componente en el conjunto de datos.
- **Media:** especifica el valor promedio de la componente en el conjunto de datos.
- **Desviación Estándar:** especifica la desviación estándar del conjunto de datos en la componente.
- **Valores faltantes:** es el número de datos que no tienen especificado su valor en la componente.
- **Prioridad:** este es un parámetro que determina la importancia de esta componente.
- **Escalamiento:** especifica el método de escalamiento aplicado en la componente. Los posibles métodos son: *Varianza, Rango y Ligado.*
- **Transformación:** especifica el tipo de transformación aplicado a la componente. Los posibles tipos son: *Ninguna, Sigmoidal y Logarítmica.*

Tanto el factor de prioridad, como el método de escalamiento y el tipo de transformaciones, pueden ser establecidos y modificados en el **módulo de preprocesamiento**.

4.3.2. Preprocesamiento

El módulo de preprocesamiento del Viscovery SOMine brinda la posibilidad de *pesar* la importancia de las componentes por medio de la asignación de **Factores de Prioridad** o la aplicación de algún método de **Escalamiento**; y de transformar la distribución de las componentes por medio de la aplicación de una **Transformación.**

Estas modificaciones se implementan con la finalidad abolir la influencia de datos extremos y normalizar las distribuciones de los datos sobre las componentes. A continuación se precisa en que consiste cada una de estas modificaciones.

Factor de Prioridad

El factor de prioridad da un peso adicional a la componente al multiplicar su escala interna por un valor. Es decir, cada uno de los valores de los datos es esa componente serán multiplicados por este factor.

Si el factor de prioridad es fijado más grande que uno, esta componente será internamente extendida en su rango.

En contraste, si se utiliza un número menor que uno, esta componente será comprimida, por lo tanto se volverá menos importante durante el proceso de entrenamiento.

Cuando el valor de este factor es cero, la componente se vuelve irrelevante para el proceso, este caso particular es de gran importante ya que de esta manera

puede haber parámetros asociados a los datos que no tengan ningún impacto en el ordenamiento de los datos.

Escalamiento

El escalamiento puede ser entendido en términos del **Factor de Prioridad**. Por default cada componente es escalada por su varianza, es decir, si su rango es menor que 8 veces la desviación estándar el hecho de dividir entre su desviación estándar ; de otro modo el escalamiento es puesto en función del **Rango**.

Este método heurístico ha demostrado ser un escalamiento bastante natural y ha sido introducido por conveniencia. Por tal razón, es conveniente dejar que Viscovery realice el escalamiento. Cambiar el escalamiento de la propuesta por default a la opuesta (i.e. de varianza a rango o viceversa) tiene el mismo efecto que aplicar un factor de prioridad más alto. El impacto relativo de esta componente en el proceso de entrenamiento será creciente. El **ligamiento de componentes** implica aplicar el mismo factor de prioridad a las dos componentes.

Modificación de Datos

Al seleccionar un rango para una componente y aplicar un factor de amplificación, es posible adicionar o remover registros de datos en una región específica del espacio de datos. Al amplificar los rangos es una acción delicada y tiene que ser manejada con cuidado. Menos crítico es la supresión de datos puede ser muy útil en la eliminación de datos extremos en mejorar la visualización de los histogramas. Si el usuario está interesado en regiones en particular del conjunto de datos, una amplificación puede ser un medio eficiente.

Cualquier modificación del conjunto de datos no solo será reflejada en el histograma de un componente en específico, también cambia los histogramas de otros componentes como en todos los registros de datos, no solo las componentes son añadidas o removidas.

Transformación

A aplicar una transformación el usuario puede influir la densidad de características en la distribución de una componente. Los dos tipos de transformación además de la identidad son logarítmica y sigmoidal. La aplicación de una transformación redefine la representación interna de una componente con la función especificada. Esto ciertamente cambia las distancias entre los registros, esto tiene un impacto intrínseco en las relaciones de vecindad dentro del conjunto de datos.

Supóngase que los registros en una componente en específico está muy cargado hacia algún lado. Al aplicar una transformación logarítmica equiparará la distribución, de tal manera que la función logarítmica brindará una mayor resolución a los valores pequeños en el histograma. En consecuencia valores pequeños de esta componente tendrán mayor impacto en el Clustering de los datos.

De manera similar la aplicación de una transformación sigmoidal puede tener como resultado una distribución más balanceada al reducir el centro del histograma. Usando la configuración de default la aplicación y la activación de la

transformación sigmoidal puede ser utilizado para reducir el impacto de datos extremos durante el proceso de entrenamiento.

4.3.3. El SOM en el Viscovery

El algoritmo de minería de datos que utiliza el Viscovery SOMine está basado en el Batch Map. Esta versión del algoritmo está mejorada por una mezcla de técnicas que aceleran el proceso de entrenamiento.

El algoritmo Batch Map utilizado por el Viscovery determina el valor de los vectores de referencia calculando un promedio pesado de los promedios de los conjuntos de Voronoi en cada nodo. El peso para cada dato es determinado por una función Gaussiana que depende de la distancia física entre cada nodo dentro de una vecindad del nodo donde este dato quedó empatado. El radio de la función Gaussiana es llamado tensión. Teóricamente, todos los datos contribuyen, con distinto peso, a cada nodo en el mapa. Sin embargo, la vecindad es truncada a cierta distancia, es decir, cuando la función Gaussiana es muy cercana a cero; por lo tanto la contribución de un dato con otros nodos, distintos al nodo que mejor lo represeta, prácticamente se desvanece.

Durante el proceso de entrenamiento, el número de nodos en el mapa no está fijo sino crece, desde un número pequeño hasta el número deseado de nodos. La razón entre el largo y el ancho del mapa inicial es determinada por la magnitud de las dos componentes de mayor rango en el conjunto de datos. Para todos los pasos intermedios, la razón aparente en el mapa se mantiene lo más parecido posible a la razón final. Los valores de los vectores de referencia en los mapas nuevos son inicializados al interpolar los valores de 3 nodos en el mapa previo.

Cada mapa es entrenado para cierto número de iteraciones utilizando una tensión decreciente. Cuando se incrementa el número de nodos, este incremento es compensado por un correspondiente aumento en la tensión. Este proceso, donde crece el tamaño del mapa junto con incrementos y decrementos graduales de la tensión, es equivalente a usar el tamaño final del mapa desde el principio, pero es más rápido ya que opera con mapas más pequeños.

4.3.4. Proceso de Entrenamiento

La ejecución del proceso de entrenamiento depende de la **determinación de los parámetros básicos del mapa**, estos parámetros básicamente consisten en la determinación de la arquitectura del mapa y en la forma de las función vecindad que será utilizada para la actualización de los pesos de los vectores de referencia. Estos parámetros básicos son:

- **Número de Nodos:** Se puede determinar el número de nodos que se pretende tenga el mapa, el programa automáticamente ajusta el número de nodos de tal manera que pueda ser representado de manera rectangular.
- **Razón del Mapa:** Se especifica la razón (entre largo y alto) del mapa que se desea. Si se selecciona automáticamente, el radio del mapa será determinado a partir del plano principal del conjunto de datos fuente. Se

puede elegir la opción de que la forma de la retícula sea cuadrada lo cual indica que la razón es 1.

- **Tensión:** Especifica la tensión en el mapa resultante. Este valor determina la habilidad del mapa para representar a los datos de entrada, valores altos (≥ 2) en este parámetro tendrán como consecuencia un mapa muy rígido*. Los valores típicos para este parámetro son entre 0,3 y 2,0.

Además de estos parámetros básicos, existen otros que determinan la manera en que la retícula cambia a lo largo del proceso de entrenamiento, al igual que la tensión, estos parámetros forman parte de los parámetros de la variante del SOM que utiliza el sistema.

La configuración de un vector de parámetros de entrenamiento se denomina **Cédula de Entrenamiento**. Estas configuraciones pueden ser modificadas por el usuario de acuerdo a las necesidades específicas. Los parámetros sobre los cuales se pueden realizar las modificaciones son:

- **Factor de Escalamiento:** este factor determina que tan rápido crecerá el mapa. Cada paso del crecimiento el número horizontal de nodos se incrementa de acuerdo a este factor. (El número vertical de nodos será fijado de acuerdo a los requerimientos del radio)

- **Altura del mapa inicial:** el mapa inicial tendrá tantos nodos verticales como se especifique aquí. El número horizontal de nodos será determinado de acuerdo a los requerimientos del radio. La altura mínima posible son tres nodos.

- **Cédulas predeterminadas de entrenamiento:** seis de estas configuraciones de parámetros son predeterminados de acuerdo a la conveniencia del usuario, las opciones son: Rápido, Normal y Preciso, cada una de estas opciones puede hacerse de manera veloz o exacta. Estas configuraciones corresponden a cédulas internas dentro del proceso de entrenamiento. A mayor precisión se tienen más iteraciones y por lo tanto el entrenamiento toma más tiempo.

Una vez determinada la cédula de entrenamiento, se lleva acabo el proceso de entrenamiento, mientras este corre se puede observar su evolución del proceso en una ventana que despliega las curvas de error de cuantización y distorsión normalizada; además en esta ventana indica la duración estimada del entrenamiento.

La distorsión normalizada mide que tan bien el mapa esta aproximando la distribución de los datos. Es normalizada con respecto a la tensión intermedia, el número de nodos y el número de datos. Por otro lado, el error de cuantización de un nodo es el promedio de las distancias cuadradas de este nodo a todos los datos que son asociados. El error de cuantización del mapa es la suma de todos los errores de cuantización de sus nodos.

Concluido el proceso de entrenamiento, el sistema despliega toda una gama de mapas y visualizaciones; el siguiente paso se refiere a la exploración de los mapas. En este proceso de exploración se puede obtener: relaciones entre variables, visualización de clusters y la inspección de la "preservación de la topología" en el mapa; entre otras cosas. En la siguiente sección se examinan cada uno de estos aspectos.

4.4. Exploración y Uso de los Mapas

De manera tal que la exploración de los mapas se puede entender como el proceso por medio del cual, se tiene la intención de extraer información valiosa, partiendo de la inspección visual de una gama de mapas.

En la práctica es de esperarse que existan aspectos en el mapa que sean más significativos dentro del contexto de una aplicación en particular; por lo tanto, es difícil establecer una métodología general para la exploración de los mapas. Sin embargo, las propiedades generales de los mapas pueden ser utilizadas durante el proceso el proceso de exploración y son significativas en casi cualquier aplicación . Estas propiedades son la preservación de la topología y a la distribución de los datos en un despliegue ordenado; basándose en ellas es posible el establecimiento de relaciones entre variables, la visualización de clusters y la inspección de relaciones de vecindad entre los nodos en el mapa. A continuación se examinan las distintas visualizaciones y la utilidad que cada una de estas puede tener para la extracción de información útil acerca del conjunto de datos.

4.4.1. Mapas de Componentes

El despliegue de los **Mapas de Componentes** tiene la particularidad de representar la distribución de los valores de cada variable de los datos en un mapa. Este mapa denominado "component picture" representa el promedio de los valores de la variable correspondiente a los datos asociados a cada nodo. La distribución de estos promedios se puede visualizar por medio de una escala de color que corresponde al rango de valores que los datos toman en la variable correspondiente. Los valores mínimos están representados por azules, los intermedios por verdes y amarillos, y los valores máximos por rojos.

La exploración de los mapas de componentes puede ayudar a establecer relaciones entre las distintas variables. Como un ejemplo de estas relaciones, en la figura 32 se observan los mapas de las componentes v_i y v_j: estos mapas presentan un patrón muy similar en cuanto a su coloración; la principal diferencia es que donde un presenta color azul el otro presenta color rojo y viceversa.

Si se extraen los valores de los datos en estas variables se puede observar el siguiente comportamiento:aquí (figura 33) se observa que es posible encontrar un modelo lineal que represente la relación entre las dos variables en una buena porción de los datos.

Además de los mapas correspondientes a cada una de las variables, existe la posibilidad de visualizar otros dos mapas: **Frecuencia** y **Error de Cuantización** (ver figura 34). El mapa de Frecuencia indica cuantos datos del conjunto

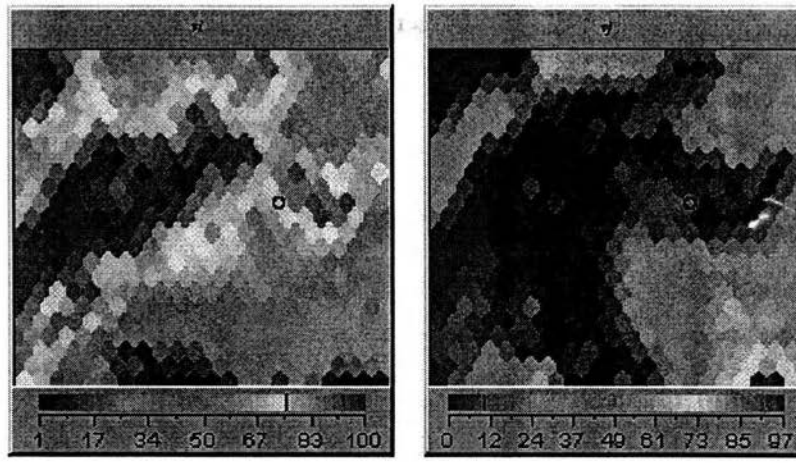

Figura 32: Mapas de las componentes v_i y v_j.

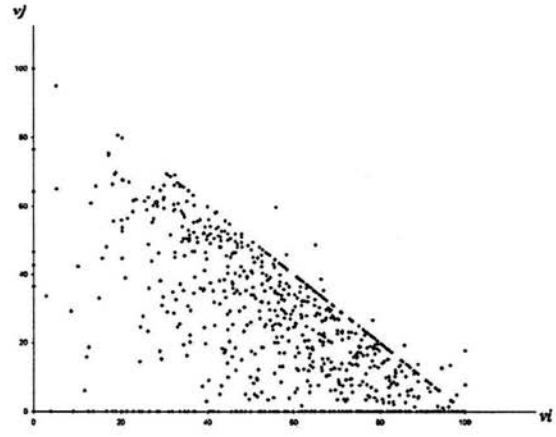

Figura 33: Proyección de los datos sobre las variables v_i y v_j.

de entrenamiento han sido asociados a cada nodo. En este mapa, entre más oscuro es el rojo de un nodo más alta será la frecuencia de los datos asociados a este nodo. El mapa de Error de Cuantización muestra el error de cuantización en cada nodo. Esta medida es utilizada para determinar que tan bien cada nodo representa a los datos asociados. Entre más oscuro es el verde de los nodos, más grande es el error de cuantización.

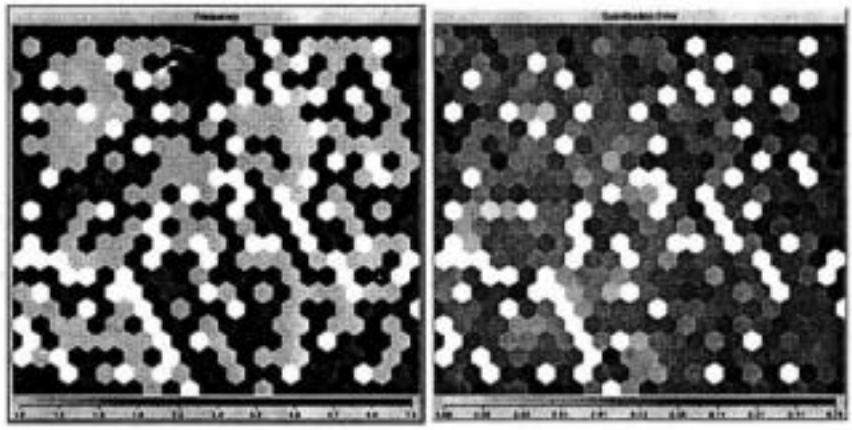

Figura 34: Mapas de las componentes *Frecuncia* y *Error de Cuantización*.

4.4.2. Mapa de Clustering

Además de la visualización de los mapas de componentes, Viscovery divide el mapa en regiones, que representan clusters del conjunto de datos, para realizar esta división el sistema brinda la posibilidad de utilizar tres Técnicas distintas.

Estas técnicas combinan algunos algoritmos clásicos de clustering con la topología inducida por el mapa y algunas medidas de distancia especialmente definidas para dicha topología. Además, el sistema permite evaluar el desempeño de los algoritmos de clustering por medio del cómputo de criterios de clustering. Los métodos disponibles tienen en común que el usuario puede seleccionar a las componentes que son consideradas en la generación de las particiones. A continuación se describen cada uno de estos métodos. **Ward Clusters**

Esta opción es equivalente a aplicar el algoritmo clásico de Ward a los vectores de referencia de las neuronas en el mapa. Es decir, en lugar de aplicar el algoritmo Ward a todo el conjunto de datos, el método solo se aplica al conjunto de vectores de referencia de las neuronas. En el punto de partida del algoritmo, cada neurona representa a un cluster. En cada paso dos clusters distintos se unen en uno solo. Los clusters seleccionados son aquellos que tienen la distancia mínima entre todas las distancias entre clusters, en este caso es la distancia entre los **centroides** (ver sección 2.3.3), que es la misma que utilizada en el algoritmo

Ward original. La principal desventaja de este método es que los clusters que define no necesariamente son regiones conexas en el mapa.

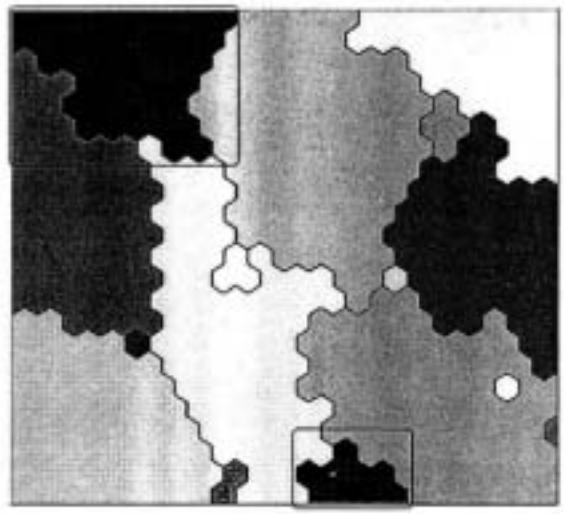

Figura 35: Aplicación del método *Ward*.

En la figura 35 se visualiza el resultado de la aplicación de este algoritmo sobre un mapa, las regiones enmarcadas con el rectángulo corresponden contienen neuronas que pertenecen al mismo cluster. El sistema computa un criterio de Clustering el cual es aplicado a cada miembro de la jerarquía producida por el algoritmo.

SOM-Ward Clusters

El método combina la información local del ordenamiento en el mapa con el algoritmo Ward de Clustering jerárquico, de esta manera se garantiza que las regiones definidas sobre el mapa resultarán conexas. Como en el algoritmo Ward original, en el punto de partida del algoritmo SOM-Ward, cada nodo es considerado un cluster. En cada paso dos clusters son unidos en uno solo: aquellos con la mínima distancia de acuerdo a una medida de distancia especial denominada SOM-Ward. Esta distancia toma en cuenta cuando dos clusters son adyacentes en el mapa, lo que tiene como consecuencia que solo se pueden unir clusters adyacentes en el mapa.

De esta manera, este algoritmo solo produce clusters representados por regiones conexas en el mapa (ver figura 36). Al igual que en el método anterior, el sistema computa un criterio de clustering el cual es aplicado a cada miembro de la jerarquía producida por el algoritmo SOM-Ward.

SOM-Single-Linkage Clusters

Este algoritmo primero determina los separadores, es decir, líneas entre nodos vecinos. Los clusters se originan a partir de regiones delimitados por curvas

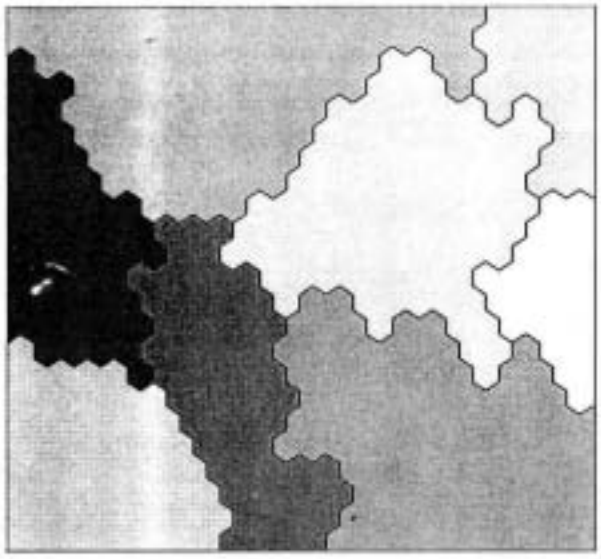

Figura 36: Aplicación del método *SOM-Ward*.

cerradas de separadores. Para determinar cuando entre dos nodos vecinos hay un separador es utilizado el parámetro umbral de cluster: Si la distancia en el espacio de entrada, entre dos nodos vecinos es mayor que el umbral, un separador es puesto en la arista de adyacencia entre los dos nodos.

En este método se define un índice de similitud entre los nodos dado por el umbral de similitud y la distancia en el espacio original de los datos. En este método se puede dar la situación de que entre dos nodos dentro de un cluster exista un separador, a este tipo de separador se le denomina separador interno. El otro parámetro que se toma en cuenta para la visualización es el de tamaño mínimo de cluster: cuando una curva de separadores define una región con un número de nodos menor que este número, la coloración de la región es gris oscuro. Las regiones de este tipo son denominadas áreas de separación.

4.4.3. Proyección de los Datos

Una de las propiedades más importantes del algoritmo SOM es la denominada "Preservación de la Topología" la cual significa que datos similares corresponderán a nodos cercanos en el mapa. En el Viscovery esta relación de cercanía puede ser observada por medio de la activación de la opción "Neighborhood". Esta opción permite mostrar aquellos nodos cuyo vector de referencia es más cercano a un nodo en específico.

La vecindad (Neighborhood) de un vector de referencia es el conjunto (con tamaño predeterminado) de nodos en el mapa cuyos vectores de referencia son

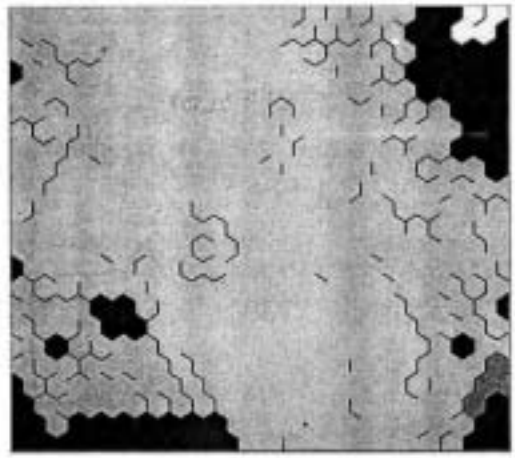

Figura 37: Aplicación del método *SOM-Single-Linkage-Clusters*

más cercanos. Este conjunto es visualizado con marcas rojas que varían su intensidad de acuerdo a la cercanía; entre más cercano el color es más oscuro.

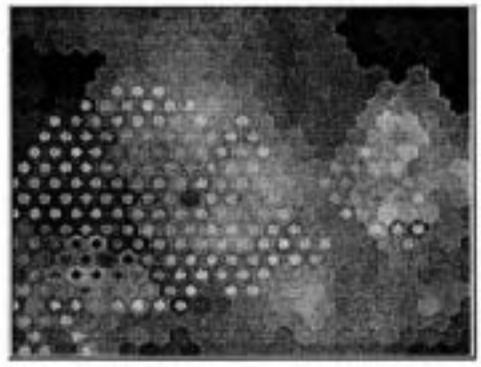

Figura 38: Relaciones de vecindad de un nodo

En la figura 38 se puede observar el conjunto vecindad del nodo que está señalado con un círculo negro. En este ejemplo también se observa que el conjunto de vecindad de un nodo no necesariamente es conexo en el mapa. Esta observación indicaría que la proyección del conjunto de datos no está preservando la topología de manera exacta. La determinación de "medidas de preservación topológica" es uno de los temas más importantes en el desarrollo de herramientas analíticas para el SOM. Este tema será tratado en la sección 4.2.3.

Una forma de visualizar las relaciones de cercanía entre los vectores de ref-

erencia de manera global es por medio del método U-Matrix. Para cada nodo se calculan los promedios de las distancias entre este y sus vecinos inmediatos en la retícula. Por lo que cada nodo tienen asociado un valor, el conjunto de estos valores es asociado a una escala cromática y a cada nodo se le asocia un color.

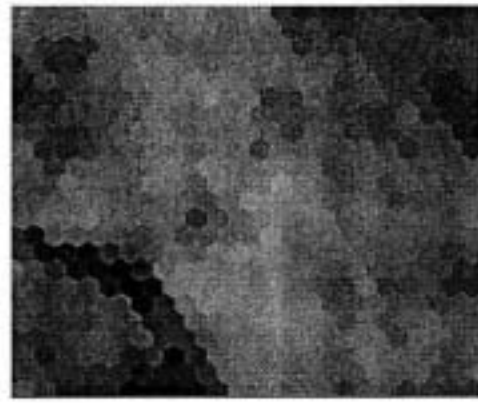

Figura 39: Visualización de la *U-Matrix*

4.4.4. Interpretación y Evaluación de los Mapas

A pesar de que la metáfora visual provista por los mapas puede ser útil para lograr entendimiento intuitivo, no está claro para que tipo de aplicaciones esta representación es válida. Dado que el SOM trata sobre todo preservar estructuras locales, la interpretación de los mapas debe hacerse localmente, es decir basándose en las relaciones entre los vectores de referencia dentro de los conjuntos vecindad de cada nodo. Aunque la interpretación de la estructura global puede ser útil, debe tenerse especial cuidado cuando se infieren propiedades globales del conjunto de datos a partir de la representación bidimensional hecha por el SOM.

Otro elemento que puede ser útil para la interpretación es la asignación de etiquetas sobre los nodos. Si los datos además de vectores numéricos contienen un descriptor en texto, este descriptor puede ser asignado al nodo asociado al dato. De esta manera es posible adicionar referencias con información a la visualización del conjunto de datos y contar con elementos que faciliten la interpretación de los mapas dentro un contexto específico.

En general, la calidad de un mapa debe ser evaluada por un experto en el área de aplicación [48]. Además existen formas analíticas de medir el desempeño del mapa, ejemplos de estas medidas son los mencionados: error de cuantización y distorsión normalizada. Una forma general de una medida de error de cuantización puede ser encontrada en [90]. Otra medida es la propuesta en [79] la cual combina la idea del error de cuantización con la función vecindad, de esta

manera lo que se obtiene es un promedio pesado de las distancias cuadradas entre los elementos de una vecindad, cuando el radio de la función vecindad es cero, esta medida es equivalente a la de error de cuantización.

4.5. Análisis Inteligente de Información Científica

Hace unos cinco años, un grupo de investigadores del Laboratorio de Dinámica no Lineal de la Universidad Nacional Autónoma de México y del Instituto Finlay de Cuba, se han avocado a explorar las bases digitales de información biomédica aprovechando las nuevas tecnologías para el análisis inteligente de datos y el descubrimiento de conocimiento, salvando así las limitaciones que tiene la aplicación de otros métodos tradicionales a tan grandes volúmenes de datos.

Durante este lapso se han desarrollando toda una métodología que ocupa diversos sistemas de software y se ha ido aplicando experimentalmente en la realización de investigación [88] cienciométrica. La **cientometría** es una disiplina que involucra el análisis de las bases de datos de publicaciones científicas. Una de las principales herramientas utilizadas en el análisis de las bases de datos es el Viscovery SOMine. La utilización del SOM para el análisis de bases de datos de documentos es una de las nuevas aplicaciones en las cuales ha sido implementado el algoritmo [56]. En lo que sigue se ilustrará un ejemplo, desarrollado por el grupo antes mencionado, en donde se busca determinar la relevancia de diferentes sustancias químicas en las investigaciones sobre la tuberculosis.

La investigación de vacunas contra la tuberculosis se ha vuelto un problema de gran actualidad ya que se trata de una enfermedad re-emergente para la cual no se cuenta aún con alguna vacuna suficientemente efectiva. De acuerdo a la *Organización Mundial de la Salud* (OPS) esto constituye una situación de emergencia para nuestra orbe.

A los especialistas, que trabajan en vacunas contra la tuberculosis, les interesa conocer la forma en que está evolucionando el uso de diferentes sustancias en este tipo de investigaciones, a nivel mundial. Haciendo uso de las técnicas de la Minería de datos nuestro grupo analizó 2987 artículos de investigación contenidos en las bases de datos de MedLine (literatura biomédica) e investigó el uso de 8,961 diferentes sustancias que aparecen reportadas en las investigaciones de un lapso de 22 años (1980-2002).

Se observó que no todas tenían la misma incidencia en los 22 años de análisis y el estudio de frecuencias de ocurrencia reveló que, en la década de los 80´s las sustancias en las que se trabajaba más eran los agentes antineoplásticos (45 investigaciones) y la Ciclofosfamida (32 investigaciones), mientras que en la década de los años 90´s otras sustancias, como los adyuvantes inmunológicos, los interferones y los antígenos pasaron a ocupar los primeros lugares (208, 116 y 106 respectivamente). Asimismo se concluyó que actualmente (período 2000-2002), los adyuvantes (128) y los interferones (102) se siguen utilizando pero también se observa una emergente tendencia a la investigación de vacunas sintéticas y de DNA.

Dado el interés de los investigadores asociados con nuestro grupo, posteriormente especializamos la investigación a una familia de sustancias que tienen efecto en la modificación de la respuesta inmuno-biológica: las interleucinas (Interleukins). El propósito de este estudio fue, primero, identificar los diferentes tipos de Interleucinas, que son considerados en las investigaciones sobre vacunas contra la tuberculosis y después estudiar la evolución que ha tenido su utilización durante el período de análisis (1980-2002). Se identificaron las sustancias Interleukin-1, Interleukin-2, Interleukin-4, Interleukin-6 y Interleukin-12 en un conjunto de 2,600 sustancias (1600 resultados de investigación).

Entrenando una *red neuronal* (usando el sistema de software Viscovery SOMine) se generó mapas específicos para representar las sustancias relacionadas con la Interleukina-1 y la Interleukina-12. A continuación (Figura 40 y 41) se desplegaron los mapas correspondientes a los períodos (1990-1999) y (2000-2002). A pesar de que la interleukina-1 apareció con una frecuencia mayor que la Interleukina-12, en el período 2000-2002, los mapas producidos (figura 40) exhiben claramente que esta última sustancias aparece asociada a un número considerablemente mayor de sustancias.

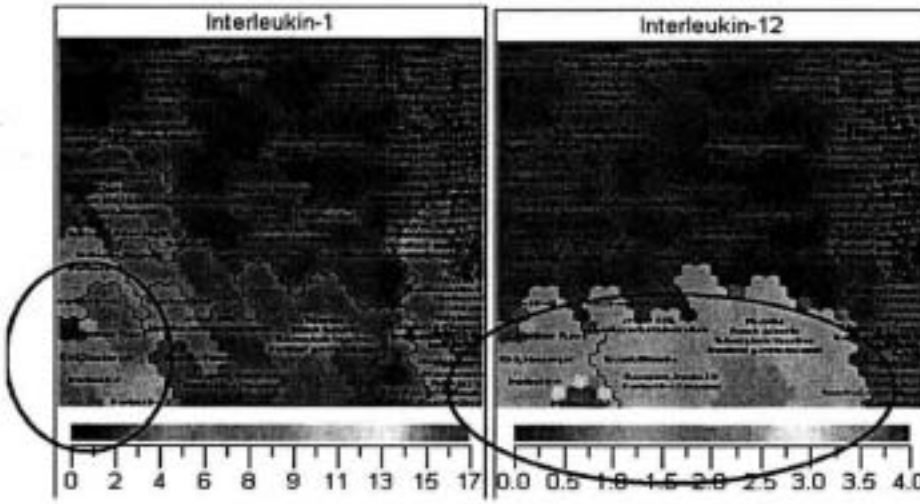

Figura 40: Representación del uso en la investigación de las sustancias Interleukin-1 e Interleukin-12 para el periodo 2000-2002.

Por otra parte, el análisis retrospectivo del lapso (1990-1999) mostró que la aparición de la Interleukina-1 predominaba respecto a la Interleukina-12, dado que esta última se asociaba con muy pocas sustancias en este período. Compárese los resultados de la figura 40 con los obtenidos en la figura 41. (2000-2002).

Lo mostrado son solo dos ejemplos de las análisis que se pueden hacer basado en el principio de la Minería de datos y textos. Estos son validos para otros

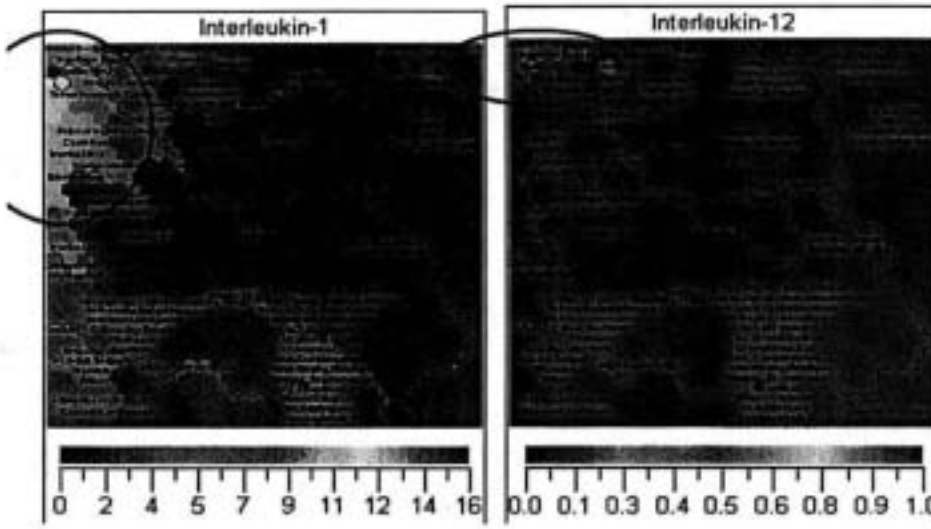

Figura 41: Representación del uso en la investigación de las sustancias Interleukin-1 e Interleukin-12 para el periodo 1990-1999.

campos del conocimiento, solo se necesita identificarel problema y aplicar el modelo correspondiente.

5. Concluciones

En este trabajo se abordó la problemática del *Análisis Inteligente de Datos* desde la perspectiva los métodos matemáticos –algoritmos- por medio de los cuales se realizan tareas útiles para la exploración y el descubrimiento de conocimiento en grandes conjuntos de datos.

Como se observo en la sección 2.3, las técnicas clásicas del *Análisis Exploratorio de Datos* tienen la gran limitante de poseer una alta complejidad computacional y por lo tanto su aplicación en el análisis de grandes conjuntos de datos es poco viable.

La alternativa que se propuso para resolver esta problemática es la utilización de *Redes Neuronales Artificiales*, en particular el algoritmo SOM que fue expuesto en el capítulo 3. Dadas las propiedades expuestas de este modelo, se puede concluir que efectivamente representa una herramienta de gran utilidad en el análisis de grande conjuntos de datos multidimensionales. Una de las propiedades más útiles del SOM es que brinda una forma de visualizar las *relaciones de similitud subyacentes* en el conjunto de datos. Además, se observó que el SOM es una alternativa viable para la realización de las tareas de clustering, proyección y cuantización. En este sentido el SOM es una herramienta que por si sola aproxima una solución a varios de los problemas clásicos en el *Análisis Exploratorio de Datos*. La versatilidad del SOM invita a pensar que muy pronto, el uso de esta herramienta será más común y se presentará en una gran variedad de campos de aplicación.

Sin embargo, aún hay muchas preguntas, por hacerse y por responder, con relación a resultados teóricos que ayuden reconocer y entender propiedades generales del algoritmo. En este trabajo los aspectos teóricos del algoritmo fueron tratados de una manera muy breve; sin embargo, es evidente que un análisis matemático de las propiedades del SOM, requiere la utilización de diversas ramas de las matemáticas y por lo tanto representa un reto muy interesante para la investigación matemática.

Como se pudo apreciar, el poder del SOM es tal que, la ausencia de resultados teóricos no ha impedido que el algoritmo sea considerado por importantes empresas de desarrollo de software e incluido en una buena cantidad de sistemas para la minería de datos y la visualización de información.

En el ejemplo de aplicación al análisis de información científica se mostró como el SOM puede ser utilizado con fines de investigación académica y como una herramienta para el análisis de la actividad científica y tecnológica.

En conclusión, en esta tesis se presentó una pequeña muestra del gran potencial que tienen las *Redes Neuronales Artificiales* en el *Análisis Inteligente de Datos*. La adecuada explotación de este potencial puede permitir, a especialistas en diversos campos, interactuar de manera muy cercana con la información almacenada en sus bases de datos y por medio de esta interacción descubrir conocimiento. ¿A caso ésta no es una nueva forma de hacer ciencia?.

★

Referencias

[1] Afifi, Clark, "Computer-Aided Multivariate Analysis", Van Nostrand Reinhold Company, Newyork, 1990.

[2] W. F. Allaman, "Apprentices of Wonder: Inside the Neural Networks Revolution", Bantam Book, 1989.

[3] Andrews D. F., "Plots of high-dimensional data", Biometrics 28, pp. 125-136, 1972.

[4] Barlow H. B., Ünsupervised learning", Neural Computation, 1:151-160, 1989.

[5] Bauer H. U., K. R. Pawelzik, "Quantifying the Neighborhood Preservation of Self-Organizing Feature Maps", IEEE Transactions on Neural Networks, vol. 3, No. 4. 1992.

[6] Bertalanffy L.V., "Teoría General de Sistemas", Fondo de Cultura Económica, México, 1976.

[7] Berson A., Smitht S. J., "Data Warehousing, Data Mining and OLAP", McGraw Hill, 1997.

[8] Berthold M., Hand D.J., "Intelligent Data Analysis", Springer, 2000.

[9] ver en [9] pp. 249.

[10] Bertin J., "Graphics and Graphic Information", Walter de Gruyter, Berlin, 1977.

[11] Bigus J., "Data Mining with neural networks", Mc GrawHill, USA, 1996.

[12] Boden M.A., "The Philosophy of Artificial Intelligence", Oxford University Press, 1990.

[13] Born, R., .Artificial Intelligence: the case against", Croom Helm, 1987.

[14] Bouton C., Pagès G., "Self-organization and the a.s. convergence of the one-dimensional Kohonen algorithm with non-uniformly distributed stimuli", Appl. Stochastic Process, 47, 1993.

[15] Brachman R. , Anand R., "The Process of Knowledge Discovery in Databases: A human centered approach", Advantedge in Knowledge Discovery in Databases Magazine, MIT Press, 1996.

[16] Carbonell J. G., "Machine learning: Paradigms and Methods", Special Issues of Artificial Intelligence: An International Journal, 1990.

[17] Chen C., "Information Visualization", Information Visualization, vol. 1, 2002.

[18] Chernoff H., "The use of faces to represent points in k-dimensional space graphically", Journal of the American Statistical Association 63, pp. 361-368, 1973.

[19] Cottrell M., Pagès J. C., "Etude d'un algorithme d'auto-organisation" Ann. Inst. Henri Poncaré 23 (1), (1987).

[20] Cottrell M., Fort J.C. , Pagès G., In Proc. WSOM'97, Workshop on Self-Organizing Maps, Helsinski University of Technology, Neural Networks Research Centre, Espoo, Finland 1997.

[21] Cottrell M., E. de Bodt, Letremy P., Verleysen M., "On the use of self-organizing maps to accelerate vector quantization", Neurocomputing, aceptado en Septiembre de 2003.

[22] Cottrell M., Fort J.C. ,Pagès G., "Theorical aspects of the SOM algorithm", Neurocomputing 21, 1998.

[23] Dittenbahc M., Rauber A., D. Merkl, "Uncovering hierarchical structure in data using the growing hierarchical self-organizing map", Neurocomputing 48 (2002), 199-216.

[24] Endo M.,Uendo M.,Tanabe T., "A Clustering Meted Using Hierarchical Self-Organizing Maps", Journal of VLSI Signal Processing 32, 105-118, 2002.

[25] Erwin E., Obermayer K., Shulten K., "Self-organizing maps: stationary states, metastability and convergence rate", Biological Cybernetics 67, 1992.

[26] Erwin E., Obermayer K., Shulten K., "Self-organizing maps: ordering, convergence properties and energy functions", Biological Cybernetics 67, 1992.

[27] Eudaptics, "Viscovery SOMine: User Manual, ver. 4.0", 2001.

[28] Fayyad U., Piatetsky-Shapiro G., Smyth P., "Knowledge Discovery and Data Mining: Towards a Unifying Framework", Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Oregon, USA, 1996.

[29] Fayyad U., Piatetsky-Shapiro G., Smyth P., "From Data Mining to Knowledge Discovery: An Overview", Advantedge in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996.

[30] Flanagan A., "Self-organization in Kohonen's SOM", Neural Networks 9, 1996.

[31] Flanagan A., "Sufficient conditions for self-organization in the one-dimensional SOM with a reduced width neighbourhood", Neurocomputing 21, 1998.

[32] Flanagan A., "Self-organization in the one-dimensional SOM with a decreasing neighborhood", Neural Networks 14, 2001.

[33] Flexter A., "On the use of Self-organizing Maps for Clustering and Visualization", The Austrian Institute of Artificial Intelligence, 1999.

[34] Frawley W. J., Piatetsky-Shapiro G., Matheus C. J., "Knowledge Discovery in Databases: An Overview", AI Magazine, 1992.

[35] Fort J.C., Pagès G., "On the a.s. convergence of the Kohonen algorithm with a general neighborhood function", Ann. Appl. Probab. 28 (4), 1996.

[36] Fort J.C., Pagès G., "Convergence of stochastic algorithms" Add. Appl. Probab. 28 (4), 1996.

[37] Grabmeier J., Rudolph A., "Techniques of Cluster Algorithms in Data Mining", Data Mining and Knowledge Discovery, vol. 6, 2002.

[38] Gomez A., Moreno A., Pazos J., Sierra-Alonso A., "Knowledge maps: An essential technique for conceptualisation", Data & Knowledge Engineering 33, 2000.

[39] Hastie T., Stuetzle W. "Principal Curves", Journal of American Statistical Association 8, pp.. 502-516, 1989.

[40] Heskes T., "Energy functions for self-organizing maps", Theorical Foundation SNN, University of Nijmegen, 1999.

[41] Heskes T., "Self-organizing maps, vector quantization, and mixture modeling", IEEE Transactions on Neural Networks, vol. XX no. Y, 2001.

[42] Hibera J.R., Martinez V.J., "Redes Neuronales Artificiales", AlfaOmega, 2000.

[43] Hopfield J.J., "Neural networks and physical systems with emergent properties", Proceedingns of the National Academy of Sciences 79, pp. 2554-2558, 1982.

[44] Horowitz R., Alvarez L., "Convergence Properties of Self-organizing Neural Networks", Dep. of Mechanical Engineering, University of California.

[45] Hotelling, H., .^nalysis of a complex of statistical variables into principal components", Journal of Educational Psychology 24, pp. 417-441, 1933.

[46] Jain A. K., Mao J., Mohiuddin K., "Artificial Neural Networks: A Tutorial", IEEE Computer Special Issue on Neural Computing, 1996.

[47] Jain A.K., Dubes R.C., "Algoritms for Clustering Data", Prentice Hall, Englewood Cliffs, NJ., 1988.

[48] Kaski S., "Data Exploration Using Self-Organizing Maps", Ph. D. Thesis, Helsinki University of Technology, Finland, 1997.

[49] Killman R., "The Data Warehouse Toolkit", Jonh Wiley & Sons, Inc., 1996.

[50] Kohonen T. , "Self-organizad formation of topologically correct feature maps", Biological Cybernetics, 43, 1982.

[51] Kohonen T., "Analysis of a simple self-organizing process", Biological Cybernetics, 44, 1982.

[52] Kohonen T., "self-organization and associative memory", 3ra ed. Springer-Verlag, 1989.

[53] Kohonen T., Kangas J., "Developments and Applications of the Self-Organizing Maps and Related Algorithms", Mathematics and Computers in Simulation 5-6, 1996.

[54] Kohonen T., "The self-organizing map", Neurocomputing, 1998.

[55] Kohonen T., "Exploration of Very Large Databases by Self-Organizing Maps", Neural Networks Research Centre, Helsinski University of Technology, 1998.

[56] Kohonen T., Kaski S., Honkela J., Paatero V., Saarela A.,"Self Organization of Massive Document Collection", IEEE Transactions on Neural Networks, Vol. 11, No. 3, May 2000.

[57] Kohonen T., "Self-Organizing Maps", 3ra Edición, Springer-Verlag, 2001.

[58] ver en [57] capitulo 7.

[59] ver en [57] sección 2.15.

[60] ver en [57] pp.112.

[61] ver en [57] pp. 86.

[62] ver en [57] sección 3.6.

[63] Un listado de algunas de las publicaciones puede ser consultado en: http://www.cis.hut.fi/nnrc/refs/

[64] Kozen D. C., "The Desing and Análisis of Algotithms", Dexter C. Kozen, Springer-Verlag, 1992.

[65] Kreuseler M., López N., Shumann H., "A Scalable Framework for Information Visualization", University of Rostock, Dept. of Computer Science, Germany, 1999.

[66] Lampinen J., Kostianen T., "Self-Organizing Map in Data Análisis", Proc. ESANN'2000, Bruges, Belgium, 2000.

[67] Liu X., "Progeress in Intelligent Data Analysis", Aplied Intelligence 11, 235-240, Kluwer Academic Publishers, 1999.

[68] Mannila H., "Data Mining: Machine Learning, Statistics and Databases", Dep. of Computer Science, University of Helsinky.

[69] Martinetz T.M., Villmann T., Der R. , Herrmann M., "Toplogy Preservation in Self-Organizing Feature Maps: Exact Definition and Measurement", IEEE Trasactions on Neural Networks, vol. 8, no. 2, 1997.

[70] McCulloch, Warren S., Pitts, .ª Logical Calculus of the Ideas Immanent in Nervous Activity", Bulletin of Mathematical Biophysics 5, pp. 115-133, 1943 (publicado en [12] pp. 22-39).

[71] Minsky M., Papert S.A., "Perceptrons: An Introduction to Computational Geometry", MIT Press, edición expandida 1989.

[72] cita tomada de [2] pp. 105.

[73] Mitchell M., "Machine Learning", McGraw-Hill, 1997.

[74] cita tomada de [2] pp. 49.

[75] Nilsson, N. J., "Learning Machines: Fonundations of Trainable Pattern-Classification Systems", McGraw-Hill, New York, 1965.

[76] Polanco X., Francois C., Lamirel J. C., "Using arificial neural networks for mapping of science and technology: A multi-self-organizing-maps approach", Scientometrics, Vol. 51, No. 1 (2001).

[77] Provost F., Kohavi R., .°n Applied Research in Machine Learning", Machine Learning 30, 127-132, Kluwer Academic Publishers, 1998.

[78] Ripley B.D. ed., "Pattern Recognition and Neural Networks", Cambridge University Press, 1996.

[79] Ritter H., Shulten K., "Kohonen's self-organizing maps: exploring their computational capabilities", In Proceedings of the ICNN'88, IEEE International Conference on Neural Networks, volume I, pp. 109-116, 1988.

[80] R. Rojas, "Neural Networks: A Systematic Introduction", Springer, 1996.

[81] Rosemblat F., "The Perceptron: A Theory of Statistical Separability in Cognitive Systems", Principles of Nurodynamics, Spartan, 1962.

[82] Rumelhart D. E., McClelland J.L., "Parallel Distributed Processing: Exploration in the microstructure of cognition", MIT Press, 1986.

[83] Sadeghi A., "Asymptotic behavior of self-organizing maps with non-uniform stimuli distribution", Ann. Appl. Probab. 8 (1) 1998.

[84] Sadeghi A., "Self-Organizing property of Kohonen's map with general type of stimuli distribution" Neural Networks, aceptado en 1997.

[85] Samad T., Harp S. A., "Self-organization with partial data", Network: Computation in Neural Systems 3, 1992.

[86] Shweitzer F., "Self-Organization of Complex Structures: from individual to collective dynamics –some introductory remarks", en Self-Organization of Complex Structures: from individual to collective dynamics, ed. F. Shweitzer, Gordon and Breach Science Publishers, 1997.

[87] Sotolongo G, Guzmán M. V., "Aplicaciones de las redes neuronales. El caso de la bibliometría", Ciencias de la Información. 2001; 32(1):27-34.

[88] Sotolongo G., Guzmán M. V., Saavedra O., Carrillo H., "Mining Informetrics Data with Self-organizing Maps", in: M. Davis, C.S. Wilson, (Eds.), "Proceedings of the 8 th International Society for for Scientometrics and Informetrics", ISBN:0-7334-18201. Sydney, Australia July 16-20. Sydney: BIRG; 2001: 665-673.

[89] Sotolongo G., Guzmán M. V., Carrillo H., "ViBlioSOM: visualización de información bibliométrica mediante el mapeo autoorganizado", Revista Española de Documentación Científica, 2002, 25(4):477-484.

[90] Sun Y., "On quantization error of self-organizing map network", Neurocomputing 34, 2000.

[91] Tolat V.V., "An analysis of Kohonen´s self-organizing maps using a system of energy functions", Biological Cybernetics 64, 1990.

[92] Tukey J. W., "Exploratory data analisys", Addison Wesley, 1977.

[93] Turing A. M., .°n Computable Numbers, with an Application to the Entcheidungsproblem", Proc. London Mathematical Society 43, 1937.

[94] Turing A. M., Çomputing Machinary and Intelligence", Mind LIX 2236, Oct. 1950 (publicado en [12] pp. 40-66).

[95] Widrow B., Hoff M. E., .^daptive switching cicuits", 1960 IRE WESCON Convention Record, pp. 96-104, 1960.

[96] Young G., Householder A.S., "Discussion of a set of points in terms of their mutual distances", Psychometrika 3, pp. 19-22, 1938.