UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Data SOMinning: Software para el Descubrimiento de Conocimiento en Grandes Bases de Datos de Información Científico-Tecnológica

T E S I S

C O N J U N T A

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADA Y LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A N:

MARY CARMEN TREJO AVILA

JOSÉ GUSTAVO GONZÁLEZ ANGELES

FACULTAD DE CIENCIAS
UNAM

DIRECTOR DE TESIS:
DR. HUMBERTO ANDRÉS CARRILLO CALVET

CODIRECTOR DE TESIS:
DR. FELIPE LARA ROSANO

2006

Mary Carmen,

A mi familia:
Margarita, Gabriel, Gabriel Eduardo, Aidé y Papatín†

A mi compañero de vida:
Víctor.

Gustavo,

A mi familia, que juntos los cuatro, no existe nada que nos detenga.

Agradecimientos

A Dios, por la vida que me regaló.

A mi mamá y papá por creer siempre en mí demostrándomelo con apoyo y amor incondicional, a mi hermano por su tiempo y afecto, a mi hermana por su generosidad y cariño, a Papatín† por quererme y cuidarme tanto y a TiaMaría por todas sus atenciones. A todos ellos les debo la realización de este trabajo, representante de mis primeras metas, gracias a su comprensión y esfuerzo.

A Víctor por su sincero e inigualable amor y valiosos años compartidos, en los cuales he experimentado distintas sensaciones las cuales me han llevado a ser, en gran parte, lo que soy actualmente.

Al Dr. Humberto Carrillo y Dr. Felipe Lara por el apoyo ofrecido para el inicio y término de este trabajo.

A la Mtra. Nieves por ser un pilar principal en este trabajo, demostrando ser una gran compañera otorgando su conocimiento y tiempo de manera desinteresada al presente.

A mis sinodales Guadalupe Ibargüengoitia y Javier García por sus oportunas aportaciones.

A mis amigas María Fernanda, Pamela, Itzel y Columba por ser mis cómplices brindándome una amistad franca, viviendo experiencias importantes, irrepetibles e incomparables, tal como lo es el objetivo de este trabajo.

A mis cuates computólogos Grecia, Josafat, John, Gustavo, Ruy y Paco, por todos aquellos empujones brindados durante la carrera, tanto académicamente como para el entretenimiento, aquel que sólo nosotros sabemos disfrutar. Así como también a mis cuates universitarios Rosendo, Jacob y Blas con los que he compartido distintas etapas de mi vida escolar.

A todos los que han formado parte del grupo del Laboratorio de Dinámica No Lineal durante mi estancia y que de alguna forma han contribuido en este trabajo. Especialmente a Elio Villaseñor, Luis Nava, María Victoria Guzmán y José Luis Jiménez.

Agradecimientos

Gracias papá por todo tu esfuerzo y sacrificio que hiciste para que yo estuviera aquí el día de hoy. Nunca lo olvidé y nunca olvidaré todo lo que has hecho por mí; gracias por hacer que quiera ser como tú algún día. Gracias por existir.

Mamá, no tengo palabras para hacerte saber lo feliz que soy por tenerte a mi lado. Sin tí no estaría aquí, sin tí no estaría completo, sin tí las cosas bellas de la vida no tendrían sentido. Gracias por estar a mi lado.

Marco, gracias por jugar, llorar y reir cada minuto de tu vida conmigo. Gracias por cuidarme en cada momento, por apoyarme sin condición, por ser mi ejemplo, mi amigo, mi hermano. Recuerda: tú y yo siempre. Te quiero.

Gracias Laura por darle sentido de nuevo a mi camino, por llegar a mí en uno de los momentos más difíciles de mi vida y por permanecer hasta hoy a mi lado a pesar de todo. Tú me has enseñado a disfrutar cada minuto de mi vida y recordar aquellas cosas pequeñas que hacen de este mundo un lugar mejor para vivir. Gracias por ser la razón de querer ser un mejor ser humano cada mañana. 125.

Mary, gracias por ser más que una amiga, por abrirme tu corazón y darme tu confianza. Gracias por ayudarme a saber de lo que soy capaz, por decirme mis errores y aceptarme en tu vida con ellos. No tengo palabras para describir lo que significas para mí. Te quiero.

Gracias a Manuel, Beatriz, Emmanuel y Esaú por apoyarme incondicionalmente, por ayudarme encontrar mis debilidades y mis aptitudes. Gracias por abrirme las puertas de sus corazones y hacerme sentir parte de todos ustedes.

Mins, Josafat, Ruy, John, Paco, Grecia, simplemente sin ustedes estaría perdido. Han sido un ejemplo para mí, y más aún, han sido, son y serán los mejores amigos que tendré.

Humberto, te agradezco la confianza que pusiste en mí. Gracias por darme la oportunidad de demostrar que el valor de una persona se demuestra con honestidad y trabajo. Gracias por el apoyo.

Gracias Nieves por la ayuda durante todo este tiempo, por ser una amiga y una gran persona.

Gracias Dr. Felipe Lara por su apoyo y el tiempo dedicado a este trabajo.

Al profesor Oscar Falcón (qepd), que creyó en mí y siempre lo recordaré como mi mejor

profesor en la carrera.

Gracias a mi institución, mi segundo hogar, mi universidad que me ha enseñado el significado de ser parte de esta institución: coraje, honradez, verdad y sobre todo honestidad.

Debo agradecer a todas y cada una de las personas que he conocido a lo largo de mi vida, ustedes también son parte de esto.

Índice general

Índice de figuras

Índice de tablas

Presentación

El vertiginoso desarrollo de nuevas tecnologías en el terreno de la informática, nos da acceso a grandes volúmenes de datos en cualquier tema y con relativa facilidad. Aunque en primera instancia parecería que al disponer de más datos, automáticamente disponemos de más información útil, esto es verdad hasta cierto punto, ya que los datos por sí mismos no nos aportan conocimiento que es justamente lo que queremos extraer de ellos. Nos enfrentamos entonces al problema de traducir esta gran cantidad de información en conocimiento que nos sea útil.

Antecedentes.

La revolución tecnológica que surgió durante el siglo pasado, trajo consigo resultados sorprendentes y consecuencias inevitables. Esta revolución fue dirigida por la *International Bussines Machines (IBM)*, que en 1952 produce la primera computadora diseñada para cálculos de índole científica, la IBM 701. Con el paso de las décadas, IBM consideró el hecho del crecimiento progresivo y constante de datos almacenados por las empresas e instituciones; en 1983 introduce al mercado la IBM 3330 Data Storage, un dispositivo con capacidad de almacenar 200 millones de bytes expandibles hasta 1.6 billones de bytes. Sin embargo, los datos contenidos en estos grandes y avanzados dispositivos, no respresentaban por si mismos herramientas que permitieran llevar a cabo procesos y/o decisiones benéficos para si.

En los años ochenta, la *National Agency of Space Administration (NASA)* requería de crear una clasificación por temas, de toda la información que en ese entonces poseía. Tal clasificación debía ser producto de un análisis de información automatizado, de tal manera que los resultados fuesen obtenidos en el menor tiempo posible, además de tener un grado de veracidad aceptable; para ese entonces, se habían observado los problemas que el Aprendizaje de Máquina (Machine Learning, ML) presentaba para el análisis en bases de datos. En respuesta surge el concepto de Descubrimiento de Conocimiento en Bases de Datos (Knowledge Discovery in Databases, KDD) como alternativa del análisis de bases de datos a gran escala, tomando como antecendente directo a ML e incorporándolo como parte de este proceso.

Era de suma importancia que estos nuevos métodos de análisis de la información no fueran limitados por la cantidad de datos a analizar, y debían ser aptos para ser utilizados en un futuro pues, tan sólo en esa década, la NASA pronosticaba generar un terabyte de datos diariamente.

A principios de los años noventa, se habían dearrollado algunas herramientas de software y

distintos dispositivos de almacenamiento que ayudaban a este análisis, sin embargo, éstas funcionaban de forma aislada y eran incompatibles entre sí. Algunas implementaban nuevas técnicas de análisis como aprendizaje inductivo, algoritmos genéticos, redes neuronales, estadística bayesiana, sistemas expertos e incluso conceptos de teoría de la información.

Con el paso de los años eran más y más las compañías e instituciones que requerían de estos recursos, y al mismo tiempo la cantidad de datos que se generaban, incrementaban día con día. Este es el caso de la *Biblioteca Nacional de Medicina de los Estados Unidos (National Library of Medicine, NLM)* de los Estados Unidos, pionera en el uso de la emergente tecnología en computación para la consulta de material bibliográfico impreso. En 1971, la NLM crea su primer base de datos: *MedLine*; oficialmente MedLine contiene registros de publicaciones desde 1966 a la fecha. En 1966 contenía 239 publicaciones, para 1985 contaba ya con 300,000 títulos.

En 1998 *Winter Corporation*, compañía que provee servicios de consultoría en sistemas de bases de datos, calculaba que para el año 2001 el promedio de tamaño de las bases de datos de los grandes consorcios e instituciones sería de 10 terabytes. En 1999 MedLine reportó 10 millones de títulos de publicaciones; es decir que el crecimiento es de más de 1,000 artículos diariamente; actualmente MedLine realiza actualizaciones de sus títulos mensualmente.

Debido al tamaño actual de las bases de datos, el proceso KDD representa un reto no trivial, pero con resultados que pueden tener un alto impacto en nuestro entorno; las estadísticas indican que *France Telecomm* posee la base de datos más grande con 29.2 terabytes, después se tiene a *AT&T* con una base de datos de 26.2 terabytes. Ante estas circunstancias, los requerimientos tecnológicos para un proceso como KDD, implica el uso de sistemas de cómputo de alto desempeño como procesamiento paralelo y distribuido, además de sistemas de software eficientes que integren todas las herramientas necesarias para llevar a cabo todas y cada una de las fases de las que se compone el proceso KDD.

Actualmente estos sistemas de software presentan serias limitaciones a considerar: por un lado estos sistemas sólo ofrecen soporte parcial para algunas fases del proceso como selección de datos, limpieza, transformación o minería de datos y en muchas ocasiones es necesario llevar a cabo cada fase con distinto software. Algunos sistemas de este tipo son Procite, Excel, SNNS entre otros. Por otro lado, los sistemas integrales para el proceso en su mayoría son comerciales, y los costos de sus licencias son excesivos; Viscovery® SOMine®, Clementine, DBMiner y Ghost Miner son suites integrales de este tipo.

Objetivo.

El objetivo principal de este trabajo es diseñar y desarrollar una suite integral, es decir un sistema de software que contenga todas las funcionalidades necesarias para el proceso de Descubrimiento de Conocimiento en Bases de Datos mediante el uso de la red neuronal SOM y diversas técnicas de conglomeración (clustering) y visualización. La suite cumple con el estándar de calidad ISO/IEC 9126-1. [1] Esta versión de la suite será diseñada para la investigación cien-

[1] ISO/IEC 9126-1: Organización Internacional para la Normalización o Estandarización/Comisión Electrotécnica

ciométrica, basada en el análisis bibliométrico de textos científicos.

Organización del documento.

El presente trabajo se encuentra dividido en cinco capítulos.

Los tres primeros capítulos aportan los elementos teóricos necesarios para entender el sistema. En ellos se exponen conceptos generales y definiciones de elementos que serán de utilidad a lo largo del trabajo. Los capítulos cuatro y cinco describen el primer ciclo del proceso de desarrollo de la suite. Posteriormente se presentan las conclusiones generales.

Presentación. Se ofrece una introducción, sus objetivos y antecedentes de lo que es este trabajo.

Capítulo 1. *Descubrimiento de Conocimiento en Bases de Datos.* Se definen los conceptos fundamentales de KDD, minería de datos y redes neuronales. Estos conceptos nos introducen a las ideas fundamentales sobre las cuales se basa el desarrollo de la suite.

Capítulo 2. *Redes Neuronales y el algoritmo SOM.* Este capítulo comprende la especificación de la red neuronal de Teuvo Kohonen: Self-Organizing Map (SOM), su integración como técnica de minería de datos y su visualización. Además se presenta el análisis realizado a distintas herramientas de software para redes neuronales.

Capítulo 3. *Metodología ViBlioSOM.* Se definen los conceptos de Bibliometría, Patentometría, Cienciometría e Informetría. Se presenta la metodología ViBlioSOM®, la cual permite la visualización de información bibliométrica mediante el mapeo auto-organizado.

Capítulo 4. *Diseño de una suite para la Minería de Datos: Data SOMinning.* Se exponen los requerimientos así como el diseño de la suite mediante el uso de diagramas (paquetes, caso de uso y secuencia).

Capítulo 5. *Data SOMinning y su aplicación en la investigación científica.* Se presenta un caso de estudio utilizando la suite Data SOMinning.

Conclusiones. Se presenta una reflexión final acerca del presente trabajo, mostrando el impacto y beneficios de la construcción de una herramienta con los alcances de Data SOMinning.

Internacional, referente a Ingeniería de Software, calidad de producto y modelos de calidad.

Capítulo 1

Descubrimiento de Conocimiento en Bases de Datos

Como parte del avance tecnológico de los últimos años, el acopio de datos y generación de información son parte de las tareas principales de los usuarios de sistemas de cómputo. Sin embargo, estas tareas llevan consigo consecuencias importantes: el rápido crecimiento de la información y por ende, el sobrealmacenamiento de ésta. Este conjunto de datos es almacenado en avanzados dispositivos de hardware y administrado por sistemas de software cada vez más eficientes y cada vez más costosos.

Las **Bases de Datos** son estructuras organizadas de tal forma que facilita el almacenamiento eficiente, la consulta y modificación de los datos almacenados; durante años han ayudado a enfrentar este crecimiento y sobrealmacenamiento de información, mejorando cada vez más los sistemas manejadores de bases de datos. Hoy en día es más común observar bases de datos de giga o terabytes de tamaño en distintos campos de desarrollo: instituciones educativas, institutos de investigación, empresas privadas, instituciones de gobierno, etc.

El volumen de estos almacenes de datos hacen inoperante llevar a cabo un análisis con técnicas tradicionales de la estadística, que representen beneficios potenciales para el desarrollo humano, académico, científico, tecnológico o económico.

> "Es por esto la necesidad de la generación de nuevas técnicas computacionales y herramientas que asistan a los humanos en la extracción de información útil (conocimiento) de grandes volúmenes de datos." *(U. Fayyad et al., 1996.)* [FPSS96a]

Como respuesta a esta necesidad surge el campo del **Descubrimiento de Conocimiento en Bases de Datos (Knowledge Discovery in Databases, KDD)**. Este proceso es considerado como el desarrollo que por medio de varias etapas, los datos analizados toman cierto sentido. KDD combina técnicas de múltiples campos de aplicación como inteligencia artificial, estadística, diversas técnicas de visualización y como ya mencionamos, bases de datos.

1.1. Definiciones básicas.

Como su nombre lo indica, el resultado del proceso KDD es consecuencia de llevar a cabo una serie de pasos, cada uno con tareas específicas. Podemos definir el proceso KDD como:

> "el proceso no trivial de identificar patrones válidos, nuevos, potencialmente útiles y entendibles, dentro de los datos" *(U. Fayyad et al., 1996.)* [FPSS96a]

Entendemos por **datos** al conjunto de símbolos que utilizamos para expresar o representar un valor numérico, un hecho, un objeto o una idea en la forma adecuada para su procesamiento, que bien pueden ser información almacenada en una base de datos. **Patrones** se refiere a ajustar un modelo, encontrar estructuras, o en general, hallar alguna descripción representativa de este conjunto de datos. Se dice que el proceso es no trivial debido a que el cómputo involucrado no es simple y por lo tanto requiere de aplicar técnicas computacionales de alto desempeño (por ejemplo cómputo distribuido).

Hemos mencionado la importancia de la búsqueda de patrones que formen un modelo de representación de los datos, es por esto que una de las etapas más importantes dentro del proceso es la de minería de datos. La **Minería de Datos** (Data Mining, DM) es la fase que por medio de distintas herramientas y métodos nos permite obtener de forma automática, tales patrones o información útil que una vez validada se acepta como conocimiento. Entre estas herramientas y métodos de la minería de datos, se encuentran métodos como Análisis Exploratorio de Datos (Exploratory Data Analysis, EDA) y una relación muy estrecha con la Inteligencia Artificial (Artificial Intelligence, AI) en el campo de ML.

Sin embargo, en esta etapa nos encontramos con diversos problemas; en muchas ocasiones la estructura en que los datos son representados no es la adecuada, o el conjunto de los patrones es demasiado grande, por lo que el número de variables a considerar es multidimensional. Lo anterior impide que estas técnicas y métodos para la minería de datos, sean los óptimos para llevar a cabo un análisis eficiente. Es por esto la importancia de investigar, desarrollar e implementar nuevas técnicas y métodos, que se lleven a cabo de manera que el tiempo de ejecución sea corto y los algoritmos más eficientes con resultados óptimos.

Teniendo como antecedente a ML como técnica de minería de datos, la inteligencia artificial aporta una nueva opción para esta etapa: las Redes Neuronales Artificiales.

> "Inspirados en la anatomía y fisiología del cerebro humano, las Redes Neuronales Artificiales son modelos matemáticos que permiten hacer computación inteligente." *(Wright, 1998.)* [WP98]

En una red neuronal, el procesamiento de la información se realiza por medio de un desarrollo distribuido y el cómputo se realiza en forma paralela. Desde el punto de vista de la minería de datos el procesamiento paralelo y distribuido, permite que las redes neuronales lleven a cabo el procesamiento de datos a una escala masiva.

La red neuronal de entrenamiento no supervisado **Self-Organizing Map (SOM)** diseñada por Teuvo Kohonen, resulta ser un eficiente algoritmo que permite la proyección de datos mul-

tidimensionales a una malla o retícula bidimensional denominada *mapa*, preservando la organización topológica del conjunto de datos original. En el capítulo 2 hablaremos con precisión de esta red neuronal.

1.2. Fases del proceso KDD.

Existen distintas versiones acerca de cuáles son las fases que envuelve el proceso KDD, cada versión toma en cuenta más o menos fases dentro del proceso, sin embargo la mayoría reconoce tres etapas:

1. Preprocesamiento y preparación de los datos.
2. Búsqueda de patrones o modelos.
3. Evaluación del conocimiento.

1.2.1. Preprocesamiento y preparación de los datos.

Generalmente los datos de inicio utilizados en el proceso KDD no son adecuados para ser usados en la etapa de búsqueda de patrones. Esta etapa consiste en la aplicación de técnicas para obtener los datos de modo que puedan ser procesados posteriormente.

1.2.1.1. Entendimiento del dominio de aplicación.

Al iniciar el proceso es importante llevar a cabo la selección de un conjunto apropiado de datos, la cual dependerá de los objetivos que se deseen alcanzar.

Para definir estos objetivos es necesario desarrollar un completo entendimiento del dominio en el campo de aplicación y contar con conocimiento relevante previo del conjunto de datos; para esto es ideal contar con expertos que estén asociados directamente con el campo de aplicación, pues su interpretación será fundamental para tomar decisiones y aplicar criterios adecuados para determinar patrones que lleguen a ser de importancia para llegar al objetivo.

1.2.1.2. Creación del conjunto de datos objetivo.

Una vez seleccionados los datos adecuados, se procede a realizar una nueva selección de un subconjunto de éstos con los cuales realizar el análisis y descubrimiento. Este subconjunto pueden ser variables de las muestras o un subconjunto de ejemplos que sirvan como base para el análisis.

1.2.1.3. Limpieza y preprocesamiento de datos.

Esta fase puede tomar más del 80 % del tiempo total del proceso. Consiste en una serie pasos sistemáticos los cuales, permiten obtener datos confiables.

Debemos considerar que en la realidad, el conjunto inicial de datos presenta diversos tipos de *ruido* como inconsistencia entre los datos, redundancia o duplicidad. Para solucionar este problema es necesario establecer una *normalización* de los campos que contiene cada dato por medio de criterios para la selección de atributos que sean de importancia para el proceso. Algunos de estos criterios de normalización serían establecer tipos de datos a considerar (enteros, racionales, continuos, discretos o intervalos de valores), datos que dependan del tiempo (series), etc.

La calidad de los resultados obtenidos al final del proceso, dependerán en su totalidad de la calidad de los datos obtenidos del preprocesamiento.

1.2.1.4. Reducción y proyección de datos o transformaciones.

La aplicación de los métodos de reducción y proyección dependerá si el tipo de datos lo permite, por ejemplo los datos pueden estar respresentados en texto plano, algunos pueden incluir series de tiempo, pueden ser imágenes o bien pueden ser datos estructurados. Si la representación de los datos es la adecuada, sí es posible utilizar métodos de reducción de dimensión o transformación para reducir el número de variables a considerar y encontrar características útiles para la representación de los datos; algunos métodos son: el Análisis de Componentes Principales (Principal Component Analysis), el análisis de factores o el escalado multidimensional.

1.2.2. Búsqueda de patrones o modelos.

Mediante una técnica de minería de datos, se obtiene un modelo de conocimiento, que representa patrones de comportamiento observados en los valores de las variables del problema o relaciones de asociación entre dichas variables. También pueden usarse varias técnicas a la vez para generar distintos modelos, aunque generalmente cada técnica obliga a un preprocesado diferente de los datos.

1.2.2.1. Seleccionar el método de Minería de Datos.

Uno de los aspectos más sutiles del arte de llevar a la práctica el proceso KDD, es el relacionar los métodos de minería de datos con problemas del mundo real. Este proceso puede ser a menudo no trivial, ya que los problemas tienden a ser relativamente complejos y además incluyen algunos detalles que no son necesariamente relevantes para el proceso de descubrimiento, mientras que las definiciones de los métodos para resolver estos problemas tienden a ser abstractos y complejos.

Los métodos para minería de datos pueden ser clasificados de acuerdo a las tareas específicas que desempeñan:

1. Métodos descriptivos o no supervisados: descubren patrones que permiten describir la manera en que los datos se distribuyen o agrupan en el espacio.

2. Métodos predictivos o supervisados: pronostican el valor de una variable, desde valores conocidos de otras variables.

Las metas de estas tareas se pueden generar a partir de alguna de las siguientes técnicas:

- **Clasificación**: determina una función que mapea un dato dentro de una o varias clases predefinidas.
- **Regresión**: determina una función que representa el comportamiento de alguna porción del conjunto de datos o el descubrimiento de relaciones funcionales entre variables.
- **Conglomeración (Clustering)**: identifica un conjunto finito de categorías o clases que describen los datos. A diferencia del método de clasificación, este no depende de clases predeterminadas.
- **Sumarización**: encuentra una descripción que representa el comportamiento de alguna porción del conjunto de datos o el descubrimiento de relaciones funcionales entre variables.
- **Modelación de Dependencia**: encuentra un modelo que describa dependencias significativas entre las variables.
- **Cambio y detección de desviación**: descubre los cambios más significativos en los datos a partir de medidas previas o valores normativos.

Dependiendo de la naturaleza de los datos, así como del objetivo planteado para una aplicación que utilice minería de datos, variarán las tareas a ser ejecutadas.

1.2.2.2. Seleccionar el algoritmo de Minería de Datos.

Para lograr la realización exitosa de las tareas presentadas anteriormente es importante escoger los algoritmos de minería de datos de manera correcta.

Un algoritmo de minería de datos es un procedimiento bien definido que toma datos como entrada y produce una salida en forma de modelo o patrón. Usamos el término bien definido indicando que el procedimiento se debe plantear como un conjunto finito de reglas. Para que sea considerado un algoritmo, el procedimiento debe terminar siempre después de un número finito de pasos y producir un resultado.

La mayoría de los algoritmos para la minería de datos pueden ser vistos como composiciones de algunas técnicas y principios básicos. Existen una gran variedad y un número amplio de algoritmos, estos se encuentran en diversas disciplinas como la estadística, reconocimiento de patrones, aprendizaje de máquina y bases de datos y se aplican dependiendo de las características del problema a resolver. Estos algoritmos consisten, en gran parte, de tres componentes principales, los cuales se listan a continuación:

1. Modelo de representación. Es el lenguaje o modelo matemático usado para describir los patrones a ser considerados por el algoritmo; es decir un modelo contiene los parámetros que deben ser determinados a partir de los datos. Hay dos factores relevantes: la función del modelo (e.g. clasificación, regresión, conglomeración) y la representación del modelo. Las representaciones del modelo que se utilizan con más frecuencia son:

a) Árboles de decisión y reglas de clasificación.

b) Modelos lineales.

c) Modelos no lineales (e.g. redes neuronales, k-means)

d) Métodos basados en ejemplos (método del vecino más cercano).

e) Modelos gráficos de dependencias probabilísticas (e.g. redes Bayesianas).

f) Modelos de aprendizaje relacional (e.g. Prolog).

La elección del modelo de representación para cada problema debe hacerse cuidadosamente ya que si se elige una representación inadecuada puede provocar que ningún entrenamiento describa la estructura del conjunto de datos de manera correcta. El modelo determina tanto la facilidad de la manipulación de los datos, así como la fácil interpretación para el usuario. Por lo general, los modelos más complejos manipulan mejor los datos, pero suelen también ser más complicados de entender y de usar confiablemente.

2. Criterio de evaluación. Son medidas cuantitativas que nos indican qué tan bien un patrón (un modelo y sus parámetros) encontrado cumple las metas (útil, novedoso, entendible, efectivo para predecir) del proceso KDD.

3. Método de búsqueda. Estos métodos, comunmente, consisten de dos elementos: búsqueda de parámetros y búsqueda de modelos. En el primer caso el algoritmo debe buscar aquellos parámetros que optimicen la evaluación del modelo, dado un conjunto de datos observados y un modelo fijo de representación. La búsqueda de modelos se presenta de forma iterativa sobre el método de búsqueda de parámetros; es decir el modelo de representación es cambiado para que una nueva familia de modelos sea considerada.

1.2.2.3. Minería de Datos.

La etapa de minería de datos del proceso KDD es considerada la más importante, debido a la integración y aplicación de forma iterativa de métodos de aprendizaje y estadísticos para la obtención de hipótesis de patrones y modelos. Esta etapa se puede definir como

> "el paso consistente en el uso de algoritmos concretos que generan una enumeración de patrones a partir de los datos procesados" *(U. Fayyad et al., 1996.)* [FPSS96a]

1.2.3. Evaluación del conocimiento.

Nosotros podemos crear modelos a nuestro gusto, pero antes de que un modelo pueda ser utilizado confiablemente, éste debe ser validado. Validación significa que el modelo está probado, de tal manera que estemos seguros que el modelo nos da valores razonables y exactos. Estas pruebas dependen directamente tanto de nuestro conjunto de datos como del campo de aplicación. La validación debe ser realizada usando un conjunto de datos independiente. Este conjunto debe ser construido semejante al conjunto de datos real utilizado para el entrenamiento, el cual no debe formar parte de este último, dicho conjunto experimental se puede considerar como representante del caso general.

Las técnicas de validación desarrolladas a lo largo de los años 80 en el campo de ML, hacen posible que las inferencias de la minería de datos sean validadas para obtener patrones o asociaciones realmente ciertas y no sólo reflejos de un manipuleo de los datos. La evaluación del modelo en cuanto a predictividad se basa en técnicas de validación cruzadas (cross validation); en cuanto a calidad descriptiva del modelo se basan en principios como el de máxima verosimilitud (maximum likelihood) o en el principio de longitud de descripción mínima (minimum description length).

1.2.3.1. Interpretación de los patrones obtenidos.

El proceso KDD no finaliza cuando los patrones han sido mostrados. El usuario debe entender qué ha sido descubierto, si este descubrimiento es novedoso, si es un nuevo conocimiento que le auxilie en la toma de decisiones, así como acotejar la coherencia que esta información representa. Para lograr esto se pueden llevar a cabo tareas tales como: la selección u ordenamiento de patrones, la visualización de los patrones extraidos o la visualización de los datos, dados los modelos extraidos. A su vez estas tareas permiten eliminar patrones redundantes o irrelevantes.

Tal es el caso de la obtención de reglas de asociación. Los algoritmos de reglas de asociación descubren patrones de la forma *Si X entonces Y*, las cuales establecen asociaciones o relaciones de correlación entre los atributos de un conjunto grande de datos. El proceso de obtención de dichas reglas se divide en dos pasos:

- Encontrar un subconjunto de datos frecuentes a partir del conjunto de datos inicial. Una alternativa es la obtención de conglomerados (clusters).
- Generar reglas de asociación entre los elementos del subconjunto obtenido en el paso anterior. Para ello se definen dos factores (de soporte y de confianza) los cuales toda regla debe satisfacer. Estos dos factores permiten cuantificar la fuerza estadística de un patrón. Entre mayores sean los valores de estos factores mayor utilidad tiene la regla.

El proceso de generación de reglas es bastante sencillo, el problema es el gran número de reglas generadas, muchas de las cuales no tienen ninguna utilidad, por lo que es necesario evaluar su validez. Además de los factores antes mencionados, para determinar el interés de una regla se pueden usar medidas subjetivas como la incertidumbre y la accionabilidad. La incertidumbre indica que las reglas son interesantes si no son conocidas por los usuarios o contradicen el conocimiento existente. La accionabilidad significa que los usuarios obtienen ventajas de las reglas.

Al llevar a cabo una interpretación de los resultados es posible regresar a cualquiera de las fases anteriores, o sea que el proceso KDD es iterativo. Se puede incluso repetir todo el proceso, quizás con otros datos, otros algoritmos, otras metas y otras estrategias. Este es un paso crucial en donde se requiere tener conocimiento del dominio del problema.

1.2.3.2. Consolidación de conocimiento descubierto.

El conocimiento descubierto se obtiene para realizar acciones, ya sea incorporándolo dentro de un sistema de desempeño o simplemente para almacenarlo y reportarlo a las personas intere-

sadas. La incorporación de este nuevo conocimiento a un sistema existente, usualmente para mejorarlo, puede involucrar corregir o resolver conflictos potenciales entre el conocimiento previo y el extraído.

Capítulo 2

Redes Neuronales y el algoritmo SOM

Por miles de años la dinámica y comportamiento del sistema nervioso ha sido una incognita para el ser humano; el estudio del sistema nervioso se remonta a la antigua Grecia, donde Platón y Aristóteles durante el siglo VI (A. de C.), dieron las primeras explicaciones teóricas del funcionamiento del cerebro y el origen de la mente, además de ser primero en establecer los principios formales del razonamiento deductivo, hasta finales del siglo XIX con el trabajo de Santiago Ramón y Cajal, quien descubriera la unidad funcional y estructural del sistema nervioso: la **neurona**. Describió prácticamente todos los tipos de neuronas de las distintas regiones del cerebro, cerebelo y la retina de muchas especies animales; pudo demostrar que las neuronas son células individuales, separadas una de otra, y que constituyen la unidad anatómica y funcional del sistema nervioso. Las neuronas están organizadas en redes de intercambio de información permitiendo que el cerebro esté conectado con todos y cada uno de los sentidos y órganos del cuerpo humano.

En 1943 Warren McCulloch y Walter Pitts proponen el primer modelo de neurona artificial, la neurona binaria que llevaba a cabo operaciones lógicas proporcionando una representación simbólica de la actividad cerebral. Las Redes Neuronales Artificiales son un intento de simular el procesamiento de información que realiza el sistema nervioso; pueden ser vistas como un sistema de procesamiento paralelo y distribuido que consiste de un gran número de unidades (neuronas) conectadas entre sí, donde cada una de ellas es un dispositivo simple de cálculo que apartir de un número variable de entradas produce una sola salida. Desde su creación, las redes neuronales artificiales han sido de mucho interés como una herramienta efectiva para la solución de distintos problemas como clasificación y reconocimiento de patrones, agrupación, aproximación de funciones, entre otros.

En 1982 Teuvo Kohonen presenta la red neuronal **Self-Organizing Map** o **SOM**, clasificada dentro de las redes neuronales de aprendizaje no supervisado y entrenamiento competitivo. El objetivo del algoritmo de aprendizaje de la red SOM, puede definirse así:

> “es un algoritmo para la visualización de datos multidimensionales que implementa un mapeo ordenado de una distribución multidimiensional en una malla regular de menor dimensión, que usualmente consiste de una malla de dos dimensiones” *(T. Kohonen, 1998.)* [Teu98]

donde radica su fuerza e importancia.

La red SOM se ha convertido en un tema importante de investigación del cual se han producido una gran cantidad de artículos y su aplicación en diversos campos de la ciencia, se utiliza cada vez con más frecuencia como una herramienta importante para el reconocimiento y visualización de patrones. Con estas bases, se han desarrollado diversos sistemas de software que permiten al usuario realizar una especificación de diseño y uso de la red SOM. Stuttgart Neural Networks Software (SNNS) y SOM Toolbox para Matlab son dos ejemplos de estos sistemas. Viscovery® SOMine® de la compañía austricaca Eudaptics Company, es un sistema de software para la minería de datos que tiene como base la aplicación de la red SOM precisamente como algoritmo de minería de datos; Viscovery® es un sistema con una interfaz amigable e interactiva con el usuario, que facilita la generación automática de mapas de conocimiento.

En la presentación de este trabajo se hizo énfasis en la necesidad de obtener a través del proceso KDD conocimiento útil, nuevo y relevate a partir de un conjunto de datos almacenados en una base de datos; hablamos también de la interpretación de los patrones obtenidos por el proceso y cómo éstos son transformados en conocimiento. Sin embargo la obtención de dichos patrones, no proporciona una representación que permita a simple vista observar la estructura, comportamiento o topología de éstos, además, tomando en cuenta que estos patrones habitan regularmente en espacios multidimensionales, resulta imposible obtener una visualización que proporcione tal información. Ante estas circunstancias, las redes neuronales artificiales representan una solución dentro del proceso KDD.

Es por esto nuestro interés hacia el estudio e implementación de la red SOM dentro del proceso KDD; la red SOM se convierte en una alternativa para la solución a estos problemas. En el contexto del proceso KDD, la obtención de mapas topológios de datos n-dimensionales implica facilitar la búsqueda y el descubrimiento de información valiosa a través de la exploración de estos mapas.

En este capítulo definimos los fundamentos teóricos de la red SOM, la importancia de su uso dentro del proceso KDD en la fase de minería de datos y el impacto de los resultados obtenidos en esta fase. Además, analizaremos los sistemas de software Stuttgart Neural Networks Software (SNNS) y SOM Toolbox para Matlab, como alternativas para automatizar el análisis y visualización de información a través del SOM.

2.1. Redes Neuronales Biológicas.

La teoría y modelado de redes neuronales artificiales está inspirada en la estructura y funcionamiento de los sistemas nerviosos, donde la neurona es el elemento fundamental. Existen neuronas de diferentes formas, tamaños y longitudes. Estos atributos son importantes para determinar la función y utilidad de la neurona.

Las neuronas son células vivas y, como tales, contienen los mismos elementos que forman parte de todas las células biológicas. Además, contienen elementos característicos que las diferencian. En general, una neurona consta de un cuerpo celular más o menos esférico, del que

sale una rama principal, el **axón**, y varias ramas más cortas, llamadas **dendritas**. A su vez, el axón puede producir ramas en torno a su punto de arranque, y con frecuencia se ramifica extensamente cerca de su extremo.

Una de las características que diferencian a las neuronas del resto de las células vivas, es su capacidad de comunicarse. En términos generales, las dendritas y el cuerpo celular reciben señales de entrada; el cuerpo celular las combina e integra y emite señales de salida. El axón transporta esas señales a los terminales axónicos, que se encargan de distribuir información a un nuevo conjunto de neuronas. Las neuronas y las conexiones entre ellas, llamadas **sinapsis**, son la clave para el procesado de la información.

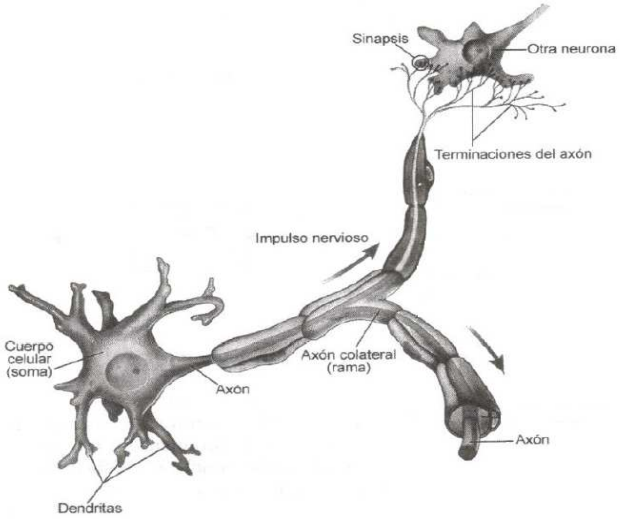


Figura 2.1: Estructura de una neurona biológica típica.

2.2. Redes Neuronales Artificiales.

2.2.1. Origen.

Alan Turing, en 1936, fue el primero en estudiar el cerebro e intentar simular su funcionamiento de manera computacional; sin embargo, los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron Warren McCulloch, un neurofisiólogo, y Walter Pitts, un matemático, quienes, en 1943, lanzaron una teoría acerca de la forma de trabajar de las neuronas.

El modelo propuesto por McCulloch y Pitts representa a cada neurona como una función booleana de dos estados: en reposo o activada. Cada neurona recibe un conjunto de señales de entrada procedentes del mundo exterior o de otras neuronas (donde cada señal es un valor numérico) y obtiene la suma ponderada de las mismas. El estado de la neurona se actualiza de acuerdo a la siguiente regla: si el resultado de la suma excede a cierto umbral entonces la

neurona se activa y por lo tanto emite una señal de salida con valor de 1. Si por el contrario es menor entonces la neurona permanecerá en estado de reposo y emite el valor de 0.

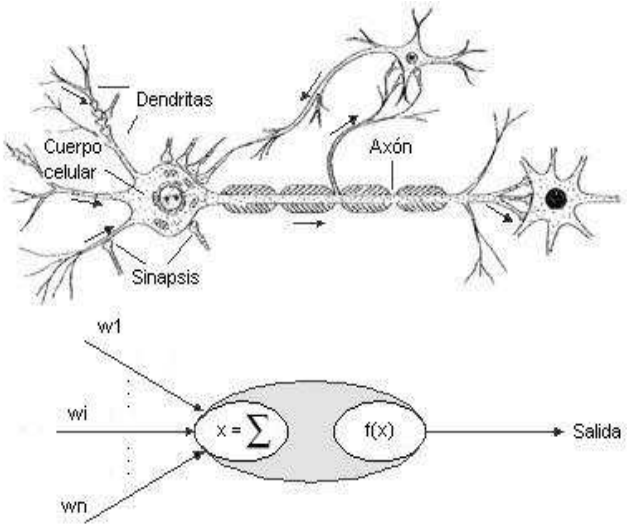


Figura 2.2: Analogía entre una neurona biológica y un neurona artificial McCulloch-Pitts.

2.2.2. Definición.

Existen diferentes formas de definir qué son las redes neuronales una de ellas es:

> "Las **Redes Neuronales Artificiales** son redes de elementos simples interconectadas masivamente en paralelo (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico"*(T. Kohonen, 1998.)* [Teu98]

La compleja operación de las redes neuronales es el resultado de abundantes lazos de realimentación y cambios adaptativos de sus parámetros, que pueden definir incluso fenómenos dinámicos muy complicados.

2.2.3. Modelo general.

La siguiente estructura genérica de neurona artificial es la establecida por el grupo Proceso Distribuido Paralelo (Parallel Distributed Processing, PDP) de la Universidad de Princeton. Se denomina neurona a un dispositivo simple de cálculo que, a partir de un vector de entrada procedente del exterior o de otras neuronas, proporciona una única respuesta o salida. Los elementos que constituyen la neurona de etiqueta i son los siguientes:

- Conjunto de entradas, $X_j(t)$.
- Pesos sinápticos de la neurona i, w_{ij} que representa la intensidad de interacción entre cada neurona presináptica j y la neurona postsináptica i.

- Regla de propagación $\sigma(w_{ij}, x_j(t))$, que proporciona el valor del potencial postsináptico $h_i(t) = \sigma(w_{ij}, x_j(t))$ de la neurona i en función de sus pesos y entradas.
- Función de activación $f_i(a_i(t-1), h_i(t))$, que proporciona el estado de activación actual $a_i(t) = f_i(a_i(t-1), h_i(t))$ de la neurona i, en función de su estado anterior $a_i(t-1)$ y de su potencial postsináptico actual.
- Función de salida $F_i(a_i(t))$, que proporciona la salida actual $y_i(t) = F_i(a_i(t))$ de la neurona i en función de su estado de activación.

De manera general, la operación de la neurona i puede expresarse como:

$$y_i(t) = F_i\Big(f_i\Big[a_i(t-1), \sigma_i(w_{ij}, X_j(t))\Big]\Big)$$

2.2.4. Arquitectura.

La arquitectura o topología de las redes neuronales consiste en la organización y disposición de las neuronas en la red formando capas o agrupaciones de neuronas más o menos alejadas de la entrada y salida de la red. En este sentido, los parámetros fundamentales de la red son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas.

Se conoce como capa o nivel a un conjunto de neuronas cuyas entradas provienen de la misma fuente y cuyas salidas tienen el mismo destino.

En las redes monocapa (1 capa) se establecen conexiones laterales entre las neuronas que pertenecen a la única capa que constituye la red. También pueden existir conexiones autorrecurrrentes (salida de una neurona conectada a su propia entrada).

Las redes multicapa son aquellas que disponen de conjuntos de neuronas agrupadas en varios niveles o capas. Normalmente, todas las neuronas de una capa reciben señales de entrada de otra capa anterior, más cercana a las entradas de la red, y envían señales de salida a una capa posterior, más cercana a la salida de la red; a estas conexiones se les denomina conexiones hacia adelante o feedforward. Sin embargo, en un gran número de estas redes también existe la posibilidad de conectar las salidas de las neuronas de capas posteriores a las entradas de las capas anteriores, a estas conexiones se las denomina conexiones hacia atrás o feedback.

En la figura 2.3 se muestran 5 redes de arquitecturas diferentes: (a) Un Perceptrón de una capa conectado completamente, (b) Un Perceptrón multicapa conectado completamente, (c) Un Perceptrón multicapa modular, (d) Una red recurrente conectada completamente y (e) Una red recurrente conectada parcialmente.

2.2.5. Mecanismo de aprendizaje.

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje

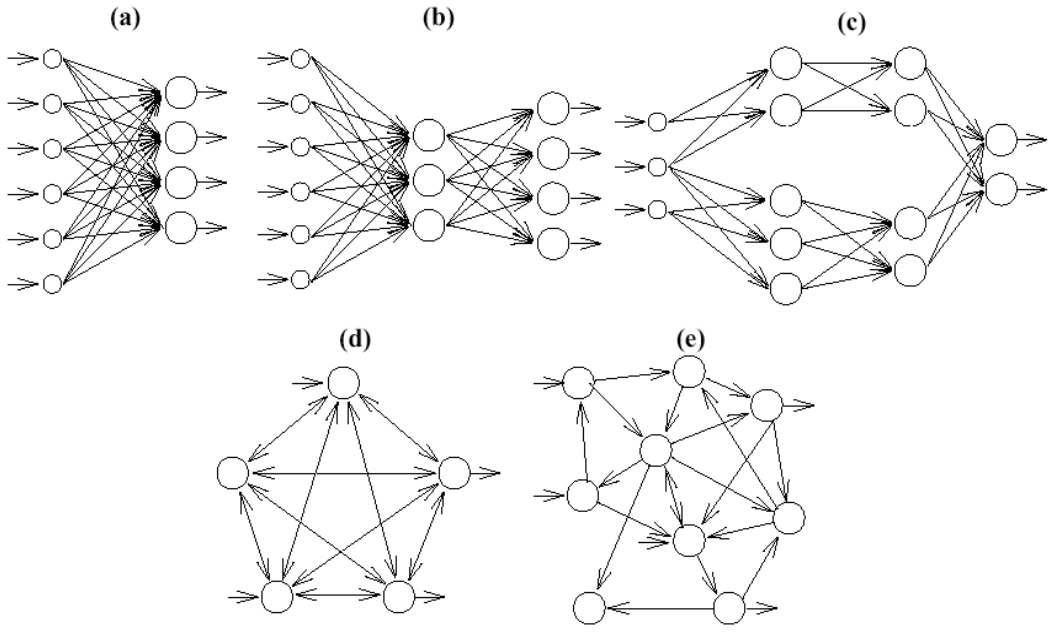


Figura 2.3: Redes Neuronales Artificiales con diferentes arquitecturas.

se reducen a la destrucción, modificación y creación de conexiones. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor distinto de cero.

Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por tanto se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables o el margen de error es menor o igual al que se ha definido como aceptable.

Un aspecto importante respecto al aprendizaje en las redes neuronales es el conocer cómo se modifican los valores de los pesos; es decir, cuáles son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red aprenda una nueva información. Estos criterios determinan lo que se conoce como la regla de aprendizaje de la red. De forma general, se suelen considerar dos tipos de reglas: las que responden a lo que habitualmente se conoce como aprendizaje supervisado, y las correspondientes a un aprendizaje no supervisado. La diferencia fundamental entre ambos tipos de aprendizaje está en la existencia o no de un agente externo (supervisor) que controle el proceso de aprendizaje de la red.

2.2.5.1. Redes con aprendizaje supervisado.

La técnica mayormente utilizada para realizar un aprendizaje supervisado consiste en ajustar los pesos de la red en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, una función de error cometido en la salida.

Existen varias formas de calcular el error y luego adaptar los pesos con la corrección correspondiente. Una de las más implementadas utiliza una función que permite cuantificar el error global cometido en cualquier momento durante el proceso de entrenamiento de la red, lo cual es importante, ya que cuanto más información se tenga del error cometido, más rápido se puede aprender.

2.2.5.2. Redes con aprendizaje no supervisado.

Las redes con aprendizaje no supervisado no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta; por ello, suele decirse que estas redes son capaces de auto-organizarse. Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presentan en su entrada.

En algunos casos, la salida representa el grado de familiaridad o similitud entre los datos que se le están presentando en la entrada y la información que se le ha mostrado hasta entonces (en el pasado). En otro caso podría realizar una agrupación o clasificación de patrones o categorías según su similitud indicando, la salida de la red, a qué categoría pertenece la información presentada a la entrada, siendo la propia red quien debe encontrar las categorías apropiadas a partir de las correlaciones entre las informaciones presentadas.

Finalmente, algunas redes con aprendizaje no supervisado lo que realizan es un mapeo de características, obteniéndose en las neuronas de salida una disposición geométrica que representa un mapa topográfico de las características de los datos de entrada, de tal forma que si se presentan a la red datos similares, siempre sean afectadas neuronas de salida próximas entre sí, en la misma zona del mapa.

2.2.6. Otra clasificación.

Kohonen divide a las redes neuronales artificiales, según su funcionamiento, en tres categorías:

1. Redes de transferencia de señal. La señal de entrada se transforma en una señal de salida. La señal atraviesa la red y experimenta una transformación de un cierto tipo. Estas redes usualmente tienen un conjunto de funciones básicas predefinidas, las cuales son parametrizadas.

2. Redes de transición de estados. Son aquellas en las cuales el comportamiento dinámico de la red es esencial. Dada una señal de entrada, la red converge a un estado estable, que, si se tiene éxito, corresponde a una solución del problema que se le presentó.

3. Redes con aprendizaje competitivo. Todas las neuronas de la red reciben la misma señal de entrada; las celdas compiten con sus vecinas laterales y la que mayor actividad tiene es la ganadora. En la siguiente sección se explicará en detalle un modelo fundamental para

el objetivo de este trabajo que se incluye en esta clasificación llamado Self-Organizing Map.

2.2.7. Ventajas.

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante. Las principales ventajas ofrecidas por éstas son:

- Aprendizaje adaptativo. Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
- Auto-organización. Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- Tolerancia a fallos. La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden recuperar.
- Operación en tiempo real. Los cómputos neuronales pueden ser realizados en paralelo, para ello se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
- Fácil inserción dentro de la tecnología existente. Una red neuronal puede ser rápidamente entrenada, comprobada, verificada y trasladada a una implementación en hardware de bajo costo, por lo que es fácil insertar dichos modelos para aplicaciones específicas dentro de sistemas existentes.

Las redes neuronales artificiales ofrecen una manera conveniente de construir un modelo implícito sin tener que formar un modelo tradicional físico del fenómeno subyacente. En contraste con los modelos tradicionales estos se pueden aplicar sin la presencia de un conocimiento a priori del problema. Estos modelos se pueden utilizar para distinguir el caso general del fenómeno actual dándonos la idea del cómo se comporta el fenómeno en la práctica. Las redes neuronales en poco tiempo se han vuelto altamente prometedoras para la solución de distintos problemas, donde los modelos tradicionales han fallado o son excesivamente complicados de construir. Debido a la naturaleza no lineal de las redes neuronales, éstas pueden expresar fenómenos mucho más complejos que algunas técnicas de modelado lineal.

2.3. Self-Organizing Map (SOM).

Teuvo Kohonen, profesor de la Facultad de Ciencias de la Información (Universidad Tecnológica de Helsinki), presentó en 1982 un modelo de red neuronal artificial con capacidad de generar de manera automática agrupaciones de datos multidimensionales y proyectar dichas

agrupaciones en mapas bidimensionales de manera que las relaciones de similitud entre los datos se representan por la cercanía de sus proyecciones en los mapas. Con este modelo intenta simular los mapas de los fenómenos sensoriales y motores existentes en el cerebro.

Esta red es de tipo auto-organizado, esto es, que de manera automática clasifica conjuntos de datos de los que no se conoce a priori ningún tipo de organización. La red, a partir de un proceso de auto-organización, proporciona un resultado, que depende de la relación de similitud existente entre dichos patrones de entrada.

El algoritmo de aprendizaje de la red SOM está basado en el aprendizaje no supervisado y entrenamiento competitivo, lo cual quiere decir que no se necesita intervención humana durante el mismo y que se necesita saber muy poco sobre las características de la información de entrada. Podríamos, por ejemplo, usar la red SOM para clasificar datos sin saber a qué clase pertenecen los mismos. La idea de entrenamiento competitivo consiste en determinar cuál de las neuronas es la que mejor representa a un estímulo de entrada dado. A esta neurona se le considera neurona ganadora y tiene la capacidad de inhibir a las otra neuronas; es decir, el vector de pesos de estas neuronas no serán ajustados de igual forma que el vector de la neurona ganadora.

Algunas características de esta red son:

- Los datos deben tener un grado de redundancia elevado para realizar su clasificación; para ello, el conjunto de datos de entrada es presentado una y otra vez al algoritmo.
- Está formada sólo por 2 capas o niveles (una capa de entrada y una de salida).
- Permite establecer relaciones de similitud en un conjunto de datos.

2.3.1. Algoritmo básico.

La red SOM está constituida por un arreglo bidimensional de neuronas:

$$H = \{\eta_1, \eta_2, \ldots, \eta_k\}$$

donde cada neurona tiene asociado un vector de pesos (o vector de referencia) representado de la siguiente forma:

$$m_i = (m_{i1}, m_{i2}, \ldots, m_{in})$$

El vector de pesos de las neuronas, es de la misma dimensión que los vectores de entrada (datos de entrada), es decir que es n-dimensional.

A su vez, la localización de una neurona en el arreglo bidimensional está representada por su vector de localización:

$$r_i = (p_i, q_i) \in \mathrm{N}^2$$

Las neuronas interactúan entre ellas por medio de relaciones laterales que se activan durante la actualización de los vectores de pesos. Estas relaciones responden a la relación de distancia física entre una neurona y sus vecinas. Usualmente, las neuronas están conectadas unas con otras en una topología hexagonal o rectangular. En la figura 2.4 podemos observar (a) una estructura hexagonal y (b) una estructura rectangular.

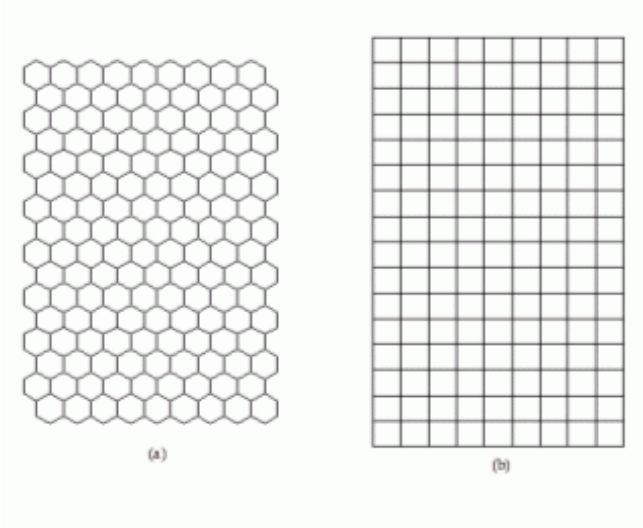


Figura 2.4: Arquitecturas usualmente usadas en SOM.

Comúnmente se definen las distancias entre las neuronas de acuerdo a la distancia Euclideana entre los vectores de localización, sin embargo, en ocasiones es más práctico usar otras funciones de distancia. En cada tiempo t se define una vecindad de actualización $N_c(t)$ con radio $\rho(t)$. Esta es una vecindad alrededor del vector de localización r_c de la neurona ganadora η_c. Dicha vecindad delimita las neuronas cuyos vectores de referencia serán actualizados en el tiempo t. El radio de actualización $\rho(t)$ es decreciente en el tiempo para lograr la convergencia del algoritmo; es decir, que éste tenga siempre un término.

En la figura 2.5 podemos observar vecindades de distintos tamaños. En el hexágono más pequeño se encuentran todas las neuronas vecinas que pertenecen a la segunda vecindad más pequeña de la neurona ubicada en el centro.

En el algoritmo básico de la red SOM, la arquitectura y el número de neuronas son determinados desde el comienzo, sin sufrir cambios durante el entrenamiento. La elección de estos dos parámetros determinan la escala del modelo resultante, ya sea obteniendo un modelo preciso o general. Un modelo preciso obtendrá muchos más grupos para poder clasificar los datos de entrada, evitando que se pueda generalizar el espacio en clases que describan adecuadamente estos datos. En el caso de los modelos generales se puede perder información que caracterice a un grupo específico, incluyendo dos grupos en un solo conglomerado.

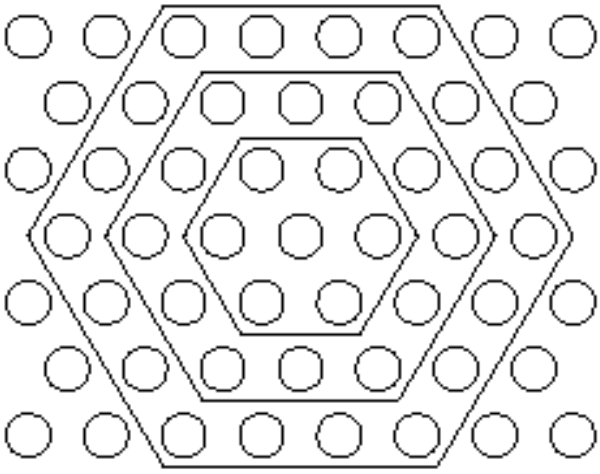


Figura 2.5: Vecindad de una neurona.

2.3.1.1. Iniciación.

Kohonen propone tres distintas formas de iniciación para los valores de los pesos: al azar, utilizando las primeras muestras e iniciación lineal. En la iniciación al azar se asignan valores aleatorios a los vectores de referencia; se utiliza cuando se sabe muy poco o nada sobre los datos de entrada en el momento de comenzar el entrenamiento. La iniciación utilizando las primeras muestras utiliza los primeros datos de entrada asignándolos a los pesos; tiene la ventaja de que automáticamente se ubican en la parte correspondiente del espacio de entrada. La iniciación lineal está enfocada a la aplicación de componentes principales; tiene la ventaja que éstos adaptan el mapa a los valores más significativos. De cualquier forma se obtiene una configuración inicial de los vectores de referencia: $\{m_1(0), m_2(0), \ldots, m_n(0)\}$.

2.3.1.2. Entrenamiento.

El entrenamiento es un proceso iterativo a través del tiempo. Requiere un esfuerzo computacional importante, y por lo tanto, consume un tiempo considerable. El aprendizaje consiste en elegir una neurona ganadora, para cada dato de entrada, por medio de una medida de similitud y actualizar los valores de los pesos en la vecindad de la ganadora; este proceso se repite varias veces para poder ir refinando (acotando) el error y acercar las neuronas a una representación más adecuada de los datos de entrada.

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 siguiente manera:

$$X = \{x(t) : t = 1, 2, \ldots, l\}$$

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

En un paso del entrenamiento, un vector muestra se toma de los datos de entrada; este vector es presentado a todas las neuronas en la red y se calcula la medida de similitud entre la muestra ingresada y todos los vectores de referencia. La unidad más parecida (Best Matching Unit, BMU) se elige como el representante con la mayor similitud con la muestra de entrada; esta similitud usualmente se define con una medida de distancia vectorial, por ejemplo la euclideana. La norma euclideana de un vector x se define como:

$$||x|| = \sqrt{\sum_{i=1}^{n} x_i^2}$$

donde;

x_i corresponde al valor de la componente i del vector x,

n corresponde a la dimensión del vector x.

Por lo tanto, la distancia euclideana en términos de la diferencia de la norma euclideana entre dos vectores se define como:

$$d_e(x, y) = ||x - y||$$

donde;

x corresponde al vector x,

y corresponde al vector y.

Para cada t la BMU(t), usualmente denotada con $m_c(t)$, es aquella con el vector de referencia que más se parece al vector de entrada $x(t)$. Se define formalmente como la neurona para la cual

$$||x(t) - m_c(t)|| = min_i\{||x(t) - m_i(t)||\}$$

donde;

$x(t)$ corresponde el vector de entrada en el tiempo t,

$m_c(t)$ corresponde al vector de referencia que representa la BMU,

i corresponde a la neurona i,

$m_i(t)$ corresponde al vector de referencia que representa la neurona $m_i(t)$.

Luego de encontrar la BMU, se actualizan todos los vectores de pesos de la red SOM.

Durante el procedimiento de actualización, la BMU se actualiza para acercarse aún más al vector de entrada. Los vecinos de la BMU también se actualizan de manera similar utilizando un factor de razón de aprendizaje de menor valor. Este procedimiento acerca a la BMU y a sus vecinos topológicos hacia la muestra ingresada.

El esfuerzo computacional consiste en encontrar una BMU entre todas las neuronas y actualizar cada uno de los vectores de referencia en la vecindad de la unidad ganadora. Si la vecindad es grande, entonces más neuronas deberán ser actualizadas; este es el caso que se presenta en el comienzo del entrenamiento, donde se recomienda utilizar vecindades grandes. En el caso de redes con muchas neuronas, se utiliza gran parte del tiempo buscando a la ganadora. Obviamente que dependiendo del diseño del software utilizado y el hardware estas consideraciones serán más o menos significativas.

A través del procedimiento de actualización descrito, la red forma una red elástica que durante el aprendizaje cae en una nube formada por los datos de entrada. Los vectores de referencia tienden a posicionarse allí donde los datos son densos, mientras que se tiende a tener pocas neuronas donde los datos de entrada están más dispersos.

La regla de actualización de la red SOM para una unidad m_i en el tiempo t, es la siguiente:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)]$$

donde;

t representa un estado en el tiempo.

Por lo tanto, y como se mencionó anteriormente, este es un proceso de entrenamiento a través del tiempo. El vector de entrada $x(t)$ es tomado en el instante t para ser procesado, h_{ci} es la función vecindad la cual es decreciente en función de la distancia entre el vector de localización r_c de la neurona ganadora η_c y el vector de localización r_i de la neurona η_i.

La función de vecindad incluye el factor de razón de aprendizaje $\alpha(t)$ el cual sirve para congelar el aprendizaje de las neuronas a lo largo del tiempo y de esta forma obtener la convergencia. Este factor de aprendizaje es una función decreciente en el tiempo que toma valores en el intervalo $(0, 1)$. Un ejemplo de función vecindad es la forma Gaussiana, la cual se define de la siguiente manera:

$$h_{ci}(t) = \alpha(t) e^{\frac{(\|r_i - r_c(t)\|)^2}{2\rho(t)^2}}$$

donde r_i corresponde al vector de localización de la neurona por actualizar, $r_c(t)$ corresponde al vector de localización de la neurona ganadora y $\rho(t)$ es el radio de actualización en el tiempo t.

Se pueden utilizar otras funciones de vecindad como la función que se presenta en la figura 2.6. La única restricción es que sea decreciente en función de las distancias alrededor de la neurona η_c. Por lo tanto, también podría ser constante dentro de la vecindad de la neurona ganadora $N_c(t)$.

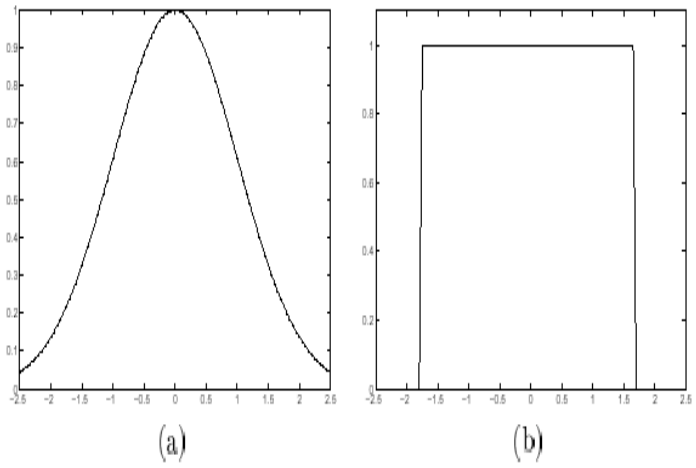


Figura 2.6: Funciones de vecindad.

En la figura 2.6 se pueden observar dos funciones de vecindad: (a) función Gaussiana y (b) función constante.

El factor de razón de aprendizaje utilizado en la función vecindad es una función decreciente en el tiempo. Dos formas comúnmente usadas son la función lineal y la inversamente proporcional al tiempo *t*.

En la figura anterior 2.7 se pueden observar tipos de factores de razón de aprendizaje: (a) la función lineal decrece a cero linealmente durante el aprendizaje y (b) la función inversamente proporcional decrece rápidamente desde su valor inicial.

Los valores de factor de razón de aprendizaje α se definen de la siguiente manera:

$$\alpha(t) = \alpha(0)(1 - \frac{t}{r})$$

para $t < r$ y $\alpha > 0$, donde r corresponde al número de vectores de entrada utilizados en el entrenamiento.

Se debe determinar el valor inicial de α, que define el valor inicial del factor de razón de aprendizaje. Usualmente, cuando se utiliza una función inversa, el valor inicial puede ser mayor que en el caso lineal. El aprendizaje se realiza usualmente en dos fases:

- En la primera etapa se utilizan valores relativamente altos de α (desde 0.99 a 0.3).
- En la segunda vuelta se utilizan valores más pequeños. Esto corresponde a adaptaciones que se van haciendo hasta que la red funciona correctamente.

La elección de los valores iniciales de α y la forma en que éstos van variando, pueden transformar sensiblemente los resultados obtenidos.

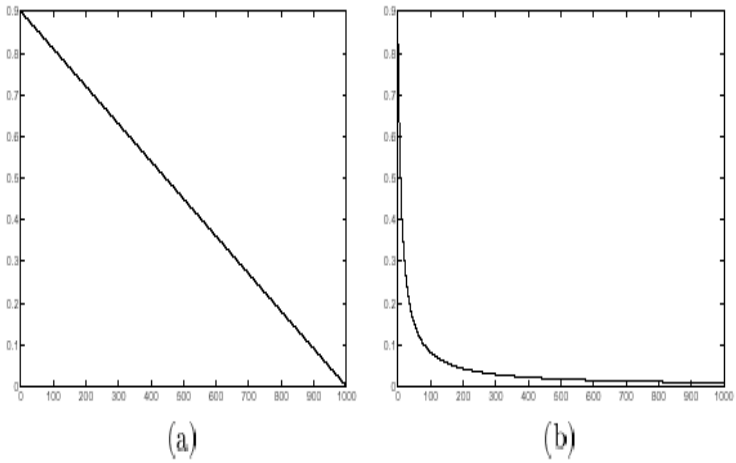


Figura 2.7: Factores de razón de apredizaje.

2.3.1.3. Visualización.

La red SOM es una aproximación de la función de densidad de probabilidad de los datos de entrada y puede representarse de una manera visual.

La representación U-Matrix (Unified distance Matrix) de la red SOM visualiza la distancia entre neuronas adyacentes. La misma se calcula y se presenta con diferentes colores entre los nodos adyacentes. Un color oscuro entre neuronas corresponde a una distancia grande, que representa un espacio importante entre los valores de los patrones en el espacio de entrada. Un color claro, en cambio, significa que las neuronas están cerca unas de otras. Las áreas claras pueden pensarse como *clases* y las oscuras como *separadores*. Esta puede ser una representación muy útil de los datos de entrada sin tener información a priori sobre las clases.

En la figura 2.8 podemos observar las neuronas indicadas por un punto negro. La representación revela que existe una clase separada en la esquina superior derecha de la red. Las clases están separadas por una zona negra. Este resultado se logra con aprendizaje no supervisado, es decir, sin intervención humana. Enseñar a una red SOM y representarla con la U-Matrix ofrece una forma rápida de analizar la distribución de los datos.

2.3.1.4. Validación.

Se pueden crear la cantidad de modelos que se quiera, pero antes de utilizar alguno de ellos, debe ser validado. Validar un modelo significa que debe ponerse a prueba para asegurar que devuelve valores razonables y certeros. La validación debe realizarse usando un conjunto independiente de datos; este conjunto de datos es similar al utilizado para el entrenamiento pero no parte de él; puede verse a este conjunto de prueba como un caso representativo del caso

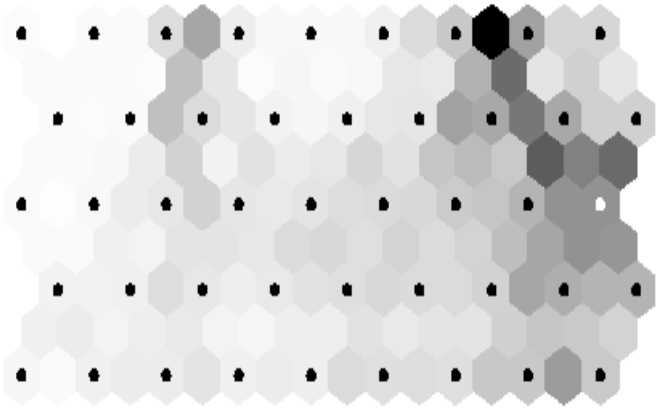


Figura 2.8: U-Matrix.

general.

2.4. SOM en la Minería de Datos.

Hemos mencionado los aspectos teóricos que definen los objetivos, arquitectura y funcionamiento de la red neuronal SOM,señalando con especial énfasis, la importancia de obtener una representación visual de la topología de los datos y las relaciones que existen entre ellos por medio de su clasificación y la formación automática de conglomerados entre datos similares.

En el contexto de la minería de datos, la red SOM es considerada como una herramienta altamente efectiva y sofisticada para el análisis de datos. La aplicación de la red neuronal y su algoritmo de aprendizaje como método y algoritmo para la míneria de datos, se traduce en resultados con un alto impacto en la toma de decisiones posteriores al análisis de estos resultados.

En 1997, Samuel Kaski en [Sam97] hace un importante estudio donde introduce la red SOM para la minería de datos en textos, obteniendo como resultado, estructuras visibles dentro del mapa que facilitan la observación de las relaciones existentes entre neuronas vecinas. En años recientes la minería de datos en textos, se ha convertido en un tema importante para la realización de distintas investigaciones; como resultado de éstas, se ha logrado establecer diversos métodos a través de los cuales se lleve acabo la búsqueda de estos patrones de forma consistente: búsquedas por medio de palabras clave (keywords), exploración del contenido de una colección de textos organizados de alguna forma previamente definida y el filtrado de textos.

Cabe señalar que la organización de los textos se realiza bajo un esquema jerárquico, que el usuario define dependiendo de los resultados a los que desee llegar. El filtrado de textos se refiere principalmente a descartar del conjunto de textos aquellos que no sean del interés del

usuario.

> "El algoritmo SOM de Kohonen puede ser usado dentro del campo EDA para la minería de datos de grandes conjuntos de datos multidimensionales" *(S. Kaski, 1997.)* [Sam97]

En 1996 la NASA requería implementar un control de calidad en sus bases de datos (incluyendo bases de datos en línea) y catálogos de literatura astronómica, todos ellos recopilados desde 1989. Se desarrolló una interfaz de usuario para tal fin, donde la red SOM agrupaba una serie de palabras clave o *keywords* de la literatura utilizada dentro del mapa, donde la dimensión de cada entrada tenía un estimado de 463 elementos; los resultados obtenidos fueron más allá de lo que se esperaba. El mapa agrupó los keywords por su similitud y gracias a esto, fue posible crear nuevos catálogos de bases de datos astronómicas con base en la información recuperada a través del mapa.

La cantidad de información generada por la Astronomía y Astrofísica va desde bibliografía general, hasta datos de estrellas, galaxias, etc. *SIMBAD* (Set of Identifications, Measurements and Bibliography for Astronomical Data) es una base de datos creada y administrada por el Centre de Données Astronomiques de Strasbourg (CDS, Francia), que contiene información acerca de 1,500,000 estrellas, 1,250,000 objetos no estelares como galaxias, nébulas planetarias, entre otras. SIMBAD da acceso a alrededor de 10,321 artículos de Astronomía y Astrofísica; la organización de estos artículos en distintas áreas con base en sus keywords se lleva acabo por la red SOM. El website de SIMBAD presenta este mapa de forma que el usuario puede interactuar de tal manera, que presionando el mouse sobre alguna neurona de la red SOM, el mapa despliega la información asocidada a esta neurona, véase la figura 2.9. Para mayor información consultar el website de SIMBAD en la siguiente dirección: `http://simbad.u-strasbg.fr/A+A/map.pl`.

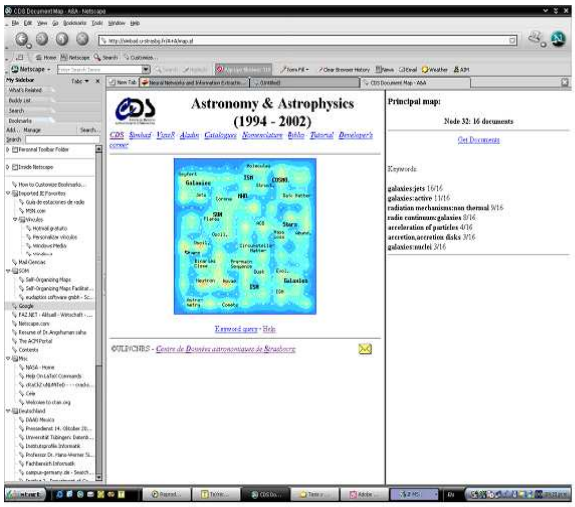


Figura 2.9: Base de datos SIMBAD.

En 1994 Gerhard Kranner crea en Viena el Centro de Investigaciones en Software, que ahora conocemos como Eudaptics Company, que durante un periodo de tiempo de 3 años desarrollaron distintos productos de software que implentara la red SOM. Casi al mismo tiempo en que Kaski llevaba a cabo su estudio de aplicación de la red SOM a la minería de textos, Eudaptics creaba el sistema **Viscovery®**, un sistema de software comercial para la minería de datos, que integra multiples métodos estadísticos.

Actualmente el sistema Viscovery® es una suite que consta de distintos módulos para el análisis de información como: evaluación y análisis de riesgos, marketing, análisis de fraudes, planeación y monitoreo de procesos, análisis genético, diagnósticos médicos, entre otros. Dentro de la suite se encuentra **Viscovery® SOMine®**, que como su nombre lo indica es el módulo que realiza el proceso de minería de datos a través de la red SOM; Viscovery® SOMine® implementa algunas variantes de la red SOM como el algoritmo Batch Map para el entrenamiento de la red, además de definir vecindades de tipo hexagonal en el mapa.

Viscovery® SOMine® organiza los datos de dimensión n en mapas, con base en su similaridad; el mapa resultante puede ser usado para indentificar y evaluar características ocultas en los datos. La principal arma de Viscovery® SOMine® es el despliegue visual de los mapas generados por la red SOM; el sistema permite interactuar con el mapa de tal manera que es posible obtener los datos que contienen las neuronas, despliege de distintos algoritmos de visualización y de conglomerados, entre otras 2.10.

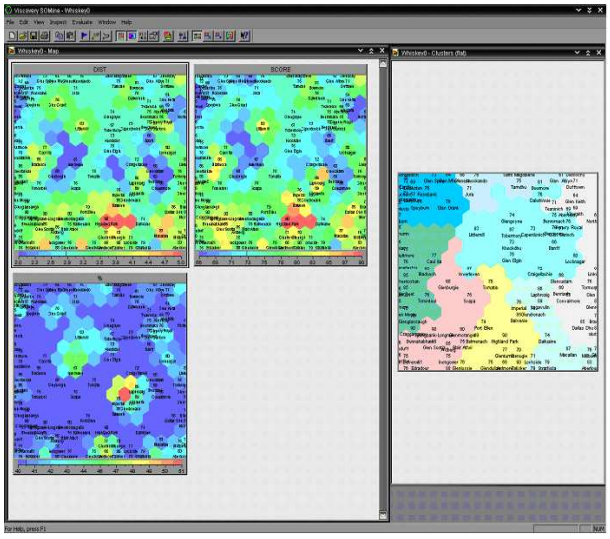


Figura 2.10: Mapas de la red SOM generados por Viscovery® SOMine®.

El sistema Viscovery® SOMine® es acutalmente uno de los sitemas de minería de datos más importantes dentro del mercado; algunas de las empresas que lo utilizan son Visa y Green Peace.

Dentro del campo de la Bioinformática actualmente, la búsqueda de patrones ocultos den-

tro de largas series de información genética es de suma importancia; los avances en el estudio del cáncer han traido consigo, complejos cambios en los patrones genéticos. Para poder identificarlos es necesario realizar análisis dentro del código genético y así, obtener un diagnóstico confiable acerca del tipo de tumor, terapia, etc. del paciente. Como sabemos, este tipo de información es demasiado compleja, y el espacio en el que habitan los datos es mucho mayor del que podemos analizar utilizando métodos clásicos. Una herramienta para la minería de datos como la red SOM, es necesaria.

La mayor parte de estos análisis están basados en hipótesis tomadas a priori, perdiendo en muchas ocasiones información precisamente por la naturaleza multidimensional de los datos y el método de análisis. Por medio de la red SOM, se ha podido identificar mas información de la que se había obtenido anteriormente, además de facilitar la inspección de la información dentro de los conglomerados que integran el mapa, generando nuevas hipótesis con una mínima supervisón del algoritmo.

En los sectores financieros, económicos y de marketing, la red SOM para la minería de datos tiene distintas aplicaciones:

- Análisis de estados financieros.
- Análisis de oportunidades de inversión.
- Monitoreo de desempeño financiero.
- Predicción del comportamiento de mercado.
- Pronósticos financieros.
- Pronósticos de indicadores macro-económicos.
- Administración de conocimiento en Economía.
- Análisis de preferencias de consumidores.
- Predicción del comportamiento de consumidores.
- Segmentación de mercado.
- Definición de estrategias de mercado.

2.4.1. Ventajas y beneficios de la red SOM en la Minería de Datos.

En la minería de datos, el objetivo de la red SOM es crear un mapa en 2 dimensiones, de los datos preservando la topología del espacio en cual viven estos datos, con el fin de analizar las relaciones (que en muchos casos se encuentran ocultas o son no lineales) entre ellos.

Anteriormente mecionamos algunas aplicaciones de la red SOM en la minería de datos, sin embargo: ¿por qué utilizarlo?. Algunas ventajas que podemos mencionar de la red SOM son las siguientes:

- Atenuación local de las neuronas. El ordenamiento topológico trabaja de inicio en una vecindad de radio previamente definido para cada neurona del mapa, es decir, la actualización se realiza sobre las neuronas que se encuentren dentro del rango de este radio y conforme el proceso de entrenamiento avanza, el radio de la vecindad disminuye hasta que el ordenamiento o modificación del valor de las neuronas se realize de forma local (figura 2.11).

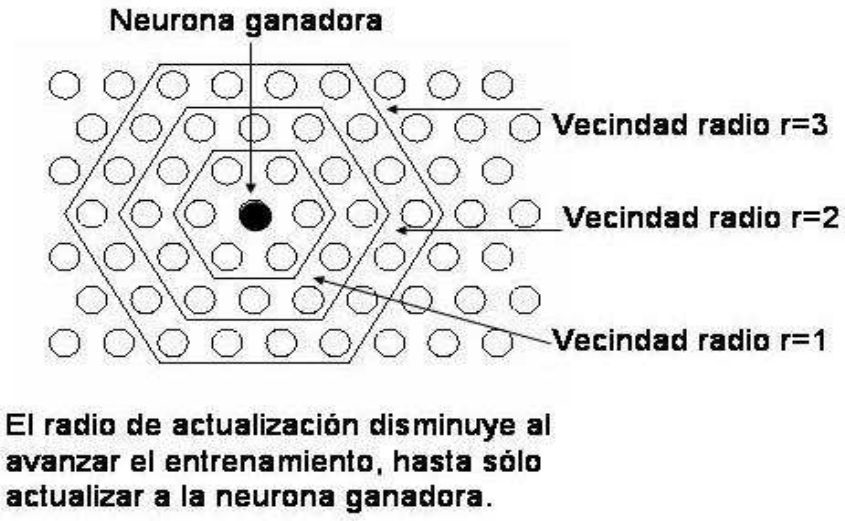


Figura 2.11: Disminución del radio de actualización durante el entrenamiento.

- Visualización sencilla. El mapa es representado como una malla regular (rectangular o hexagonal), facilitando la construcción de diversos tipos de visualizaciones (U-Matrix, visuzalización de componentes, relaciones de vecindad de cada neurona, frecuencia, error de cuantización, entre otras) e implementación de interfaces de usuario.

- Conglomeración automática. La red SOM es considerada también como un algorítmo de conglomeración; cada neurona es considerada como un conglomerado por sí mismo. Cabe señalar que la red SOM genera los conglomerados sin previa información de los datos que le presentan como vectores de entrada.

- Complejidad. La complejidad computacional de la primera iteración del entrenamiento es de *O(md)* donde *m* es el número de unidades del mapa (neuronas) y *d* la dimensión de los vectores de entrada, esto implica, que al término del entrenamiento, la complejidad total es de *O(nmd)*, donde *n* es el número de vectores de entrenamiento.

  > "La complejidad total del proceso de entrenamiento depende del número de iteraciones del entrenamiento" *(J. Vesanto, 2000.)* [Juh00]

  Si el número de neuronas es proporcional a $\sqrt{n}$, la complejidad del entrenamiento crece de forma lineal con respecto al tamaño de los datos en *O(nd)*.

- Reemplazo de valores perdidos. El algoritmo de entrenamiento soluciona eficientemente este problema; si el vector de entrada presenta uno o más valores faltantes, la red SOM sólo toma aquellas coordenadas donde sí los halla.

2.5. Implementaciones de la red SOM en Matlab y SNNS.

2.5.1. SOM Toolbox para Matlab.

Matlab es un entorno de programación totalmente integrado creado en 1984 por MathWorks, Inc. Matlab está orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos. Actualmente dispone de una amplia gama de bibliotecas de apoyo especializadas, denominadas Toolboxes, que extienden significativamente el número de funciones incorporadas en el programa básico. Destacando entre éstos se encuentra el SOM Toolbox, creado en 1997 por un grupo de trabajadores, llamado equipo SOM Toolbox, del Laboratorio de Información y Ciencias de la Computación de la Universidad Tecnológica de Helsinki.

Para poder hacer uso de SOM Toolbox es necesario tener instalado Matlab versión 5.2 o posterior. Matlab está disponible para los sistemas operativos Windows, Unix y Mac OS X, del cual se puede obtener una versión de prueba en internet. El SOM Toolbox está disponible, en el URL `http://www.cis.hut.fi/projects/somtoolbox/`, bajo la licencia GNU [1] (General Public License), la cual permite usarlo, modificarlo y distribuirlo bajo los estatutos establecidos por los autores.

SOM Toolbox provee herramientas para el diseño, implementación, visualización, simulación y análisis de la red neuronal de Kohonen, de la cual se han mencionado sus amplias ventajas anteriormente. Cabe señalar que el grupo de desarrollo trabaja en el campo de minería de datos, por lo que el Toolbox está enfocado a este propósito permitiendo visualizaciones eficaces.

2.5.1.1. Características generales.

Es necesario que el conjunto de datos de entrada se proporcione en forma de tabla, de forma que cada renglón represente un vector de entrada. A su vez, los elementos en un renglón indican las variables o componentes significativas del conjunto de datos. Es importante que cada dato de entrada esté compuesto por el mismo conjunto de variables. Con esto se tiene que cada columna de la tabla, representa todos los posibles valores para una determinada variable. Algunos de estos valores pueden ser omitidos, pero la mayoría deben estar indicados. El tipo de valores para los datos de entrada para llevar a cabo un entrenamiento, deben ser numéricos. Existe la posibilidad de hacer uso de cadenas, pero únicamente para el etiquetado de los datos, orientado a un posible análisis posterior. Este conjunto de datos debe introducirse adaptándolo a una estructura de datos predeterminada.

Ya hemos mencionado que en el preprocesamiento de datos se puden realizar tareas como transformaciones o normalizaciones, filtros para eliminar valores erróneos o no atractivos, entre

[1] GNU Proyecto de Software Libre.

otras. En el Toolbox, únicamente se permiten realizar simples transformaciones y normalización (escalamiento) de variables.

Para llevar a cabo un entrenamiento es necesario indicar la forma en que los vectores de referencia serán iniciados. Toolbox provee dos formas: aleatoria y lineal. También es necesario indicar el algoritmo con el cual se realizará el entrenamiento, para ello se cuenta con dos opciones: algoritmo SOM básico y algoritmo batch map. El entrenamiento es realizado en dos fases: entrenamiento riguroso, tomando en cuenta un radio de vecindad y un factor de razón de aprendizaje grande, y entrenamiento de optimización, con un radio de vecindad y un factor de razón de aprendizaje pequeño. La medida de similitud utilizada en el SOM Toolbox es la distancia euclideana.

Sabemos que las neuronas son conectadas con las neuronas adyacentes por una relación de vecindad, la cual dicta la topología del mapa. En el SOM Toolbox, la topología es dividida en dos factores: estructura local de la malla y la forma global del mapa. En la figura 2.12 se muestran 3 formas de mapas diferentes: (a) el mapa usado en cualquier entrenamiento por omisión, (b) un cilindro y (c) un toro.

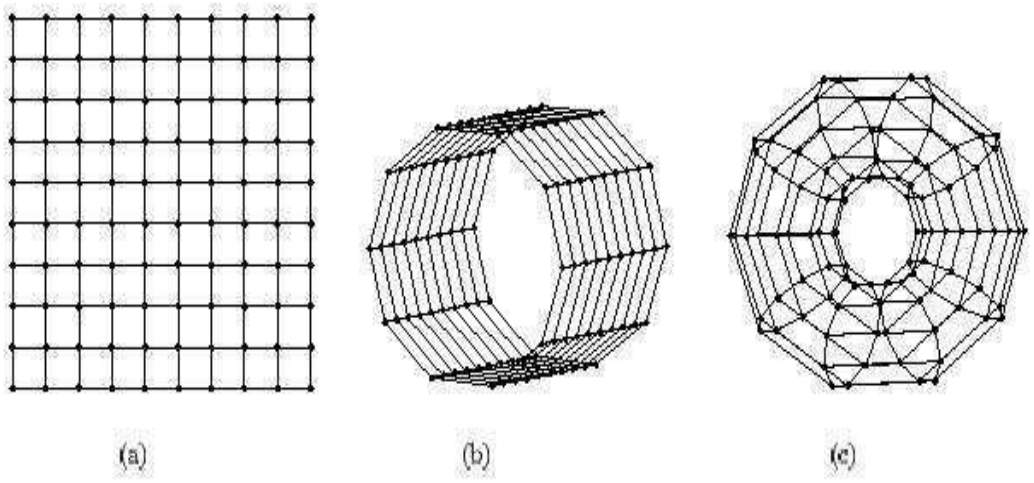


Figura 2.12: Distintas formas de mapas para utilizar en SOM Toolbox.

En el ToolBox, existe un conjunto de funciones para la visualización del mapa obtenido con el SOM. éstas se dividen en tres categorías:

- Visualizaciones de neuronas, están basadas en mostrar la malla del mapa como está en el espacio de salida.
- Visualizaciones gráficas, muestran gráficos sencillos en la localidad de cada componente del mapa.
- Visualizaciones de aclopamientos, muestran el mapa como una diagrama donde destaca la dispersión del mismo.

tamaño de los datos	elementos en el mapa	algoritmo batch	algoritmo básico	som_pak
30010	30	0.3 s	2.7 s	1.9 s
3000010	300	24 s	76 s	26 s
3000010	1000	13 min	40 min	15 min

Cuadro 2.1: Tiempos empleados al ejecutar 3 vertientes de algoritmo para la red neuronal de Kohonen utilizando SOM Toolbox.

Finalmente se permite realizar análisis de los mapas obtenidos aplicando algoritmos de conglomerados y clasificación.

En esta sección se mostrarán los resultados de las evaluaciones realizada por el grupo de desarrollo, a SOM Toolbox. El propósito de estas pruebas de funcionamiento son únicamente para evaluar el trabajo computacional de los algoritmos. No existe la intención de comparar la calidad entre los mapas resultantes, ya que no hay un método universalmente correcto reconocido para evaluarlos. Las pruebas fueron realizadas en una estación de trabajo con un solo procesador Pentium II a 350 MHz, con 128 de memoria RAM y Linux como sistema operativo. La versión de Matlab fue 5.3.

Las pruebas fueron realizadas con conjuntos de datos y mapas de diversos tamaños, y tres algoritmos de entrenamiento: `som_batchtrain`, `som_seqtrain` y `som_sompaktrain`, este último forma parte de SOM_PAK [2]. Los resultados generales se muestran en la tabla 2.1

En la tabla 2.1 el tamaño del conjunto de datos está dado por *n* x *d*, donde *n* es el número de datos de entrada y *d* es la dimensión de éstos.

Como conclusiones generales puede decirse que la función `som_batchtrain` fue claramente la más rápida. La función `som_batchtrain` es especialmente rápida cuando se aplica a grandes conjuntos de datos, mientras que con conjuntos de datos pequeños y mapas grandes es ligeramente más lento.

2.5.1.2. Ventajas y desventajas.

A pesar de que SOM Toolbox es un paquete con licencia GNU, es necesario tener a nuestro alcance Matlab para su funcionamiento. Matlab es un producto comercial el cual puede ser adquirido por dos medios; comprándolo con un costo de 1,900 dólares o mediante la obtención de una licencia con fines totalmente académicos.

Las redes neuronales requieren un uso intensivo de matrices, peculiaridad de la cual no se escapa la red SOM, Matlab provee un ambiente natural para la rápida implementación de éstas, lo que permite una mejor eficiencia de los algoritmos de entrenamiento reflejándose en su velocidad. A pesar de los beneficios que se obtienen con el manejo de matrices, éste tiene su complicación al emplear demasiada memoria. A su vez, Matlab permite la creación de funciones propias para la manipulación de la red neuronal SOM, aunque para esto es imprescindible saber

[2] SOM_PAK: Paquete, desarrollado en la Universidad Tecnológica de Helsinki (para UNIX y MS-DOS), para aplicar el algoritmo SOM.

programar en Matlab, o al menos, tener conocimientos previos de un lenguaje de programación estructurado para una pronta incorporación a la sintaxis de Matlab.

El SOM Toolbox dispone de una interfaz gráfica para el usuario (Graphical User Interface, GUI) a partir de la cual se puede crear, iniciar, entrenar, simular y manipular la red neuronal. Siempre queda la posiblidad de llevar a cabo todas estas tareas desde la línea de comandos tradicional de Matlab.

El SOM Toolbox nace ante la búsqueda de una sencilla implementación de la red neuronal SOM en Matlab, con propósitos de investigación. SOM Toolbox toma como referencia de partida SOM_PAK, por lo que este toolbox añade la posibilidad de incorporar los datos de entrada de la misma manera como se realiza en el SOM_PAK. Esta característica incrementa la cantidad de gente favorecida con este paquete, al ser reutilizables sus datos.

2.5.2. SNNS.

Sttutgart Neural Networks Software o **SNNS** es un simulador de distintos modelos de redes neuronales artificales e implementa distintos tipos de algoritmos de entrenamiento; entre estos modelos, se encuentra la red SOM. SNNS fue desarrollado y es distribuido por la Universidad de Sttutgart como software libre, sin embargo, no forma parte del software con licencia GNU ni es Licencia de Dominio Público o también conocida como **GPL**. SNNS está disponible para los sistemas operativos Windows, Unix y Unix-Like en dos distintas distribuciones: Java y `X11` [3].

2.5.2.1. Características generales.

SNNS presenta al usuario un menú principal que contiene distintas opciones:

- `FILE HANDLING`. File permite abrir archivos. SNNS maneja tres tipos de archivos diferentes en texto plano, cada uno de ellos esta define tareas en específico:
  - Archivos *.`NET`. Estos archivos definen la topología de la red y reglas de aprendizaje.
  - Archivos *.`PAT`. Son los archivos que contienen los datos de entrenamiento.
  - Archivos *.`RES`. Contienen el resultado de las salidas generadas por la red; estos archivos pueden ser interpretados de distintas formas, dependiendo del tipo de red y el tipo de problema que se esté trantando.
- `CONTROL`. Es un panel donde se asignan los valores a las variables que intervienen en la red: número de pasos, ciclos, patrones (o tambien se asignan los patrones a un archivo en específico), validación.
- `DISPLAY`. Despliega en pantalla la red cargada previamente.
- `DISPLAY 3D`. Despliega en pantalla la red cargada previamente en 3 dimensiones espaciales.

[3] Sistema de interfaces gráficas de arquitectura cliente-servidor, donde la aplicación cliente se ejecuta en el host local y sus procesos en uno o más servidores remotos.

- `BIGNET`. Esta opción permite al usuario definir una nueva red neuronal.

En ambas distribuciones SNNS permite al usuario crear distintos modelos de redes neuronales, entre los cuales se encuentran: general, que SNNS lo toma como Feed-Forward, redes Art 1 y 2, ARTMAP, redes de Jordan, Kohonen, Hopfield y Elman. Para cada red, SNNS permite asignar el valor de las varialbes que intervienen en la red como número y tipo de unidades (entrada, ocultas o de salida), reglas de aprendizaje, pesos inciales, asignación de las conexiones entre las neuronas, etc. Dentro de los algoritmos de apredizaje SNNS tiene como opciones LVQ (Learning Vector Quantization) dinámico, backpropagation, counter propagation, recurrent cascade correlation, entre otros.

Una vez que el usuario define las características de la red, `DISPLAY` permite la visualizacíon del modelo en una nueva pantalla del sistema. Esta visualización puede ser en 2D ó en 3D; SNNS implementa un control dentro de la ventana de visualización que permite la traslación y/o rotación del grafico sobre los ejes coordenados. Para la distribución en Java la visualización sólo es en 2D y aún no están implementadas las opciones de rotación y translación. Más adelante abordaremos este punto.

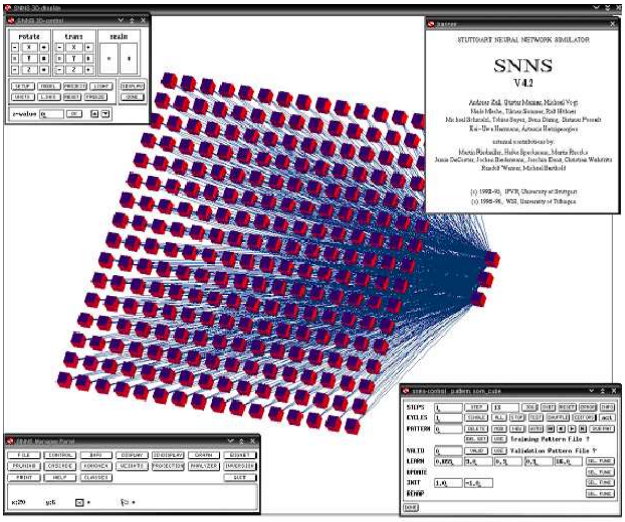


Figura 2.13: Controles y visualización en 3D de la red SOM.

2.5.2.2. Ventajas y desventajas.

Si bien un sistema de simulación como lo es SNNS automatiza el diseño de redes neuronales, al mismo tiempo presenta una serie de características a tomar en cuenta si se desea implementar la red SOM con este sistema.

Bajo la distribución `X11` la red neuronal realiza los cálculos necesarios a través de la red en uno o más servidores remotos, disminuyendo el número de opeaciones que realiza el host local, además se debe tener instalado en el host el programa cliente de `X11`; sin embargo, si no

se cuenta con una red de datos de por lo menos 100 mbps, la transferencia de datos entre el host local y los servidores remotos disminuye significativamente. En este sentido la distibución en Java pesenta serias limitaciones; como sabemos las operaciones necesarias que reuquiere un programa en Java, se hacen por medio de peticiones de recursos al sistema por parte de la máquina virtual, en conscuencia, una red neuronal con un número considerable de unidades requiere de una cantidad importante de operaciones, esto se traduce en un consumo desmedido de recursos del sistema (memoria, procesador,etc.).

El ambiente gráfico de los sistemas de software desarrollados bajo `X11`, están basados en los desarrollados para los sistemas operativos Unix y forma parte de las distintas opciones de ambientes gráficos para diversas distribuciones Unix Like; `X11` no proporciona una interfaz completa al usuario, es decir, cada ventana dentro de los sistemas `X11` son programas individuales. Para el usuario interactuar con SNNS en esta distribución, resulta complicado si éste no está familiarizado con el sistema `X11`, pues SNNS abre una nueva ventana por cada acción que realiza el sistema, lo cual no permite que el usuario "navegue" con libertad en la interfaz; además presenta problemas de implementación ya que si las ventanas son cerradas con el control adecuado, termina por completo la ejecución de todo programa. Como hemos mencionado `X11` maneja una arquitectura de cliente-servidor, por lo que las sesiones tienen un límite de tiempo para ser ejecutadas; terminado el tiempo establecido, es necesario cerrar el programa cliente y reiniciar el servidor desde el host local.

Poder crear y observar distintos modelos de redes neuronales por medio de un software significa, en el sentido práctico, interactuar directamente con los modelos lo cual permitiría identificar más facilmente problemas de implementación o interpretar de forma más clara el proceso que realiza la "caja negra" que toda red neuronal contiene. Podemos afirmar entonces, que la visualización juega un papel importante dentro de esta búsqueda de la solución que uno espera; en este sentido, SNNS en ambas distribuciones no es una herramienta muy recomendable.

Las imágenes de redes pequeñas (en cuanto número de unidades) generadas por SNNS en ambas distribuciones ofrecen una descripción amplia del modelo, sin embargo el problema aparece cuando el número de neuronas es mayor. Para `X11` la visualización en 3D es de buena calidad pero sólo se puede interactuar con esta representación del modelo por medio del control de 3D, que solamente permite rotaciones y traslaciones sobre los distintos ejes.

Existen algunos otros problemas; cuando aparecen los gráfcios de los modelos, en éstos no aparecen nunguna etiqueta. Existen datos que para el usuario es importante conocer tales como los valores de los vectores de pesos, las conexiones que existen entre las neuronas, su valor o su etiqueta; SNNS tiene las opciones para mostrar esta información en los gráficos, pero como podemos ver en la figura 2.14, con un número considerable de unidades y conexiones estos datos son practicamente ilegibles. Para habilitar estas opciones o modificar algún parámetro de la red es necesario utilizar el control `SETUP` y `CONTROL`.

Para la distribución en Java, SNNS utiliza las bibliotecas de javax.Swing proporcionando un ambiente gráfico más amigable para el usuario, sin embargo la visualización de las simulaciones carecen de calidad. Hasta ahora, en esta distribución sólo es posible ver los modelos en 2D, además, los modelos son dibujados sobre un administrador de distribución en forma de malla

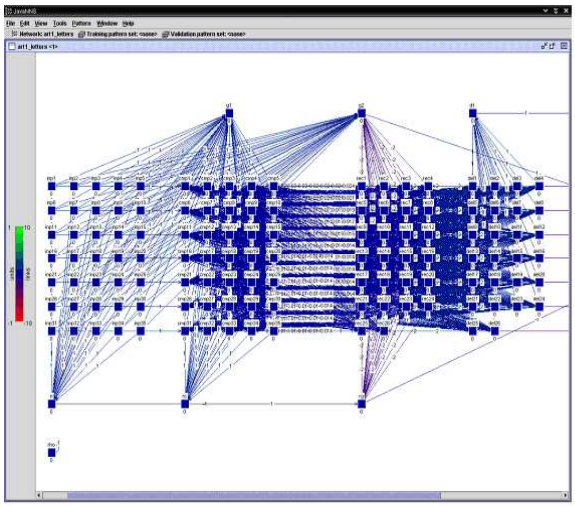


Figura 2.14: Ejemplo de una red neuronal donde no se aprecian las etiquetas de las componentes de la red en java.

o *GridLayout*[4] donde cada nodo del GridLayout representa una neurona; entre cada neurona existe un nodo vacío en el GridLayout y a cada uno de ellos se le asigna una coordenada dentro de la ventana o *frame*. Los modelos son dibujados sobre el GridLayout y por omisión, tampoco aparecen las etiquetas de los componentes de la red.

Al igual que SNNS en `X11`, se pueden habilitar estas opciones, pero el problema de ilegibilidad es el mismo, como se muestra en la figura. Como hemos mencionado, el impacto de los resultados de la red SOM radican en la observación e interpretación de la información que contiene el mapa. Consideramos que la visualización de los mapas generados por la red SOM es mejor en la distribución `X11`; la calidad de los gráficos permiten una mejor interpretación de los resultados, pero con las limitaciones antes mencionadas. La visualización en 3D no permite desplegar la información asociada de los elementos de la red.

En el comienzo de esta sección mencionamos que SNNS maneja tres tipos de archivos, éstos son creados por el sistema para definir la arquitectura de la red, los datos del entrenamiento y los resultados. El usuario tiene acceso a tales archivos, por lo que es posible modificar directamente desde los archivos algunas de las propiedades defininas en ellos; sin embargo, el formato en el que estan definidos no representan una buena opción.

[4]GridLayout, clase del paquete java.awt de Java J2SE 1.4.2.

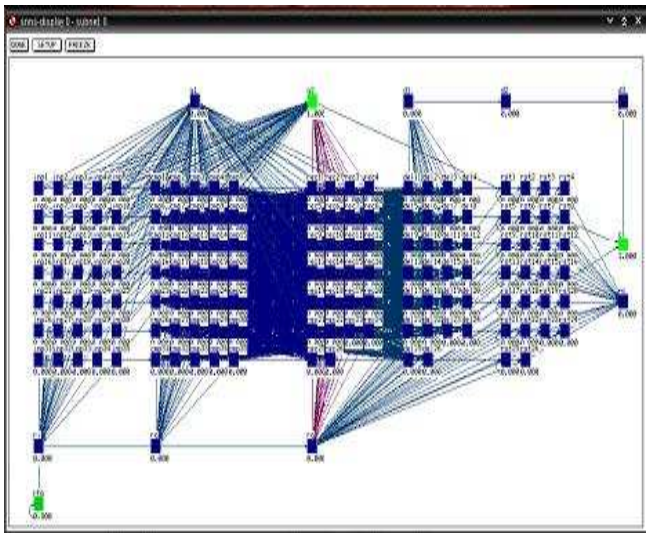


Figura 2.15: Ejemplo de una red neuronal donde no se aprecian las etiquetas de las componentes de la red en X11.

Capítulo 3

Metodología ViBlioSOM.

En los capítulos anteriores, hemos mencionado los conceptos teóricos del proceso KDD y la red neuronal SOM, además de sus aplicaciones, impacto, ventajas y desventajas. Podemos resaltar dos aspectos muy importantes: la necesidad de obtener una representación visual de las relaciones entre datos sumamente complejos para su análisis, por parte del proceso KDD y el poder de la red neuronal SOM, para organizar de forma automática los datos y su representación sobre mapas bidimensionales. Al mismo tiempo, se mencionaron algunas de las tecnologías desarrolladas para la automatización del proceso KDD y tres sistemas de software que implementan la red neuronal SOM para distintos propósitos.

Podemos pensar en la combinación de ambos conceptos, de tal forma que dentro del proceso KDD, la fase de minería de datos sea llevada a cabo por la red neuronal SOM y por medio de un algoritmo de conglomeración y la información obtenida sea representada en un mapa bidimensional. Sin embargo, este escenario no es del todo real.

Actualmente existen distintas herramientas que realizan este tipo de procedimientos, sin embargo muchas de ellas presentan limitaciones. Una de estas es la falta de validación de sus resultados o que son, en muchos de los casos, desarrolladas para determinada área temática o para solucionar un problema en particular. A estos factores, se le suma lo inaccesibles que pueden ser estos sistemas de software por lo costoso o por estar destinados a ofrecer servicios, y no a su comercialización. Esta situación ha llevado al surgimiento de diversos “modos de hacer”, y a la adaptación de distintos sistemas con propósitos diferentes para tratar de automatizar las etapas de la minería de datos y llegar al descubrimiento de conocimiento. Estas razones han motivado el surgimiento de diferentes metodologías entre las que está el ViBlioSOM®.

ViBlioSOM® (Visualización - Bibliometría - Mapas Auto-Organizados(SOM)), es una metodología desarrollada para el análisis bibliométrico que se vale, en una de sus etapas, de la visualización en forma de mapas auto-organizados. La metodología ViBlioSOM® está basada en el uso de distintos sistemas de software propietarios; uno de estos sistemas es el Viscovery® SOMine®.

En la última década, ViBlioSOM® ha obtenido importantes resultados en el descubrimiento de conocimiento y en la investigación documental, además ha permitido organizar visualmente

la información bibliométrica y ha ayudado a percibir la estructura topológica de un conjunto de datos. Pero como acabamos de mencionar, ViBlioSOM® usa distintos sistemas de software que si bien han subsanado algunas de las necesidades de los usuarios, no dejan de presentar ciertos inconvenientes, por ejemplo:

- La complejidad que han alcanzando algunos ejercicios y casos reales sobrepasaban los límites de procesamiento de algunos de los módulos o software utilizados por ViBlioSOM®.
- Existen limitaciones en cuanto a la aplicación de algunos indicadores bibliométricos más complejos y necesarios en las actividades de inteligencia empresarial o vigilancia científica-tecnológica como la identificación de señales débiles.
- El hecho de disponer de diferentes módulos o software crea incertidumbre en los usuarios al tener que emigrar de unas interfaces a otra para realizar los análisis.
- El usuario de ViBlioSOM® debe dominar varias plataformas automatizadas, lo que le da una complejidad adicional.
- Necesidad de hacer evolucionar dinámicamente los módulos del ViBlioSOM® a los niveles de desarrollo que marchan las tecnologías de la información para hacerlos operables y optimizar sus niveles de procesamiento.

En este capítulo, se definirán los conceptos de Bibliometría, Cienciometría e Informetría; se explicará cada fase de la metodología ViBlioSOM®, sus objetivos, ventajas y desventajas.

3.1. Definiciones básicas.

En la actualidad, el análisis de información es parte esencial de los procesos de inteligencia empresarial, vigilancia científico-tecnológica, gestión del conocimiento y evaluación de proyectos; en nuestro caso de estudio, el análisis de la información está centrado en el comportamiento de la actividad científica así como la necesidad de realizar una vigilancia científico-tecnológica, de tal manera que se puedan identificar líneas tecnológicas, tecnologías emergentes o en declive, e incluso las distintas líneas de investigación.

Una forma de llevar a cabo este análisis es a través de los indicadores *Bibliométricos* y *Cienciométricos*, englobados en un campo más amplio llamado *Informetría*

3.1.1. Bibliometría y Patentometría.

Un método de análisis y medición de esos documentos es la Bibliometría. El concepto de **Bibliometría** ofrecido por Spinak es:

> "El estudio de los aspectos cuantitativos de la producción, diseminación y uso de la información registrada, a cuyo efecto desarrolla modelos y medidas matemáticas que sirven para hacer pronósticos y tomar decisiones en torno a tales procesos." *(Spinak, E., 1996.)* [Spi]

Otros especialistas, particularizando su uso en las actividades bibliotecarias, la definen como:

> "la aplicación de métodos matemáticos y estadísticos al estudio del uso que se hace de los documentos dentro de los sistemas de bibliotecas y entre ellos." *(Macías Chapula, CA., 2003.)* [MC03]

Como consecuencia del desarrollo de la ciencia y la tecnología, se amplió el alcance de las disciplinas métricas a otros campos del quehacer científico. Un ejemplo es la aplicación de las técnicas métricas a la información de patentes conocida como **Patentomería**. La importancia estratégica de las patentes como fuente de información, produjo la aparición de distintos indicadores para analizar este tipo de documento, fundamentalmente para la búsqueda de oportunidades tecnológicas, así como para la evaluación de programas de investigación y desarrollo.

Conceptualmente, el término Patentometría ha sido poco abordado, una de las escasas definiciones existentes fue encontrada en el sitio web de la RAND (Research and Development) en el 2001. En dicho sitio aparecía definida como el método de evaluación asociado con la identificación de las fortalezas y debilidades de la ciencia y la tecnología, a partir de los registros de invenciones e innovaciones provenientes de un país, institución o temática determinada. La Patentometría puede aparecer además como "bibliometría de patentes".

3.1.2. Cienciometría.

La **Cienciometría** utiliza métodos matemáticos para el estudio de la ciencia y de la actividad científica en general, además de medir el nivel de desarrollo y el aporte de la ciencia a las diferentes esferas de la sociedad. A pesar de la existencia de las distintas disciplinas métricas, surgidas como materias instrumentales de otras ciencias, el término Cienciometría se ha generalizado para la denominación de los estudios de esta índole.

> "La Cienciometría estudia los aspectos cuantitativos de la ciencia como disciplina o actividad económica, forma parte de la sociología de la ciencia y encuentra aplicación en el establecimiento de las políticas científicas, donde incluye entre otras las de publicación" *(Arencibia, Ricardo J. y Araújo, Juan A., 2002.)* [AA02]

3.1.3. Informetría.

La **Informetría** estudia los aspectos cuantitativos de la información en cualquier forma, no sólo la compilada en registros bibliográficos, y abarca cualquier grupo social por lo que no se limita sólo al científico como lo hace la Cienciometría. Puede incorporar, utilizar y ampliar los diversos estudios de evaluación de la información que se encuentra fuera de la Bibliometría y de la Cienciometría. La Informetría tiene como objetivo, aumentar la eficiencia de la recuperación de información, así como identificar estructuras y relaciones dentro de los diversos sistemas de información.

Además de las investigaciones de la Bibliometría y la Cienciometría, comprende asuntos como el desarrollo de modelos teóricos y las medidas de información, para hallar regularidades

en los datos asociados con la producción y el uso de la información registrada; abarca la medición de aspectos de la información, el almacenamiento y su recuperación, por lo que incluye la teoría matemática y la modulación.

La Informetría se aplica a varias áreas, entre las que se pueden señalar:

- Las características de la productividad de los autores, medida por la cantidad de documentos publicados en un tiempo determinado o por su grado de colaboración.
- Las características de las fuentes donde se publican los documentos, incluida su distribución por disciplinas.
- Los análisis de citas, según distribución por autores, tipo de documento, instituciones o países.
- El uso de la información registrada a partir de su demanda y circulación.
- La obsolescencia de la literatura mediante la medición de su uso y de la frecuencia con que se cita.
- El incremento de la literatura por temas.
- La distribución idiomática según la disciplina o el área estudiada.

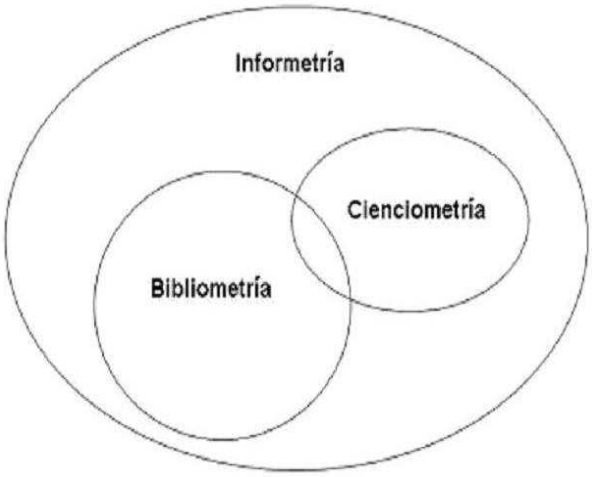


Figura 3.1: La Informetría es un campo más general donde encontramos a la Bibliometría y la Cienciometría.

Es así como el constante crecimiento de la información y de los conocimientos reflejados en publicaciones científicas y técnicas (artículos, patentes, etc.), impone nuevos requerimientos marcados por la impronta de las nuevas tecnologías de la información; a través de la Bibliometría y la Cienciometría se afronta de manera cada vez más efectiva este reto.

En el contexto actual, uno de sus “modos de realización”más prometedor es KDD, que además de apoyarse en técnicas ya establecidas como las que ofrece la estadística, suma los modernos enfoques del análisis multivariado tales como las redes neuronales.

3.2. ViBlioSOM®.

En la introducción del presente capítulo, mencionamos lo complicado que es tener acceso a herramientas eficientes y *ad-hoc* con las distintas problemáticas a las que el proceso KDD se enfrenta. Como consecuencia, los investigadores han adaptado los recursos con los que cuentan para crear nuevas *metodologías* que respondan a sus necesidades y cumplan con sus objetivos.

Hacia el año 2002, el grupo del Laboratorio de Dinámica No Lineal de la Facultad de Ciencias de la UNAM (LDNL) en colaboración con un grupo del Instituto Finlay de la Habana, propusieron en [SGC02] una metodología para el análisis y evaluación de información proveniente de bases de datos científico-tecnológicas, mediante la aplicación de modernas técnicas bibliométricas y de visualización a través de la red neuronal SOM.

ViBlioSOM® es una metodología abierta, basada en la utilización secuencial de varios sistemas comerciales de software; ViBlioSOM® se ha concebido como un proceso iterativo que modela cada etapa del proceso KDD, donde algunos de estos sistemas de software se utilizan en el preprocesamiento de los datos y otros llevan a cabo el análisis y visualización de los datos.

> “El objetivo fundamental de ViBlioSOM®, es contar con una guía o modelo de referencia que permita estandarizar los procesos de descubrimiento de conocimiento, de forma tal que se puedan validar los datos de entrada con sus resultados de salida, acortar el tiempo requerido para el análisis de los datos y automatizar el proceso.” *(H. Carrillo et al., 2002)* [SGC02]

En [CGMdlE+05], se toma como objeto de estudio una de las bases de datos bibliográficos disponibles en una de las más importantes instituciones en el mundo; el *Centro Nacional de Información en Biotecnología, (National Center for Biotechnology Information, NCBI)*. El NCBI ofrece MedLine.

MedLine es una base de datos bibliográficos producida por la NLM. Contiene aproximadamente 10 millones de registros, 15 millones de citas bibliográficas que provienen de más de 4,600 revistas que cubren los temas de la medicina, biomedicina, enfermería, odontología, oncología, medicina veterinaria, salud pública, ciencias preclínicas y otras áreas de las ciencias de la vida.

En el año 2001, se inicia una etapa de aplicación de ViBlioSOM® a diferentes problemáticas vinculadas con la biotecnología, la agricultura, la sociología, etc; motivo por el cual se recurrió al uso de MedLine. Como parte de esta expansión, se amplía la colaboración con otras instituciones científicas de Cuba y del mundo, además de que se profundiza en la investigación de la red SOM para los fines específicos de la Bibliometría así como de la minería de datos y textos.

Este último motivo, unido a la búsqueda de nuevas aplicaciones, propició el surgimiento de la colaboración con el Laboratorio de Dinámica No Lineal de la Facultad de Ciencias de la

UNAM. Uno de los primeros trabajos en conjunto estuvo enfocado al estudio de las aplicaciones de la dinámica no lineal a la biomedicina.

MedLine fue elegida de entre otras bases de datos internacionales, a causa de su carácter especializado en ciencias biomédicas, el amplio procesamiento que realiza de la literatura en la rama, la utilización de rigurosos criterios selectivos en la elección de las publicaciones y trabajos a registrar, su alta difusión y popularidad mundial y del interés mostrado por esta base de datos en la recolección de la literatura más representativa publicada por la región latino-americana (cobertura).

ViBlioSOM® tiene un vínculo muy estrecho con los procesos de inteligencia empresarial, vigilancia científico-tecnológica, gestión del conocimiento y evaluación de proyectos. Al mismo tiempo, puede ser aplicado en servicios bibliotecarios e informativos y en observatorios de ciencia y tecnología. Todo depende de la problemática concreta y del indicador que se aplique.

3.2.1. Fases de la metodología ViBlioSOM®.

Como mencionamos anteriormente, a través de indicadores bibliométricos y sistemas de software especializados, ViBlioSOM® lleva a cabo las etapas básicas del proceso KDD; sin embargo, los autores de [CGMdlE+05] proponen para ViBlioSOM® fases específicas.

3.2.1.1. Comprensión del campo de aplicación.

Es importante desarrollar un completo entendimiento en el campo de aplicación; es importante trabajar en unión con especialistas de la rama del conocimiento a la cual se asocia la problemática, si es posible se deben identificar a las personas claves que liderean los temas de investigación, desarrollo o innovación.

3.2.1.2. Adquisición y selección de archivos.

Los datos son obtenidos de MedLine por medio de peticiones desde su sitio en Internet[1]. La búsqueda es llevada a cabo por el sistema de recuperación de información *Entrez PubMed*, que básicamente se encarga de encontrar coincidencias de términos o frases que son ingresados en los cuadros de búsqueda de dicha página.

Una vez que se realiza la consulta, el usuario elige el conjunto de datos apropiados. MedLine permite almacenar los resultados en archivos con distintas opciones de formato: `XML`, `txt`, formato MedLine, entre otros. véase la figura 3.2. Para los usuarios de ViBlioSOM®, es necesario que los resultados sean almacenados en archivo con formato de texto plano (`txt`).

3.2.1.3. Preprocesamiento.

Esta fase consiste en la depuración de los datos almacenados en el archivo obtenido en la fase anterior, así como su preprocesamiento. Se realizan diferentes operaciones para eliminar datos

[1]Para mayor referencia consulte `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi`

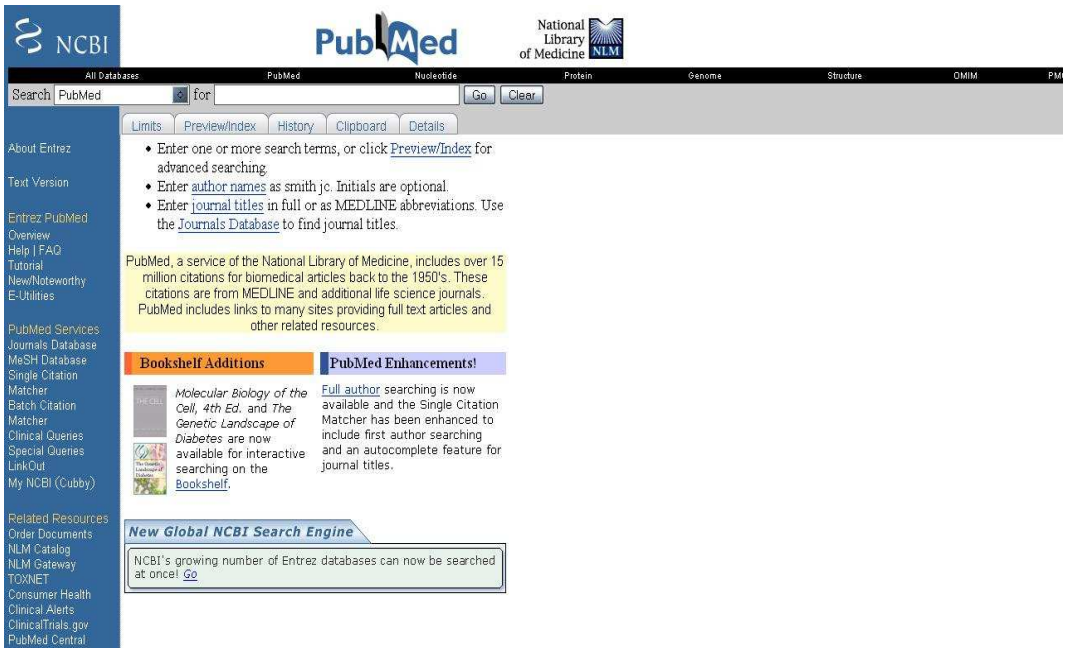


Figura 3.2: Página principal del sitio de Entrez PubMed.

espurios si es requerido, colectar la información necesaria para el modelo o registro del ruido, decidir estrategias para el manejo de campos faltantes en los datos, registrar la información de secuencias en el tiempo y cambios conocidos.

Esta tarea se lleva a cabo exportando todos los registros contenidos en el archivo obtenido desde MedLine, al sistema ProCite. **ProCite** fue creado por el Instituto de Información Científica de Filadelfia (Institute for Scientific Information of Philadelphia); es un gestor de bases de datos de referencias bibliográficas, capaz de conectarse a través de la red a distintas bases de datos y obtener registros por medio del sistema. Además, permite cambiar el formato del archivo original, a otro que sea manipulado con facilidad por alguna otra herrmienta de software.

ProCite puede conectarse a traves de Internet con alrededor de 200 bibliotecas y permite importar registros almacenados en archivos de texto de otras bases de datos (comerciales) u otras fuentes.

ProCite utiliza el protocolo ANSI **Z39.50**, que es un protocolo cliente-servidor para el intercambio de informacion entre un cliente y multiples bases de datos bibliográficas. Estados Unidos ha establecido este protocolo como estandar para el intercambio de información de bases de datos bibliográficas y es mantenido por la Biblioteca del Congreso. Z39.50 permite la obtención de conjuntos de resultados obtenidos en distintas bases de datos consultadas de forma distribuida.

ProCite permite realizar la tarea de preprocesamiento de forma eficaz: se pueden seleccionar los campos específicos, para llevar a cabo cambios globales o locales del contenido de los

```
PMID- 15575155
OWN - NLM
STAT- MEDLINE
DA  - 20041203
DCOM- 20050105
PUBM- Print
IS  - 1057-7149
VI  - 13
IP  - 12
DP  - 2004 Dec
TI  - Locally optimum nonlinearities for DCT watermark detection.
PG  - 1604-17
AB  - The issue of copyright protection of digital multimedia data has attracted
      a lot of attention during the last decade. An efficient copyright
      protection method that has been gaining popularity is watermarking, i.e.,
      the embedding of a signature in a digital document that can be detected
      only by its rightful owner. Watermarks are usually blindly detected using
      correlating structures, which would be optimal in the case of Gaussian
AD  - Department of Electrical and Computer Engineering, Beckman Institute,
      University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA.
      briassou@vision.ai.uiuc.edu
FAU - Briassouli, Alexia
AU  - Briassouli A
FAU - Strintzis, Michael G
AU  - Strintzis MG
LA  - eng
PT  - Evaluation Studies
PT  - Journal Article
PT  - Validation Studies
PL  - United States
TA  - IEEE Trans Image Process
JID - 9886191
SB  - IM
MH  - *Algorithms
MH  - *Computer Graphics
MH  - Computer Security
MH  - Hypermedia
MH  - Image Interpretation, Computer-Assisted/*methods
MH  - Information Storage and Retrieval/*methods
MH  - Models, Statistical
MH  - *Nonlinear Dynamics
MH  - *Patents
MH  - Pattern Recognition, Automated/*methods
MH  - Product Labeling/methods
MH  - Reproducibility of Results
MH  - Research Support, Non-U.S. Gov't
MH  - Sensitivity and Specificity
MH  - Signal Processing, Computer-Assisted
EDAT- 2004/12/04 09:00
MHDA- 2005/01/06 09:00
PST - ppublish
SO  - IEEE Trans Image Process 2004 Dec;13(12):1604-17.
```

Figura 3.3: Detalle de los registros obtenidos de MedLine en formato de texto plano.

registros, eliminar inconsistencias o duplicidades, importar los resultados de búsquedas realizadas en diferentes bases de datos bibliográficas y crear todas las bases de datos de diversos temas que se deseen. Una de las características más importantes de Procite, es que permite eliminar duplicados al momento en que realiza la búsqueda en distintas bases de datos, donde el usuario determina qué campos debe comparar.

El usuario puede definir cuáles términos están o no duplicados y decidir eliminarlos o no; Procite permite realizar búsquedas dentro de los diferentes campos, por medio de distintas funciones y operadores relacionales: equals(=), not equal(<>), greater than(<), less than(<), begins with, ends with, exactly, contains, empty, not empty.

Estas funciones permiten al usuario establecer distintos criterios de búsqueda y selección de datos; al mismo tiempo, el usuario puede guardar estos criterios y utilizarlos en búsquedas posteriores.

En ProCite se pueden realizar conteos utilizando los principales campos de la base de datos o utilizando los campos definidos por el usuario. De esta manera se pueden desarrollar diferentes tipos de investigaciones como por ejemplo: conocer cuál es el autor más productivo, las materias tratadas con mayor frecuencia, las revistas que más publican sobre algún tema, producción por

países, tipos de artículos, etc.

Es por esto que ProCite se considera una herramienta muy útil y poderosa para realizar estudios métricos en diferentes especialidades y ramas de la ciencia, así como para el diseño y prestación de servicios de alto valor agregado. En el caso de ViBlioSOM®, las funciones de ProCite han podido adaptarse como una herramienta de alto impacto en la fase de preprocesamiento, así como para aplicar algunos indicadores básicos.

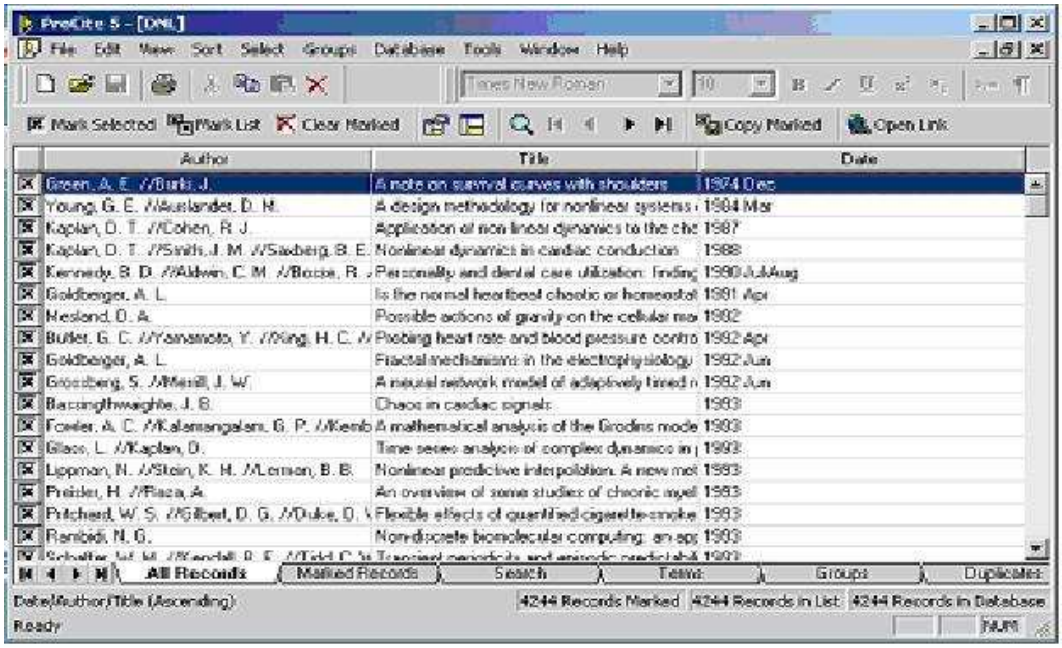


Figura 3.4: Vista de la interfaz de usuario de Procite.

Dentro de la fase de preprocesamiento se realizan las operaciones de procesamiento y transformación de datos. En esta parte del proceso, se llevan a cabo una serie de transformaciones de los datos para ponerlos en un formato, en caso de ser requerido, para el algoritmo de minería de datos. Una vez finalizada la depuración de los datos, es necesario cambiar el formato del archivo a un formato que sea compatible con **Excel** (al finalizar el preprocesamiento, se crea un archivo de texto que contiene los cambios realizados en ProCite).

Las transformaciones de los datos se realizan por medio de una *macro* para Excel llamada **ToolInf**. ToolInf consta de cinco opciones, cada una incluye diferentes variantes para realizar la operación que representa. Las opciones son:

1. Conteo de datos.

2. Identificación de datos.

3. Clasificación de registros

4. Creación de matrices.

5. Ayuda del sistema.

Los datos son exportados a Excel desde el archivo previamente procesado en ProCite; el objetivo es crear una respresentación matricial de los datos, transformados por alguna de las opciones de ToolInf. Los resultados se almacenan en nuevas hojas de cálculo o en hojas ya creadas, dependiendo de lo especificado.

3.2.1.4. Minería de Datos y visualización de los resultados.

Ambas fases se realizan con el ya mencionado sistema de software Viscovery® SOMine®, que provee de medios poderosos para analizar conjuntos de datos con una estructura compleja, sin necesidad de contar a priori con algún tipo de información estadística. La fuente de datos de entrada que requiere Viscovery® SOMine®, es un archivo que contenga una tabla de datos numéricos, ya sea en formato de texto o de Excel.

Minería de Datos.

El algoritmo de minería de datos que utiliza Viscovery® SOMine®, está basado en la variante Batch Map de la red SOM. La ejecución del proceso de entrenamiento, depende tanto de la determinación de los parámetros básicos del mapa: número de nodos en la retícula, razón del mapa y tensión; como de los parámetros que determinan la manera en que la retícula cambia a lo largo del proceso de entrenamiento, como son el factor de escalamiento, la altura del mapa inicial y la configuración de un vector de parámetros de entrenamiento (también llamada cédula de entrenamiento).

Durante el entrenamiento, Viscovery® SOMine® permite observar la evolución del proceso mostrando las gráficas de error de cuantización y distorsión normalizada e indica la duración estimada del entrenamiento.

Visualización de resultados.

Concluido el proceso de entrenamiento, Viscovery® SOMine® despliega una serie de mapas y visualizaciones. Al mismo tiempo permite al usuario elegir entre distintos algoritmos de visualización para los mapas. Viscovery® SOMine® ofrece los siguientes algoritmos de visualización:

- Mapas de Componentes.
- Conglomeraciones Ward.
- Conglomeraciones SOM-Ward.
- Conglomeraciones SOM con frontera Single Linkage.
- Proyección de los Datos.

- U-Matrix.

Posteriormente el usuario dará una interpretación del resultado obtenido, a partir de la elección de uno o distintos algoritmos de visualización.

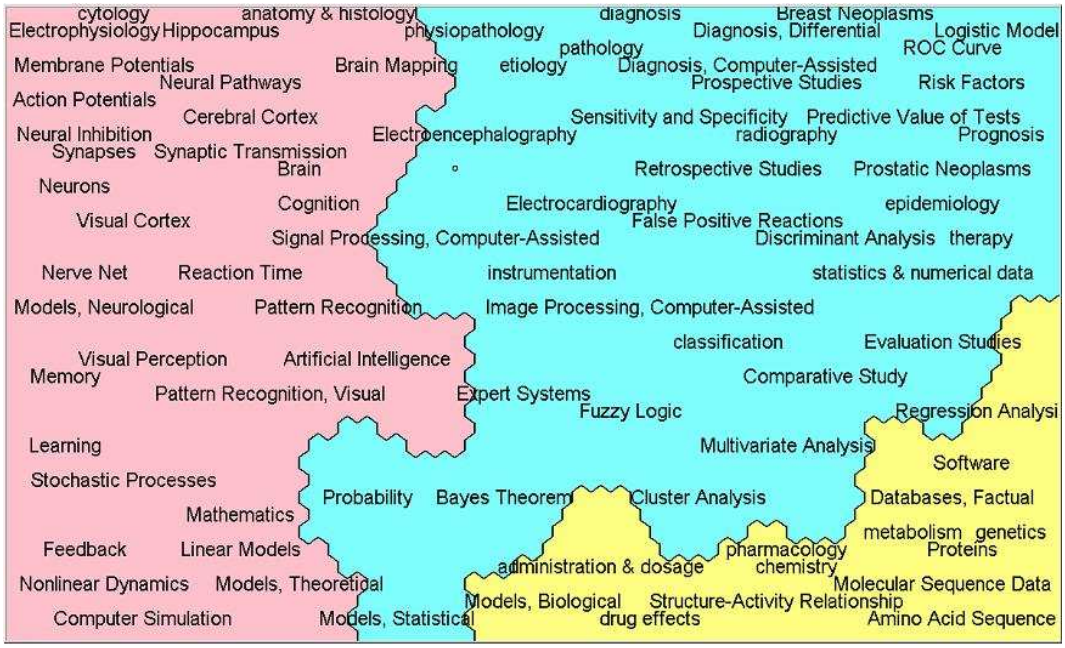


Figura 3.5: Mapa generado por el sistema Viscovery® SOMine®.

Una vez generados los mapas, el usuario plantea diversos criterios para elaborar una o varias interpretaciones acerca del problema. Cabe recordar, que si bien una representación gráfica de un conjunto de datos es útil para lograr un entendimiento intuitivo, esta representación no otorga elementos suficientes para establecer una condición de veracidad del mapa, por ello, la interpretación del experto es crucial para validar el proceso.

3.2.2. Ventajas y desventajas de ViBlioSOM®.

ViBlioSOM® ha podido sustituir la carencia de herramientas altamente sofisticadas, por una secuencia de fases que permite al usuario adaptar a la metodología su campo de aplicación, cualquiera que fuese y obtener resultados de importancia.

En la primera parte de este trabajo, mencionamos la importancia de obtener una representación gráfica de los datos que estamos analizando; una herramienta como Viscovery® SOMine® para la visualización de estos datos, permite extraer información valiosa, partiendo de la inspección visual de una amplia gama de mapas, a pesar de que no es posible establecer una metodología general para su exploración.

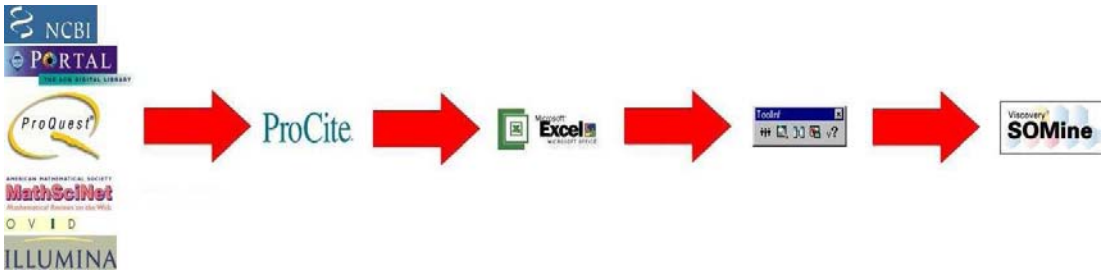


Figura 3.6: Secuencia del proceso ViBlioSOM®.

Si bien ViBlioSOM® ha aportado una nueva forma de analizar bancos de información, al mismo tiempo presenta ciertas desventajas en su uso.

ProCite permite depurar y realizar cambios globales sobre los registros que se desean analizar, esta fase puede llevarse a cabo de forma manual o automática. De forma manual, el usuario es quien busca en cada registro el término a modificar. Sin embargo, el usuario tiene la opción de automatizar esta tarea, como sería por ejemplo, importar un archivo de *Tesauro*.

Una de las principales limitaciones de ViBlioSOM® la encontramos dentro del procesamiento y transformación de los datos; los datos que se analizan pertencen a un espacio multidimensional y la dimensión de cada registro es desconocida. En Excel no es posible tener más de 250 columnas por registro, por lo que sólo podemos analizar aquellos datos cuya dimensión sea menor a 250. Esta limitante abre la posibilidad de una pérdida de información considerable.

El hecho de utilizar distintos sistemas de software para cada fase del proceso, significa modificar una y otra vez los archivos obtenidos, para adapatarlos al formato de entrada para iniciar la siguiente fase. Cada sistema utilizado por ViBlioSOM® es software de tipo comercial, por lo que es necesario realizar una inversión considerable en la compra de las licencias de cada sistema.

Para la comprensión de las distintas opciones que ofrece cada sistema y al mismo tiempo para su uso, el usuario requiere de un entrenamiento previo y en ocasiones prolongado.

Debido a esto surge la necesidad de crear una herramienta propia que integre cada fase de ViBlioSOM®, que reúna las características básicas que componen a cada uno de estos sistemas, así como de disminuir el tiempo de entrenamiento que el usuario requiere para el uso del mismo; de esta manera, se plantea el desarrollo de **Data SOMinning**.

Capítulo 4

Diseño de una suite para la Minería de Datos: Data SOMinning

Con el objetivo de solucionar las limitaciones mencionadas en el capítulo anterior, en el presente trabajo se ha diseñado y desarrollado un sistema de software que perfeccione e integre las características básicas de los sistemas utilizados por ViBlioSOM®. Este sistema es de gran utilidad para la gestión eficiente de la ciencia y la tecnología, la gestión de proyectos y para la toma de decisiones en instituciones de Investigación + Desarrollo (I+D). Este tipo de sistema puede servir a la comunidad científica en general, en la medida que permite el procesamiento automático de datos bibliográficos digitales de una de las principales bases de datos en información biomédica: *MedLine*.

Se usó TSP′i (Team Software Process) para implementar el sistema **Data SOMinning** tomando en cuenta las características de modularidad, incrementabilidad, funcionalidad y mantenibilidad. También se utilizó el paradigma orientado a objetos y el lenguaje C# para construir el sistema; la plataforma que se utilizó para implementar es Visual Studio .NET.

En este capítulo, se describirá el diseño de Data SOMinning así como los diagramas generados en el Lenguaje de Modelado Unificado (Unified Modeling Language, UML) para el sistema.

4.1. Data SOMinning: sistema de software para la Minería de Datos.

TSP′i (Team Software Process), es una técnica aplicable al desarrollo de software que provee un balance entre proceso, producto y equipo de trabajo. Un proceso TSP′i es un estándar para dar soluciones a problemas de software, y está constituido por siete fases. La unión de todas estas fases forma un ciclo, existiendo la posibilidad de realizar *n* ciclos para el desarrollo de un producto. El objetivo de cada ciclo es establecer el tamaño y el contenido del producto de software. Las siete fases son:

- Fase de Lanzamiento.

- Fase de Estrategia.
- Fase de Planificación.
- Fase de Requerimientos.
- Fase de Diseño.
- Fase de Implementación.
- Fase de Prueba.
- Fase de Postmortem.

Para el objetivo de este trabajo nos esfocamos particularmente a las fases que contemplan el desarrollo del software más que a la gestión del mismo. En el resto de este capítulo se describen algunos de los resultados obtenidos en dichas fases. Recordemos que nuestro equipo fue conformado por dos personas, a las cuales se les asignaron tareas desde el inicio del proyecto.

Con base en las definiciones teóricas, los antecedentes y el contexto actual citados a lo largo de este trabajo, hemos de determinar de forma precisa nuestro objetivo. Una vez precisado nuestra meta, mostraremos el diseño conceptual del sistema, sus requerimientos y explicaremos detalladamente las características de cada fase del mismo, así como la especificación del lenguaje de programación, arquitectura del sistema y los diagramas UML generados.

Objetivos:

- Desarrollar con base en los fundamentos teóricos del proceso KDD y la metodología ViBlioSOM®, una herramienta de minería de datos con un enfoque neurocomputacional que permita realizar análisis bibliométrico.
- Integrar todos y cada uno de los módulos de la metodología ViBlioSOM®, en un sistema de software que llamamos **Data SOMinning**.

4.1.1. Descripción de las necesidades del sistema.

Dados los elementos teóricos suficientes y las experiencias del usuario en el uso de la metodología ViBlioSOM®, el sistema debe cumplir las siguientes características:

1. Recuperar y mostrar en pantalla, los registros obtenidos de la búsqueda realizada en la base de datos MedLine. Estos registros se encuentran almacenados en archivo, bajo el formato de texto plano.
2. Normalizar los registros obtenidos de forma manual y por medio de un archivo de tesauro.
3. La fase de procesamiento deberá implementar al menos las operaciones que integran la macro para Excel llamada ToolInf.

4. Las matrices obtenidas del procesamiento de los datos serán visibles para el usuario en la misma interfaz.

5. Los parámetros para el entrenamiento de la red SOM, están basados en los definidos en el sistema Viscovery® SOMine®.

6. La retícula de la red SOM deberá definirse como una retícula hexagonal.

7. Los parámetros para el entrenamiento de la red SOM podrán ser definidos por el usuario.

8. Incluir los algoritmos de visualización, más comúnmente usados, permitidos en el sistema Viscovery® SOMine®.

9. Almacenar y recuperar las operaciones realizadas por el usuario, en un ambiente tipo escritorio.

Dado que el objetivo principal de este sistema es proveer de una amplia gama de funciones bajo una misma interfaz, los componentes visuales de ésta son un factor importante para el desarrollo. Entonces, nos dirigimos a proponer el siguiente diseño conceptual:

1. Se propone utilizar los recursos que ofrece la plataforma de desarrollo *Microsoft Visual Studio .NET*.

2. Implementar el sistema bajo el lenguaje de programación orientado a objetos *C#*.

3. Hacer uso de la interfaz contenida en *.NET Framework* llamada *GDI+* para generar las salidas gráficas.

4. Uso de *UML* para modelar los elementos que conforman el sistema mediante la herramienta *Rational Rose*.

5. Uso de *NDoc* para generar la documentación en formato de ayuda de Visual Studio .NET, a partir de los archivos `XML` generados por el compilador de *C#*.

6. Definir el sistema como un conjunto de componentes separados en tres partes:

   - Componentes visuales. (Interfaces)
   - Componentes de dominio de problema. (Algoritmos)
   - Componentes de manejo y control del flujo de datos. (Capa de Persistencia)

El sistema se desarrolló en 2 ciclos incrementales, en el primer ciclo se integraron las funciones básicas primordiales, con base en la especificación de las necesidades del sistema y en el segundo se optimizaron dichas funciones. De esta manera se estableció la funcionalidad básica:

1. Adquisición de datos.

2. Procesamiento de datos.

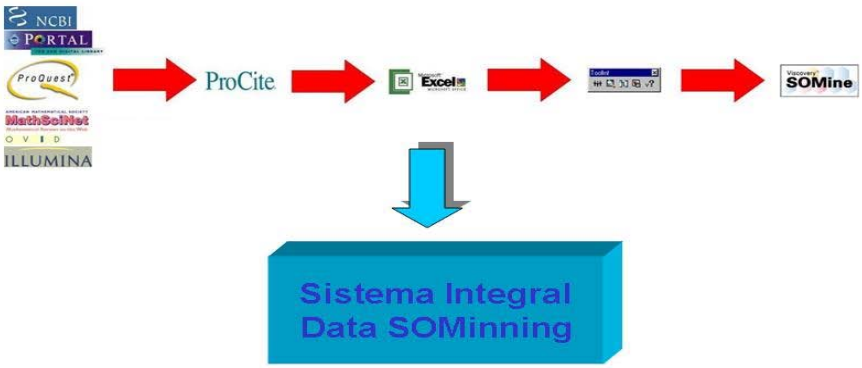


Figura 4.1: Integración de la metodología ViBlioSOM® a una suite para Minería de Datos.

3. Transformaciones.
4. Entrenamiento SOM.
5. Visualización.

La especificación de requerimientos incluyó realizar un prototipo de interfaz con el usuario de manera sencilla, que inclusive no tuvo alguna funcionalidad, sin embargo estas sí fueron establecidas.

4.1.2. Diagramas de Caso de Uso.

Una vez establecidas las funcionalidades básicas para el primer ciclo, se definieron los casos de uso que representan cada una de las funciones establecidas.

4.1.2.1. Caso de Uso: Adquisición de datos.

Figura 4.2: Diagrama de Caso de Uso para Adquisición de datos.

	Usuario	**Sistema**	**Ex.**
Adquisición de datos.	El usuario selecciona de un componente comboBox, la base de datos MeSH.	Habilita un componente TextBox, para el ingreso de la frase de búsqueda.	
	El usuario proporciona en un componente TextBox, la frase de búsqueda.	Habilita un componente Button, para poder iniciar la búsqueda.	
	El usuario inicia por medio de un componente Button, la búsqueda deseada.	Muestra en un componente CheckedListBox los términos relacionados a la frase indicado.	**E1**
E1	No hay términos relacionados respecto a la frase introducida por el usuario en el componente TextBox.	Indica al usuario que debe reingresar una nueva frase de búsqueda.	

	Usuario	**Sistema**	**Ex.**
Adquisición de datos.	El usuario selecciona de un componente comboBox, la base de datos PubMed.	Habilita un componente TextBox, para el ingreso del término de búsqueda.	
		Habilita diversos componentes, para el ingreso de límites de búsqueda.	
	El usuario proporciona en un componente TextBox, el término de búsqueda. Opcionalmente proporciona diversos limitadores de búsqueda.	Habilita un componente Button, para poder iniciar la búsqueda.	
	El usuario inicia por medio de un componente Button, la búsqueda deseada.	Muestra en un componente DataGrid los registros encontrados de la búsqueda.	**E1**
E1	No hay registros relacionados respecto al término introducido por el usuario en el componente TextBox.	Indica al usuario que debe reingresar un nuevo término de búsqueda.	

4.1.2.2. Caso de Uso: Selección de términos.

Figura 4.3: Diagrama de Caso de Uso para Selección de términos.

	Usuario	**Sistema**	**Ex.**
Selección de términos.	El usuario selecciona de un componente CheckedListBox, el(los) término(s).	Muestra en un componente Panel, la descripción de el(los) término(s) seleccionado(s).	
	El usuario selecciona por medio de un componente Button, el(los) término(s) seleccionado(s).	Muestra en un componente DataGrid los artículos asociados a el(los) término(s).	**E1**
	El usuario selecciona de un componente ComboBox, el operador de búsqueda.	Asigna el operador de búsqueda.	
E1	No hay términos seleccionados en el componente CheckedListBox.	Indica al usuario que debe seleccionar al menos un término.	

4.1.2.3. Caso de Uso: Procesamiento de datos.

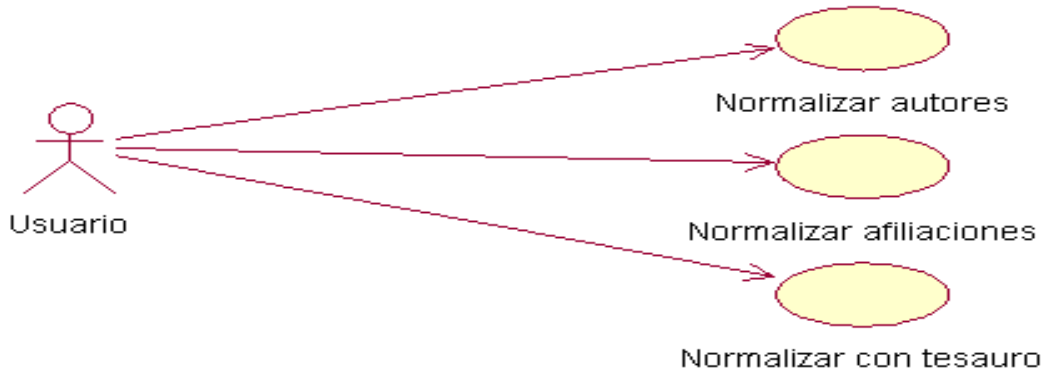


Figura 4.4: Diagrama de Caso de Uso para Procesamiento de datos.

	Usuario	**Sistema**	**Ex.**
Procesamiento de datos.	Selecciona la opción de Procesamiento de datos.	Muestra en un componente DataGrid los registros encontrados de la consulta.	
	El usuario selecciona de un componente CheckedListBox, los autores o afiliaciones a normalizar.	Añade a un componente ListBox los elementos seleccionados.	
	Introduce a un componente TextBox, el valor de los elementos seleccionados y selecciona desde un componente Button la opción de cambiar.	Cambia el valor del o los elementos seleccionados en los componentes DataGrid y CheckedListBox.	**E1**
	El usuario selecciona de un componente CheckedListBox, el(los) tesauro(s) con ellos que normalizará los registros.	Cambia el valor del o los elementos que contiene el(los) archivo(s) de tesauro(s) en el componente DataGrid.	
	El usuario selecciona de un componente ComboBox, el tesauro a editar.	Muestra en un componente TreeView el contenido del archivo de tesauro seleccionado y actualiza los valores de los componentes DataGrid y CheckedListBox.	
E1	El usuario introduce valores no válidos.	Indica al usuario introducir valores permitidos.	

4.1.2.4. Caso de Uso: Transformaciones.

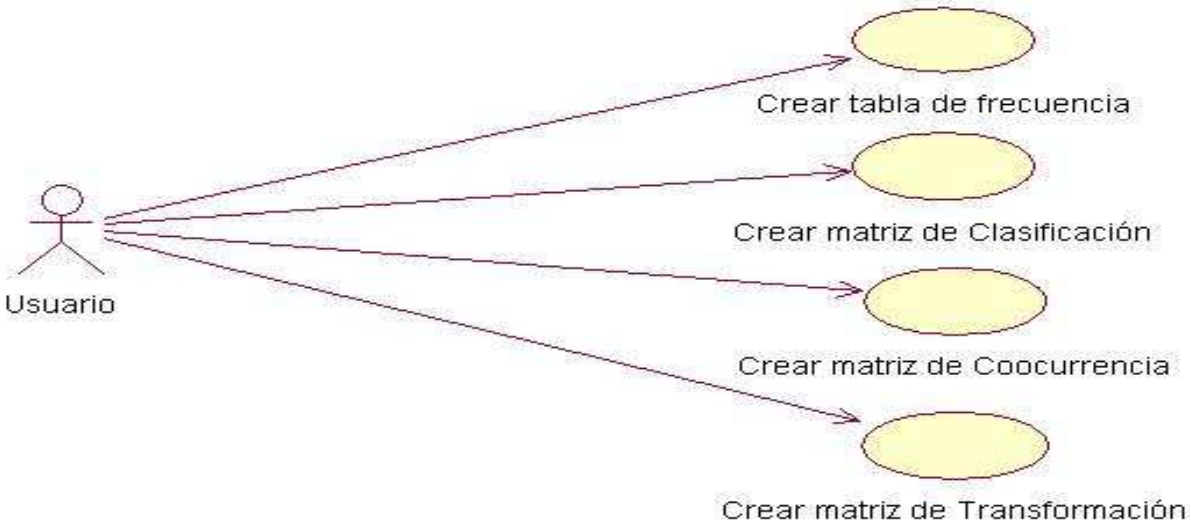


Figura 4.5: Diagrama de Caso de Uso para Transformaciones.

	Usuario	**Sistema**	**Ex.**
Transformaciones.	Selecciona la opción de Transformaciones.	Muestra en un componente comboBox, los campos posibles a partir de los cuales se pueden crear tablas de frecuencia.	
	El usuario selecciona de un componente comboBox, el campo para crear una tabla de frecuencia.	Habilita un componente Button, para poder iniciar la creación de la tabla de frecuencia.	
	El usuario inicia por medio de un componente Button, la creación de la tabla.	Calcula la tabla correspondiente.	
		Muestra en un componente DataGrid, la tabla obtenida.	
		Añade en un componente comboBox, el nombre de la tabla obtenida. Habilita dicho comboBox.	**E1**

	Usuario	Sistema	Ex.
	El usuario selecciona de un componente comboBox, la tabla de frecuencia para crear una matriz de clasificación.	Habilita un componente Button, para poder iniciar la creación de la matriz de clasificación.	
	El usuario inicia por medio de un componente Button, la creación de la matriz.	Calcula la matriz correspondiente.	
		Muestra en un componente DataGrid, la matriz obtenida.	
		Añade en dos componentes comboBox, el nombre de la matriz obtenida. Habilita dichos comboBox.	**E1**
	El usuario selecciona de un componente comboBox, la matriz de clasificación para crear una matriz de coocurrencia.	Habilita un componente Button, para poder iniciar la creación de la matriz de coocurrencia.	
	El usuario inicia por medio de un componente Button, la creación de la matriz.	Calcula la matriz correspondiente.	
		Muestra en un componente DataGrid, la matriz obtenida.	
		Añade en dos componentes comboBox, el nombre de la matriz obtenida. Habilita dichos comboBox.	**E1**
	El usuario selecciona de un componente comboBox, la matriz de coocurrencia para crear una matriz de transformación.	Habilita un componente Button, para poder iniciar la creación de la matriz de transformación.	
	El usuario inicia por medio de un componente Button, la creación de la matriz.	Calcula la matriz correspondiente.	
		Muestra en un componente DataGrid, la matriz obtenida.	
		Añade en un componente comboBox, el nombre de la matriz obtenida. Habilita dicho comboBox.	**E1**
E1	El usuario intenta crear nuevamente una tabla o matriz previamente calculada.	Muestra la tabla o matriz correspondiente.	

4.1.2.5. Caso de Uso: Entrenamiento SOM.

Figura 4.6: Diagrama de Caso de Uso para Entrenamiento SOM.

	Usuario	**Sistema**	**Ex.**
Entrenamiento SOM.	Selecciona la opción de Entrenamiento SOM.	Muestra el componente ComboBox con las matrices generadas en la fase de Transformaciones.	
	El usuario selecciona de un componente ComboBox, la matriz de entrada para la red SOM.	Añade a un componente ListBox los elementos seleccionados.	
	El usuario introduce en cada campo, las variables que intervienen en el entrenamiento de la red SOM.	Asigna las variables del entrenamiento de la red SOM.	**E1**
E1	El usuario introduce valores negativos o no válidos.	Indica al usuario introducir valores positivos o permitidos.	

4.1.2.6. Caso de Uso: Visualización.

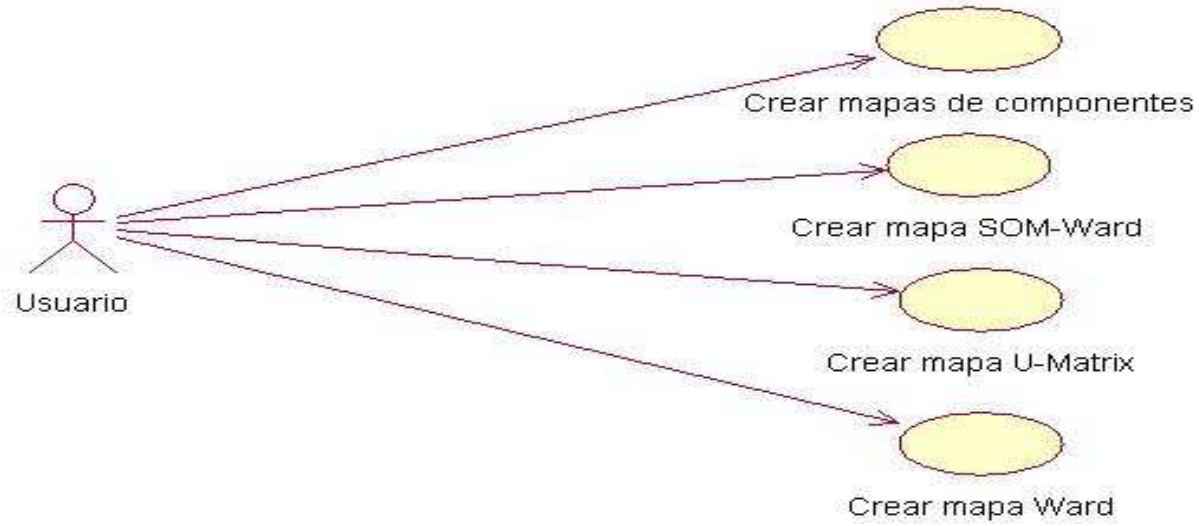


Figura 4.7: Diagrama de Caso de Uso para Visualización.

	Usuario	**Sistema**	**Ex.**
Visualización.	Selecciona la opción de Visualización.	Muestra los posibles algoritmos de visualización para los datos resultantes de un determinado entrenamiento.	
	El usuario selecciona de un componente Button, la construcción de los mapas de componentes.	Obtiene y muestra el mapa de componentes resultante.	**E1**
	El usuario selecciona de un componente comboBox, un tipo de algoritmo de visualización.	Habilita un componente comboBox, para el ingreso del tipo de métrica para el algoritmo de visualización.	

	Usuario	**Sistema**	**Ex.**
	El usuario selecciona de un componente comboBox, un tipo de métrica.	Habilita un componente comboBox, para el ingreso del tipo de frontera y el número de conglomerados para el algoritmo de visualización.	
		Habilita un componente button, para poder iniciar la creación del mapa correspondiente.	
	El usuario selecciona de un componente comboBox, un tipo de frontera.		
	El usuario selecciona de un componente comboBox, el número de conglomerados.		
	El usuario inicia por medio de un componente Button, la creación del mapa.	Obtiene y muestra el mapa U-Matriz o de conglomeración jerárquico resultante.	**E1**
E1	El usuario intenta crear nuevamente un mapa previamente calculado.	Muestra la visualización correspondiente calculado con anterioridad.	

4.2. Construcción del sistema.

4.2.1. Arquitectura del sistema.

Generados los diagramas de casos de uso, es necesario establecer la vista conceptual de los componentes que integran la estructura de la aplicación, es decir, definir una *arquitectura*. La arquitectura de las aplicaciones difieren según como está distribuido este código; para esta aplicación proponemos la **Arquitectura de Tres Capas**, véase la figura 4.8.

En este modelo una aplicación se convierte en un conjunto de 3 elementos con distintos fines:

- Capa de Interfaz Humana.
  Proporciona los componentes gráficos de la interfaz de usuario.
- Capa de Dominio de Problema.
  Es el puente entre un usuario y los servicios de datos. Responde a peticiones del usuario para ejecutar una tarea específica.
- Capa de Manejo de Datos.
  Esta capa es responsable de almacenar, recuperar, mantener y asegurar la integridad de los datos.

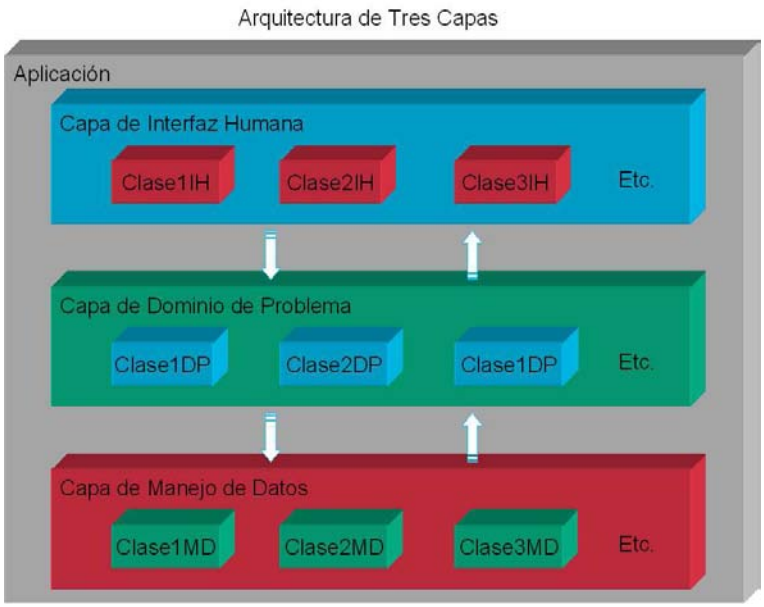


Figura 4.8: Esquema de la arquitectura de tres capas propuesta para el desarrollo de Data SO-Minning.

La separación de la aplicación en una arquitectura de tres capas, hace más fácil reemplazar o modificar una capa, sin afectar las capas y módulos restantes. Al mismo tiempo, las peticiones

realizadas desde la capa de interfaz humana a la de dominio de problema, son más flexibles en el sentido de que la interfaz humana sólo necesita transferir parámetros al dominio de problema.

4.2.2. Prototipo del sistema.

En esta sección, describimos el prototipo del sistema por medio de imágenes correspondientes a la interfaz gráfica.

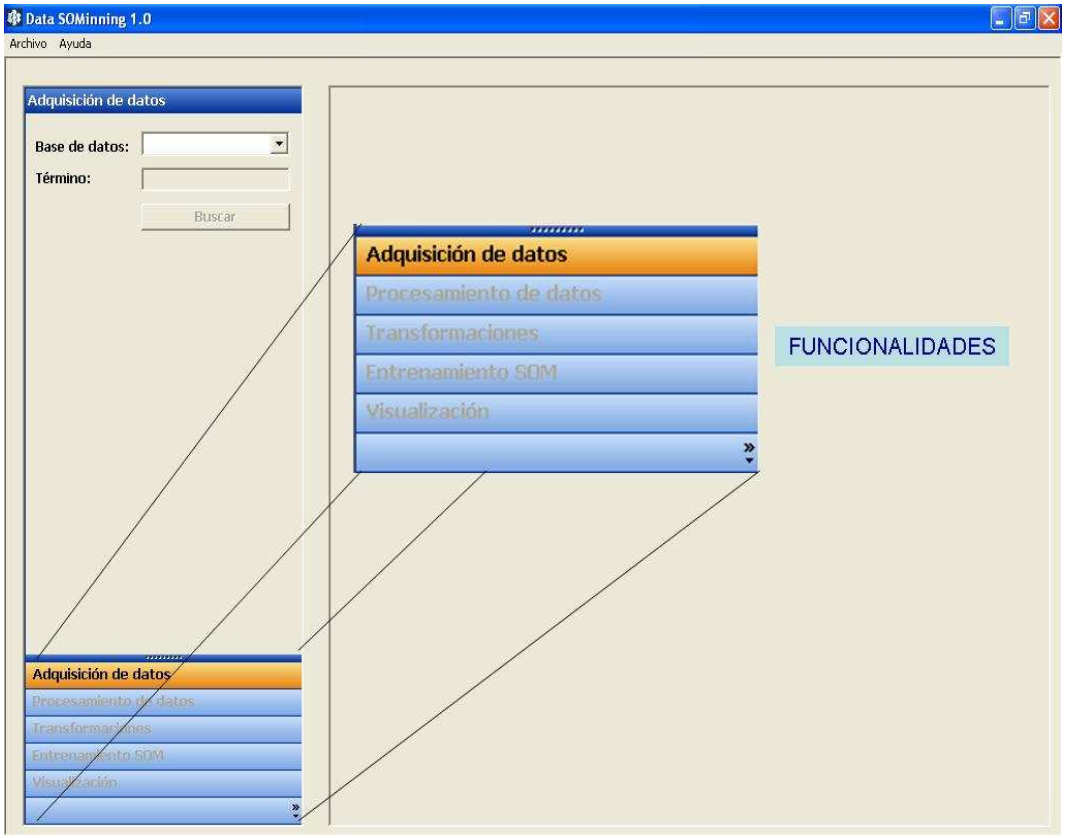


Figura 4.9: Pantalla principal de Data SOMinning.

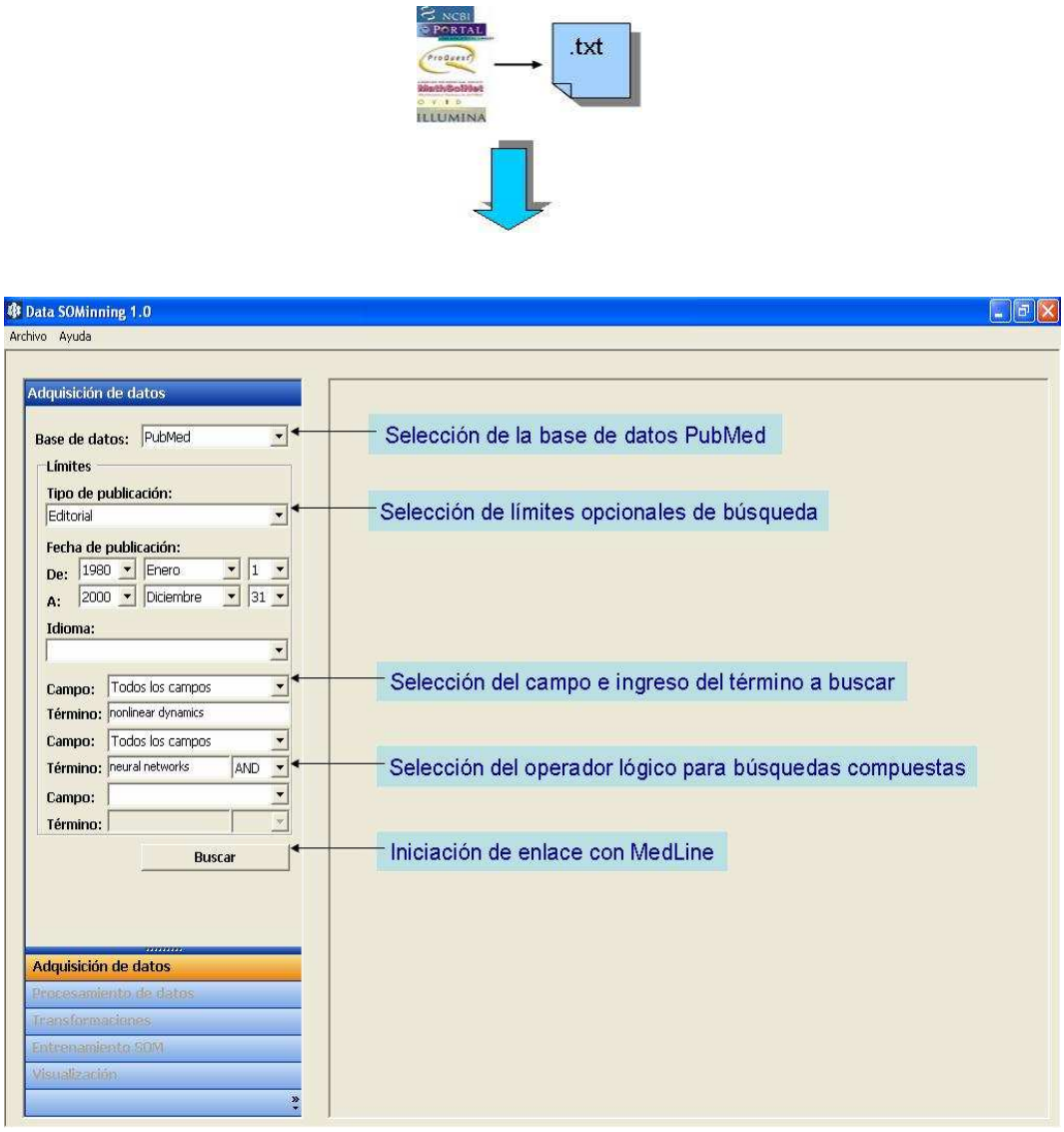


Figura 4.10: Vista de la interfaz de usuario de Data SOMinning para Adquisición de datos desde MeSH.

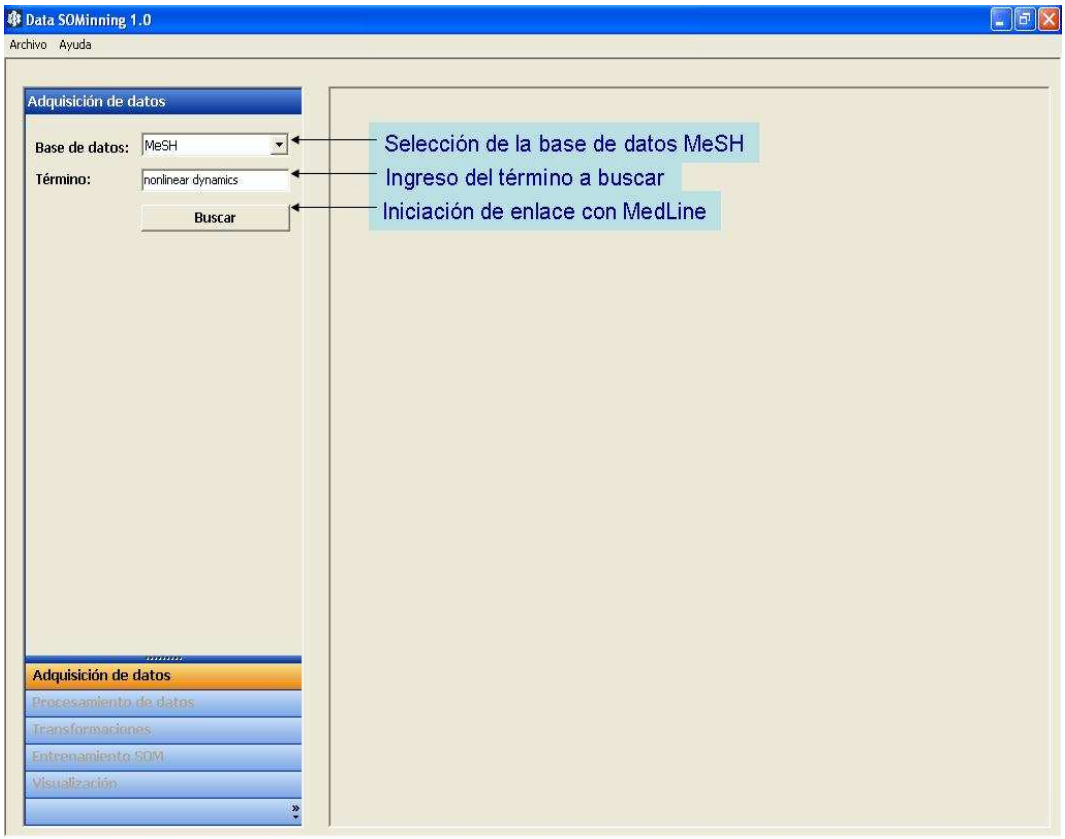


Figura 4.11: Vista de la interfaz de usuario de Data SOMinning para Adquisición de datos desde PubMed.

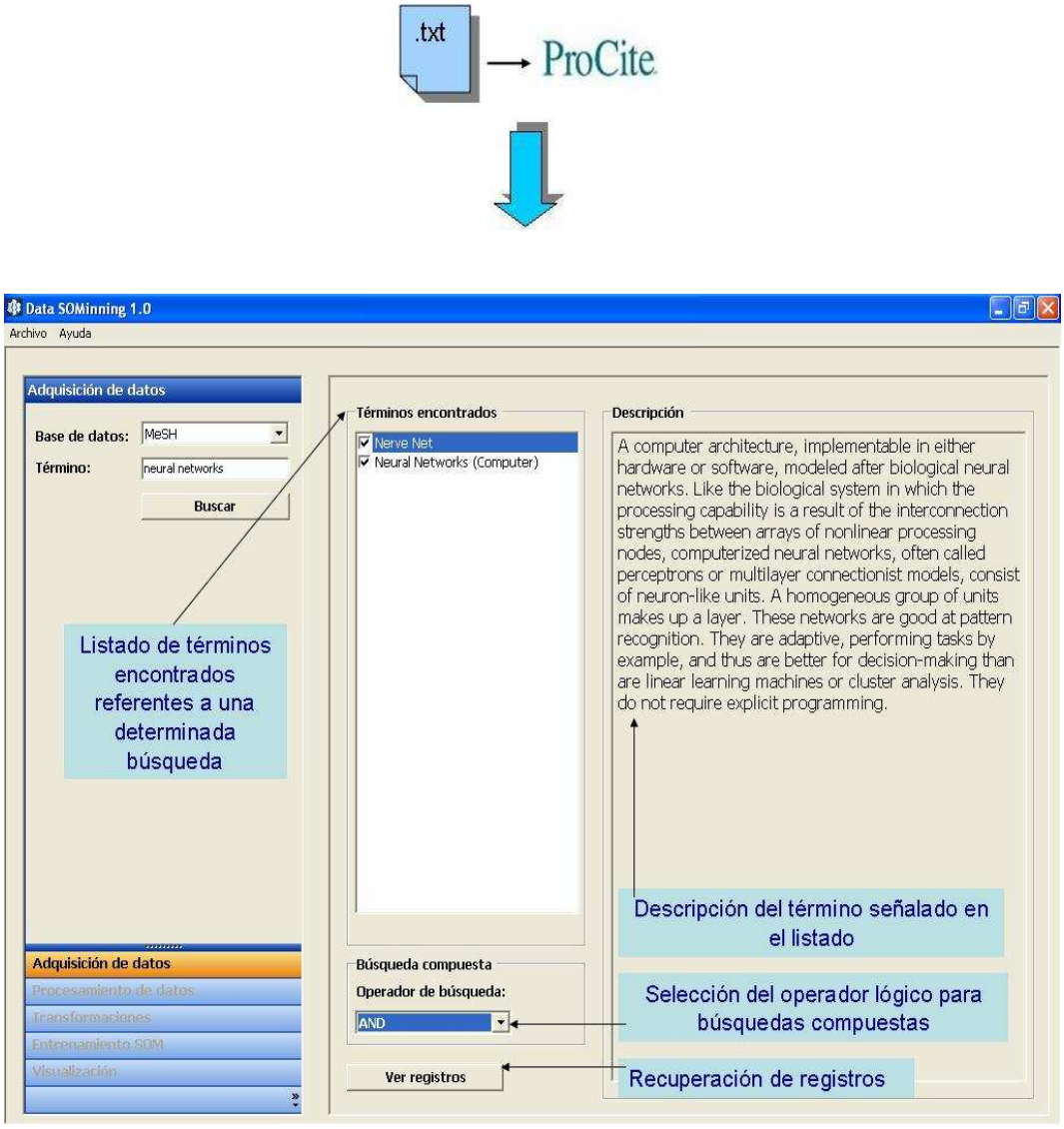


Figura 4.12: Vista de la interfaz de usuario de Data SOMinning para Selección de términos.

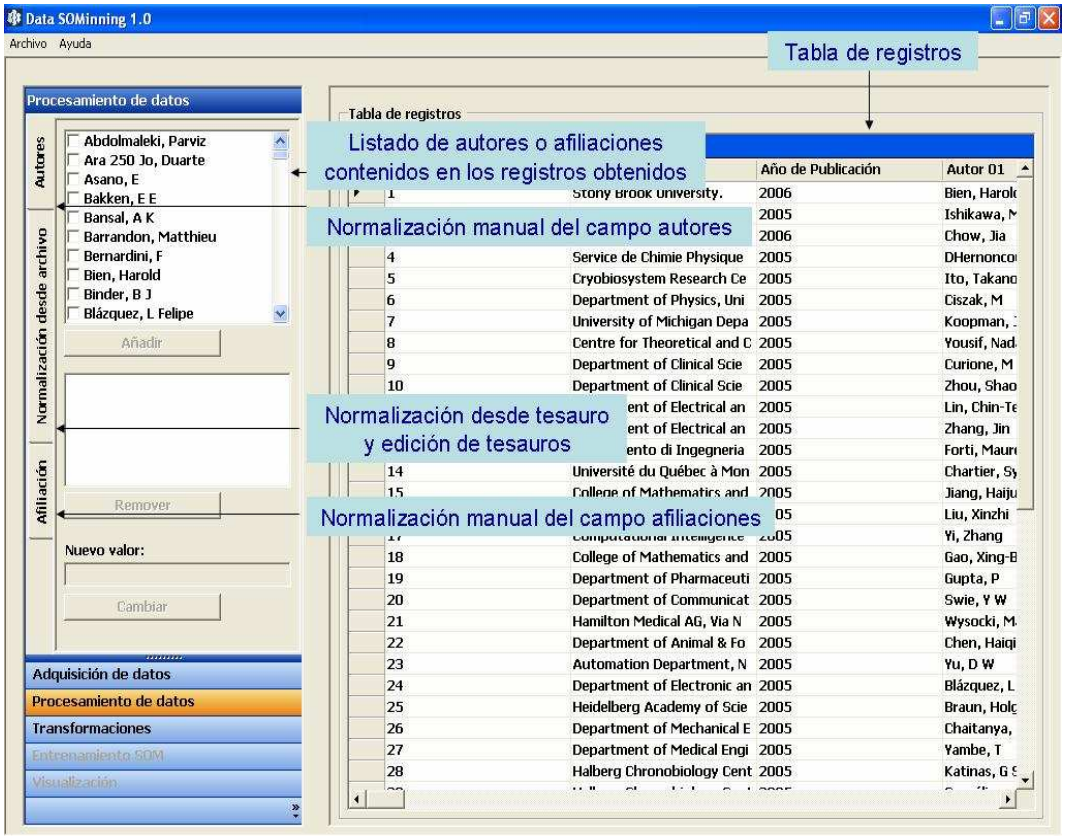


Figura 4.13: Vista de la interfaz de usuario de Data SOMinning para Procesamiento de datos.

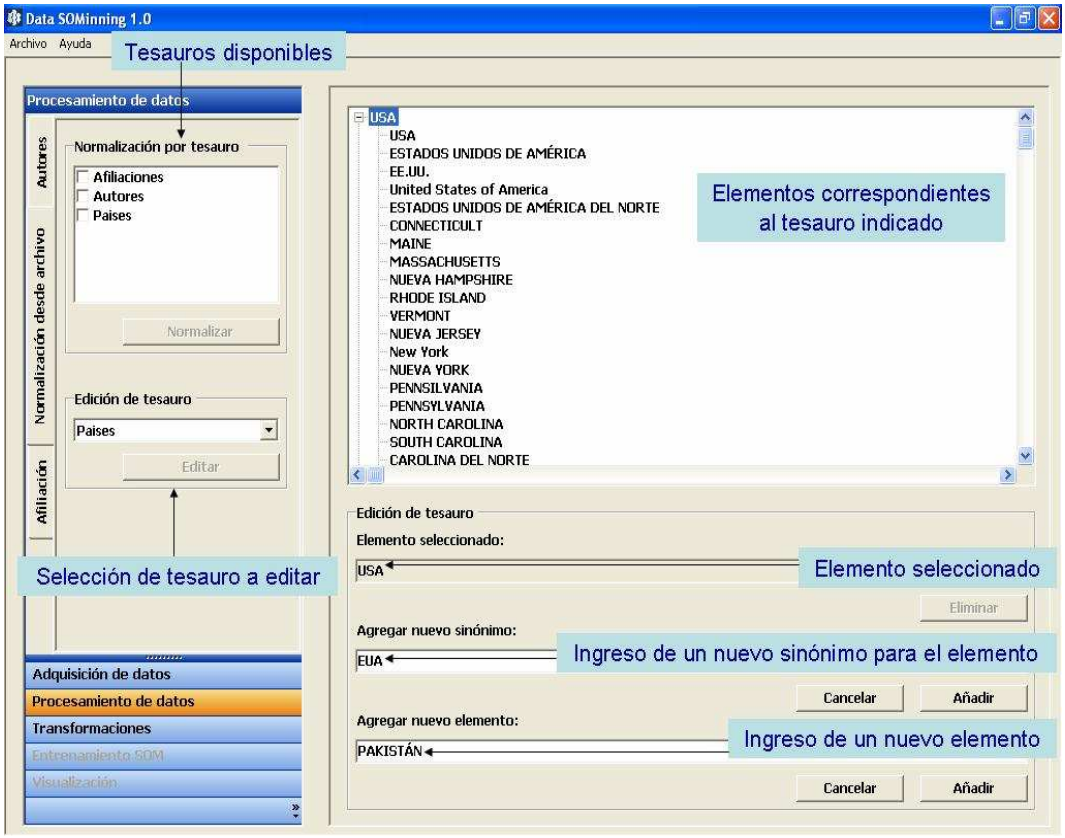


Figura 4.14: Vista de la interfaz de usuario de Data SOMinning para Edición de tesauro.

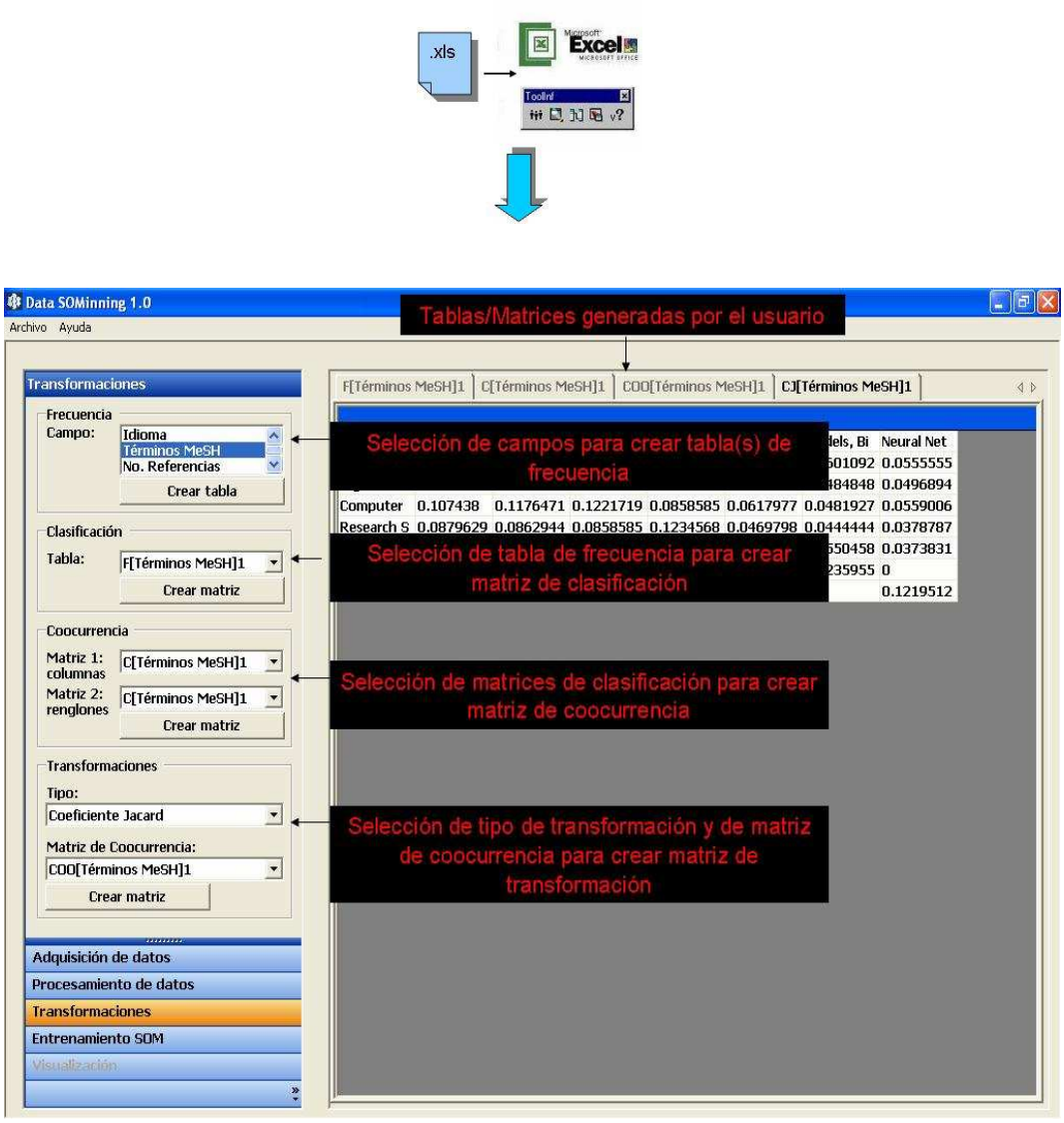


Figura 4.15: Vista de la interfaz de usuario de Data SOMinning para Transformaciones.

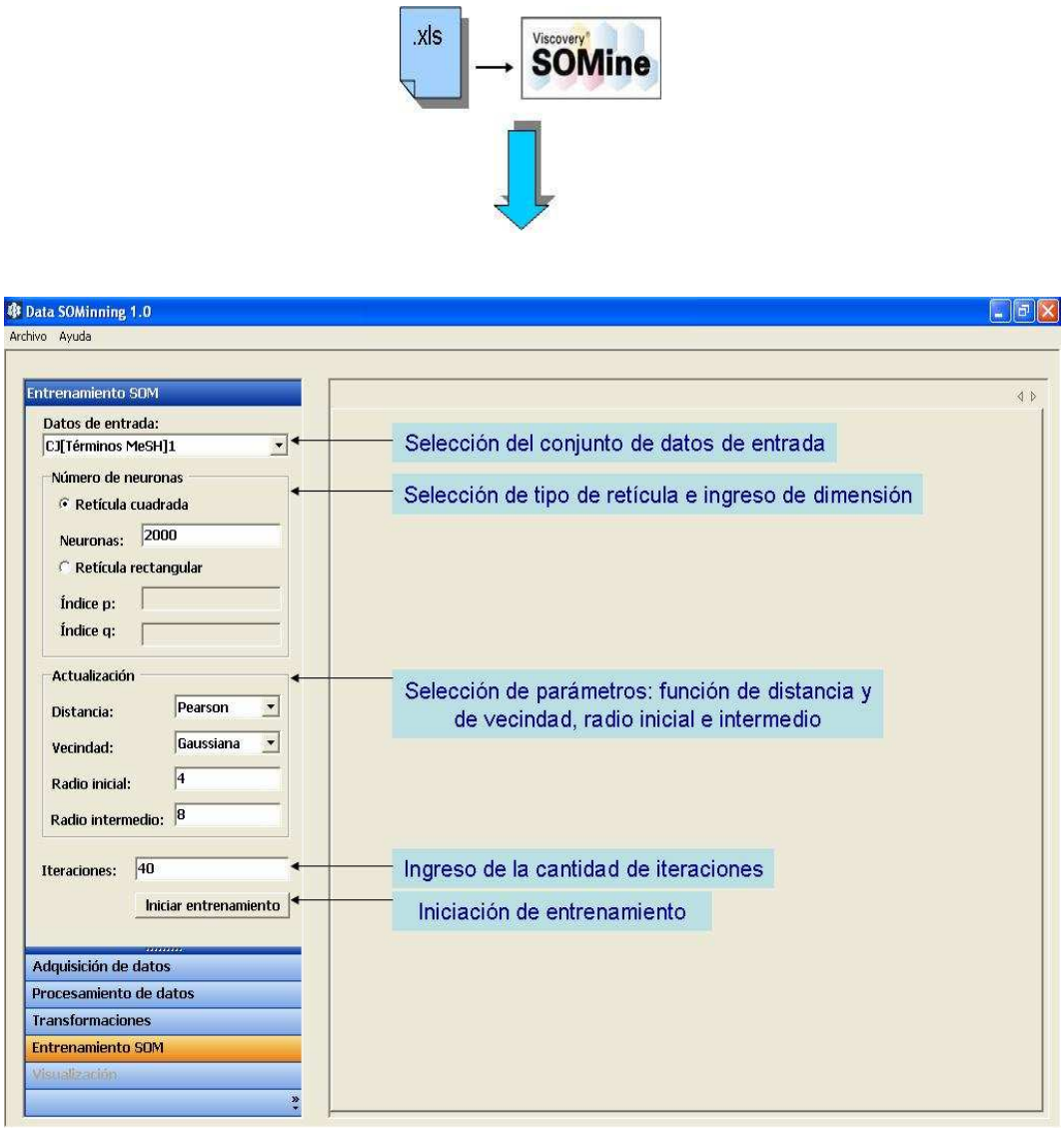


Figura 4.16: Vista de la interfaz de usuario de Data SOMinning para Entrenamiento SOM.

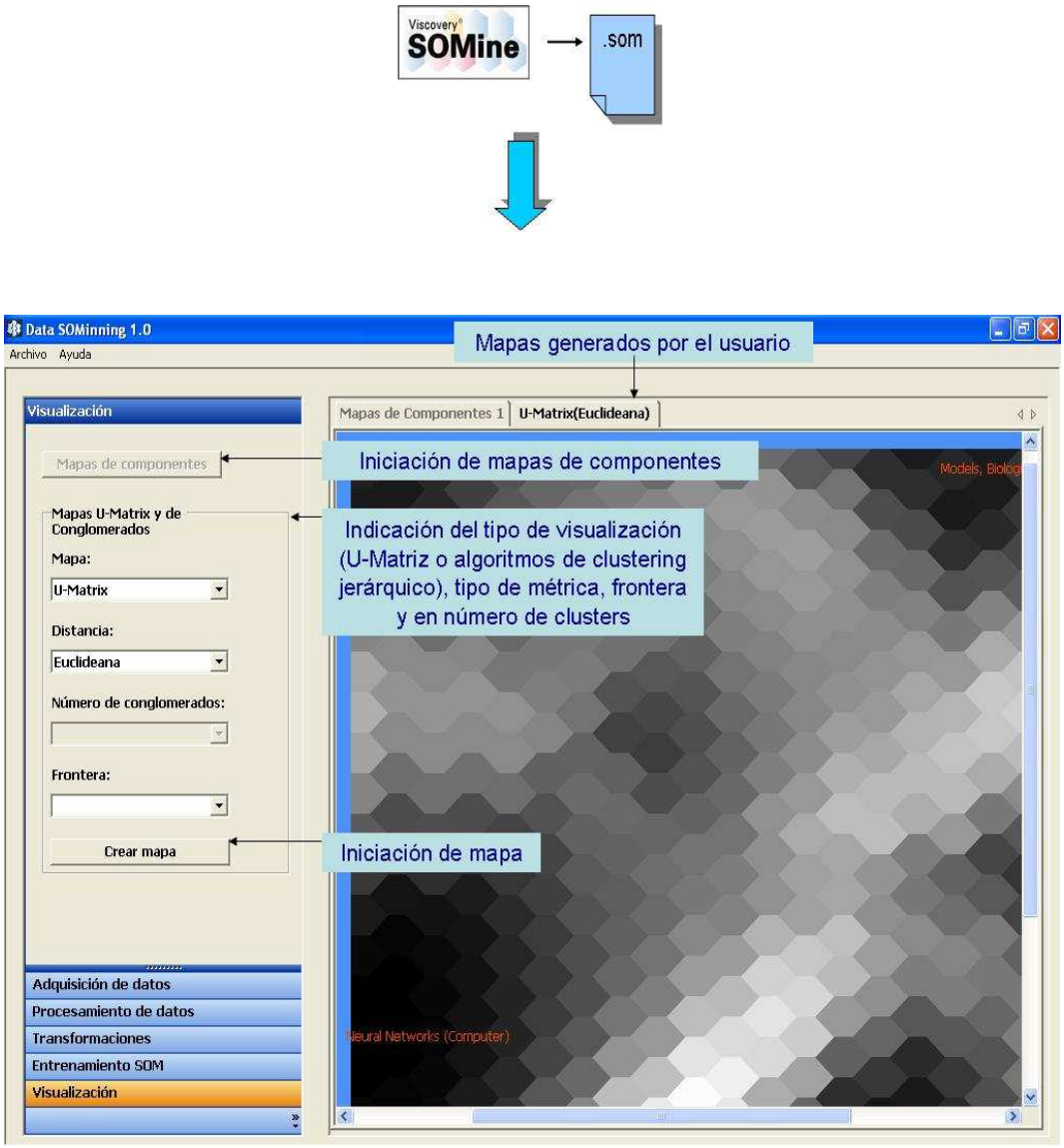


Figura 4.17: Vista de la interfaz de usuario de Data SOMinning para Visualización.

4.2.3. Diagrama de Paquetes.

Los **Diagramas de Paquetes** nos ayudan a clasificar clases en categorías, generalmente agrupándolas por funcionalidad o misión común. Dentro de cada paquete es factible definir subpaquetes con clases que realizan funcionalidades de propósito común. En esta sección, se muestra el diagrama UML de paquetes que refleja la estructura general de Data SOMinning.

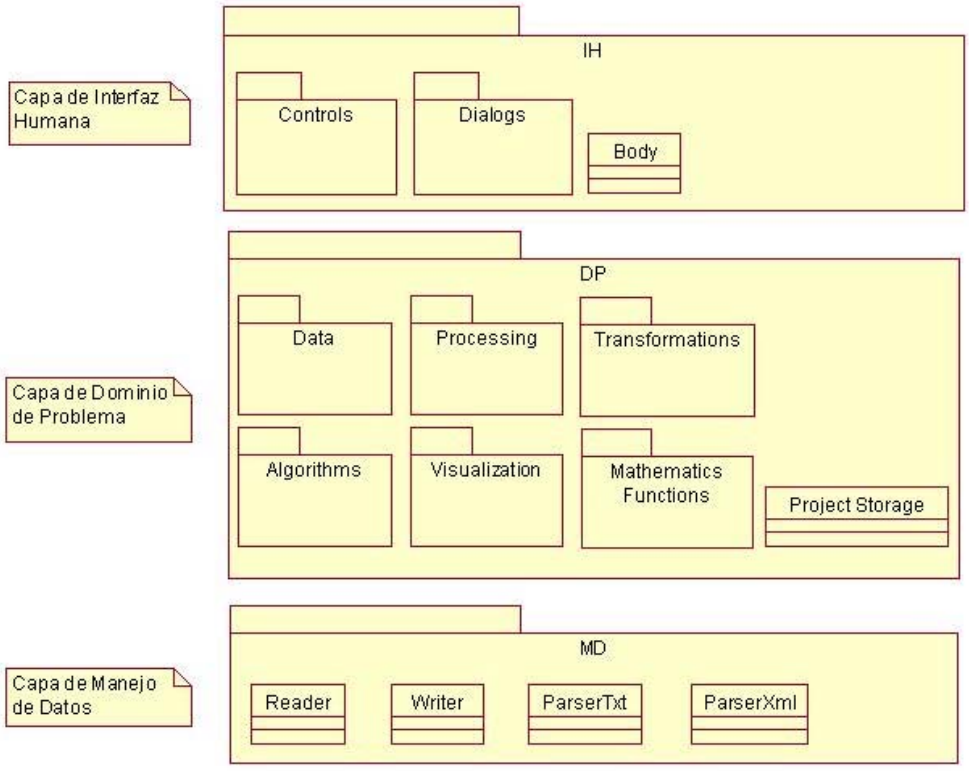


Figura 4.18: Diagrama de Paquetes de Data SOMinning.

4.2.4. Diagramas de Secuencia.

Los **Diagramas de Secuencia** pueden ser definidos como una vista que modela el comportamiento dinámico de un sistema. En esta sección, se muestran los diagramas UML de secuencia que modelan la dinámica de Data SOMinning.

4.2.4.1. Diagrama de Secuencia: Adquisición de datos desde MeSH.

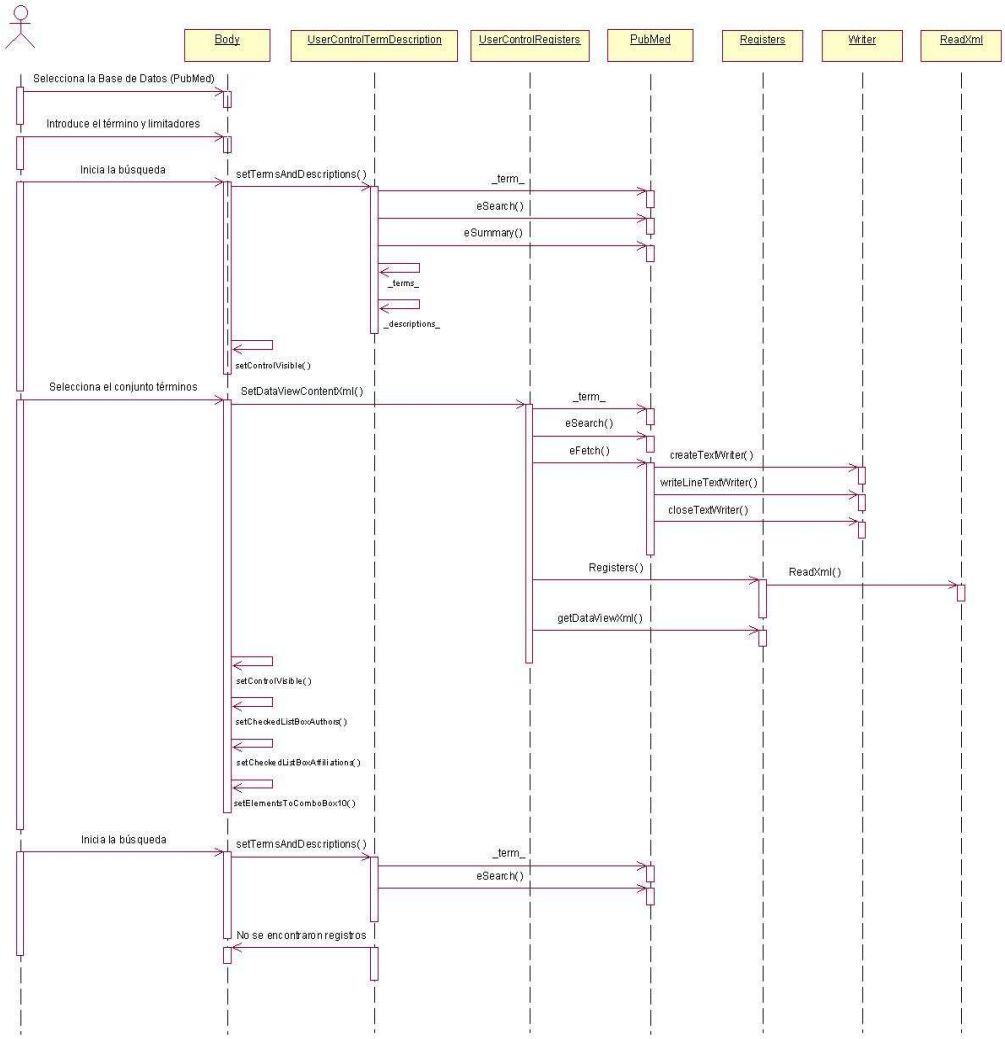


Figura 4.19: Diagrama de Secuencia para Adquisición de datos desde MeSH.

4.2.4.2. Diagrama de Secuencia: Adquisición de datos desde PubMed.

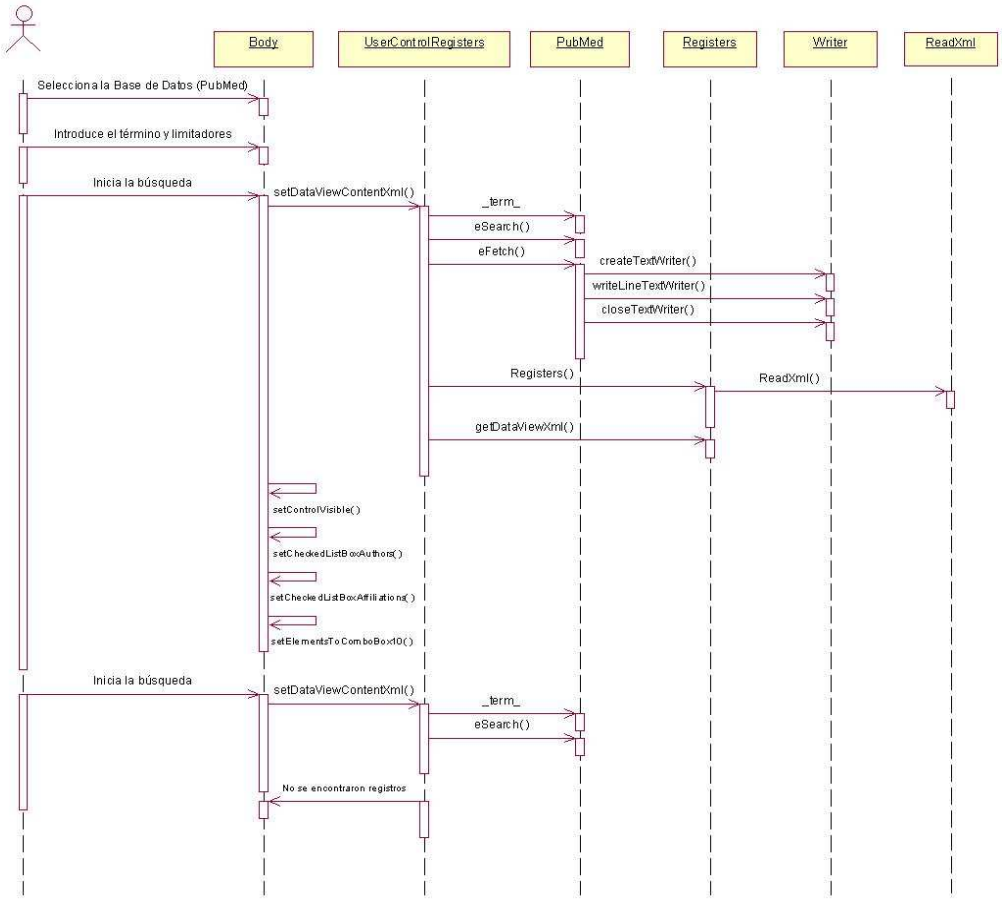


Figura 4.20: Diagrama de Secuencia para Adquisición de datos desde PubMed.

4.2.4.3. Diagrama de Secuencia: Selección de términos.

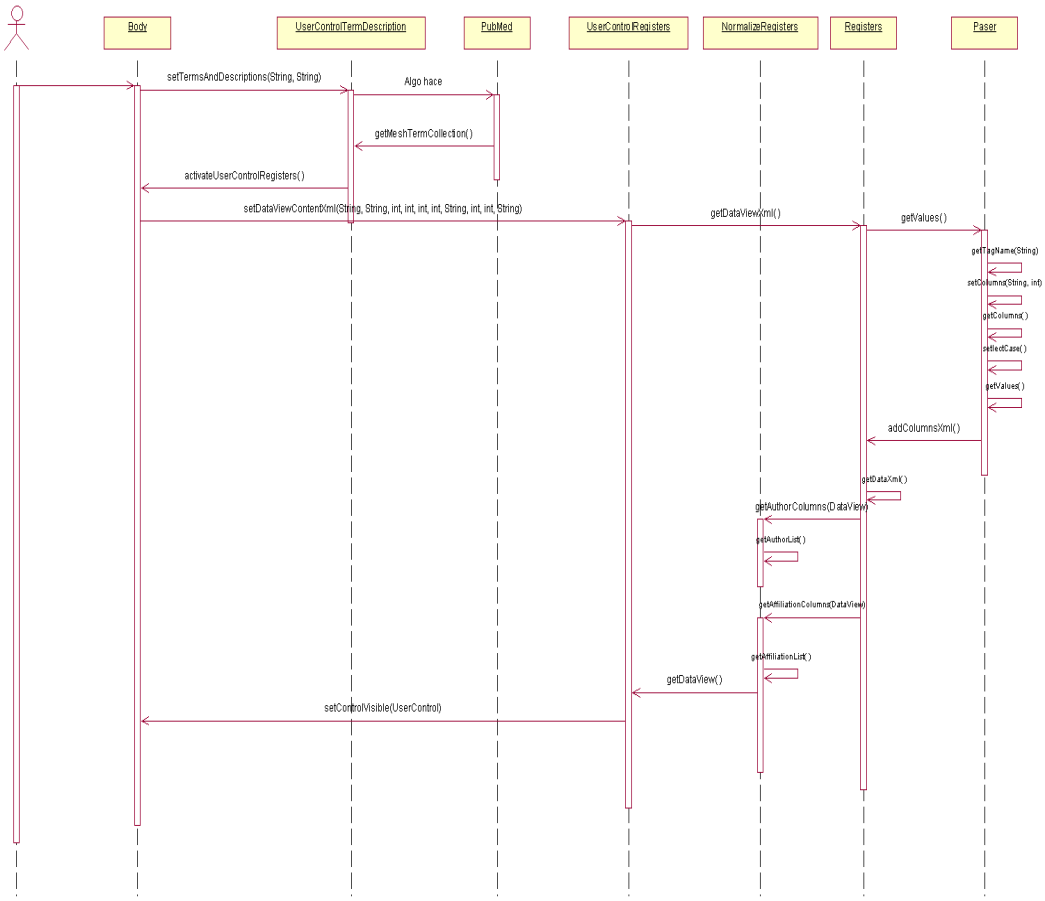


Figura 4.21: Diagrama de Secuencia para Selección de términos.

4.2.4.4. Diagrama de Secuencia: Procesamiento de datos manual.

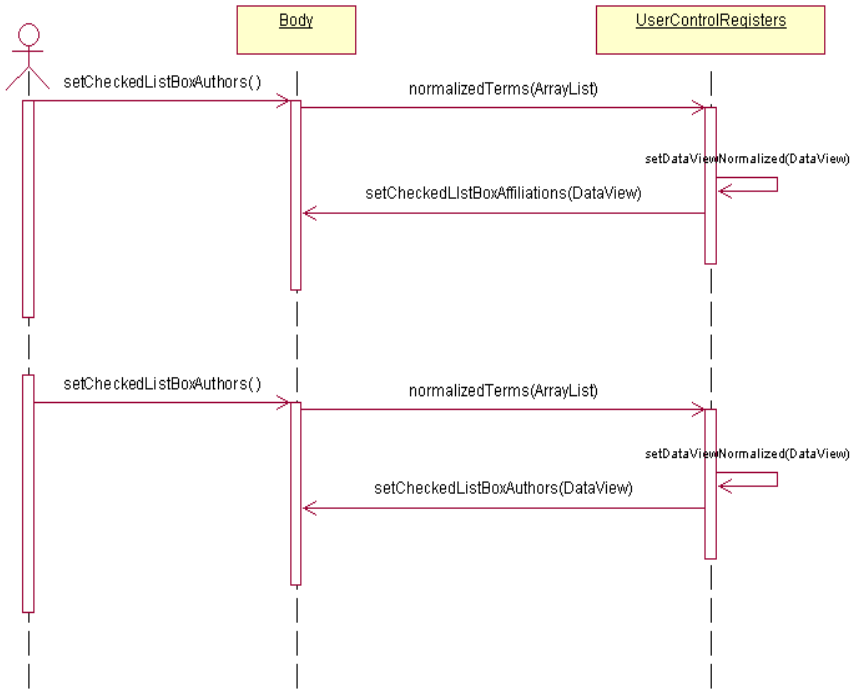


Figura 4.22: Diagrama de Secuencia para Procesamiento de datos manual.

4.2.4.5. Diagrama de Secuencia: Procesamiento de datos desde tesauro.

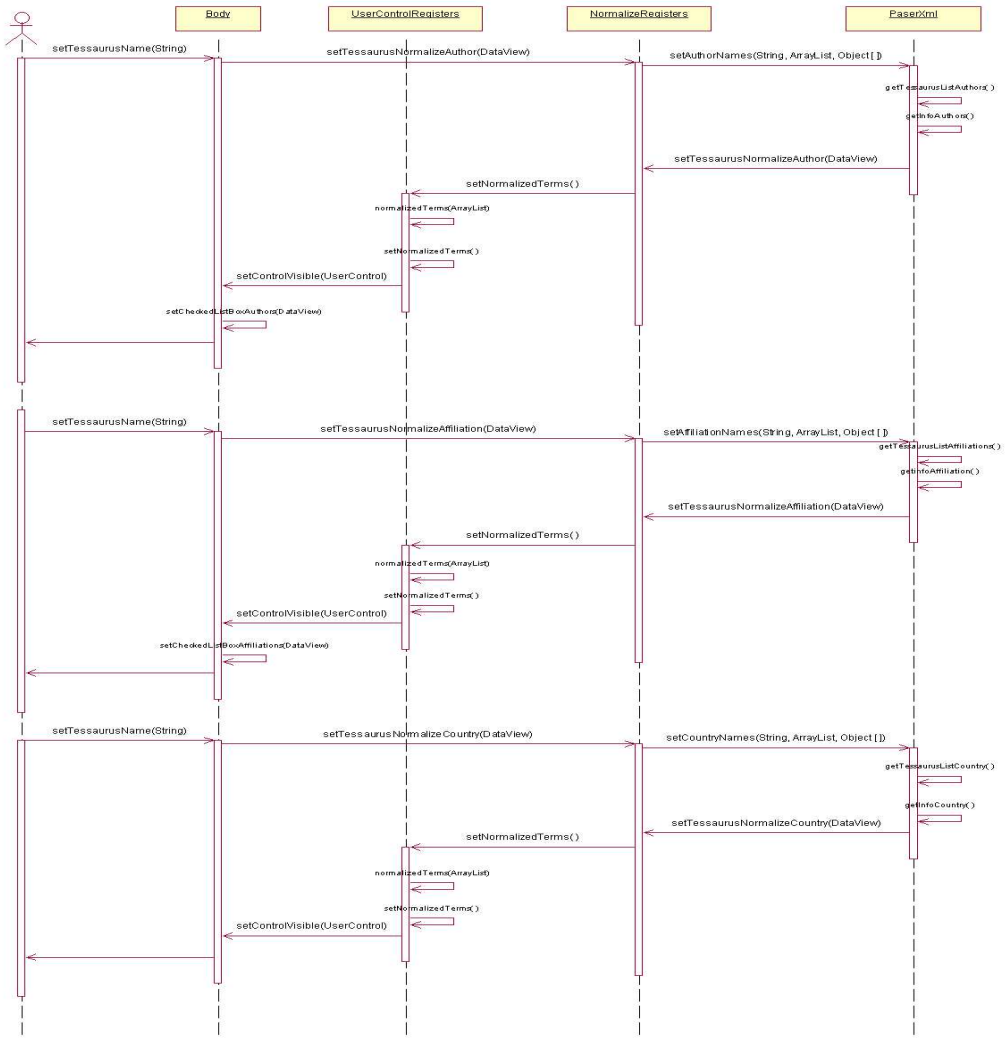


Figura 4.23: Diagrama de Secuencia para Procesamiento de datos desde tesauro.

4.2.4.6. Diagrama de Secuencia: Edición de tesauro.

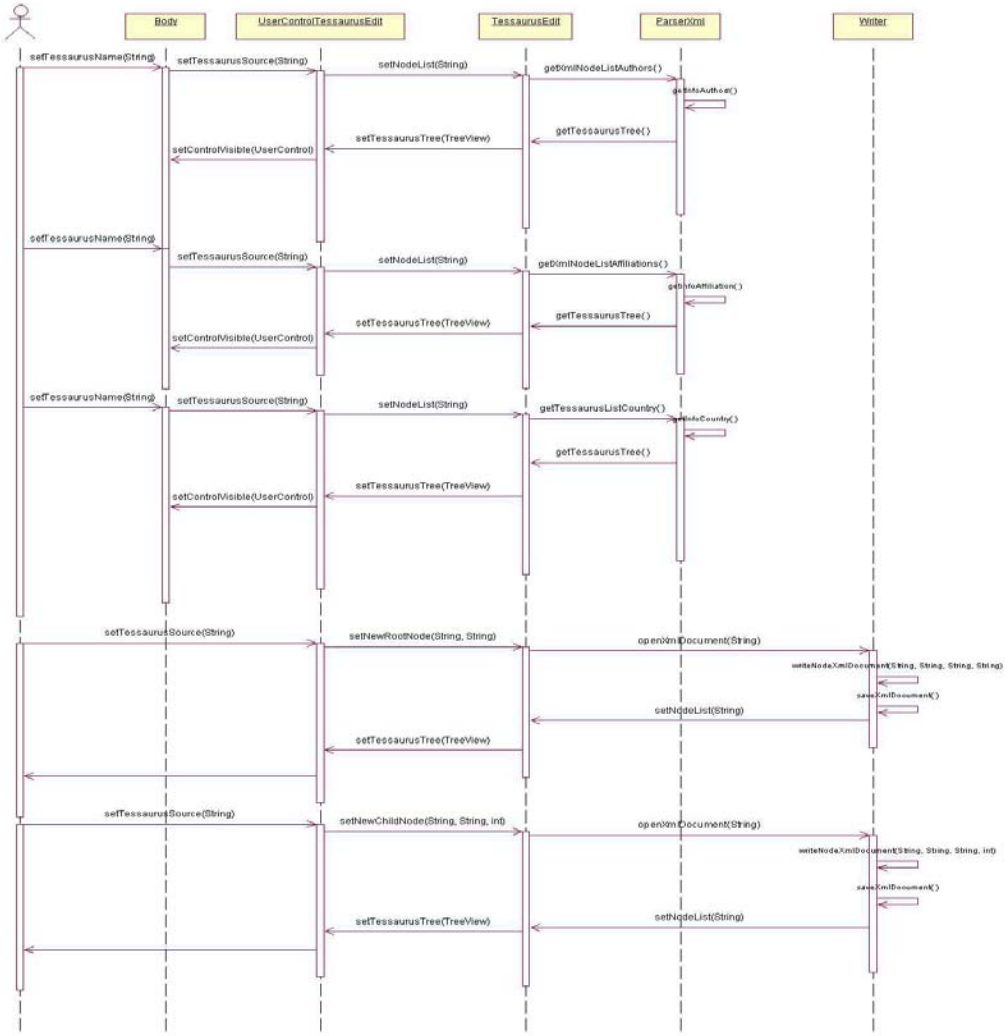


Figura 4.24: Diagrama de Secuencia para Edición de tesauro.

4.2.4.7. Diagrama de Secuencia: Transformaciones.

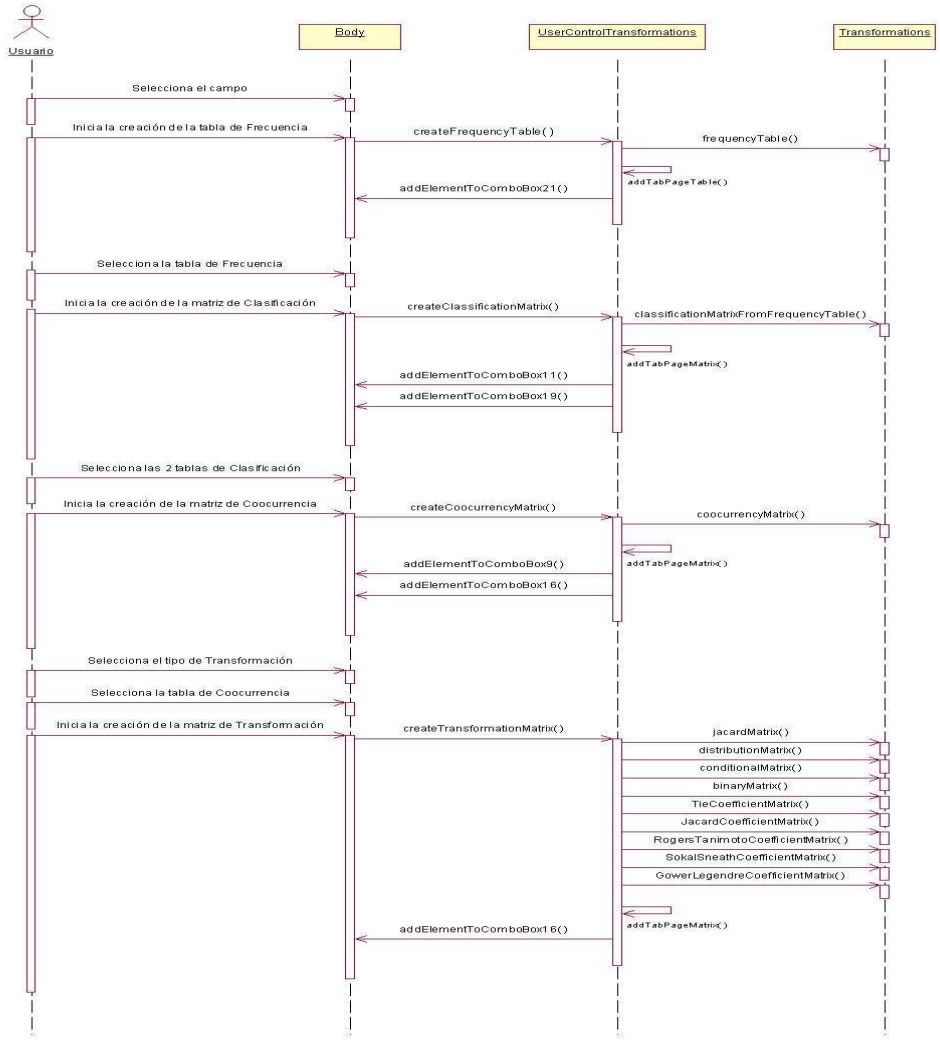


Figura 4.25: Diagrama de Secuencia para Transformaciones.

4.2.4.8. Diagrama de Secuencia: Entrenamiento SOM.

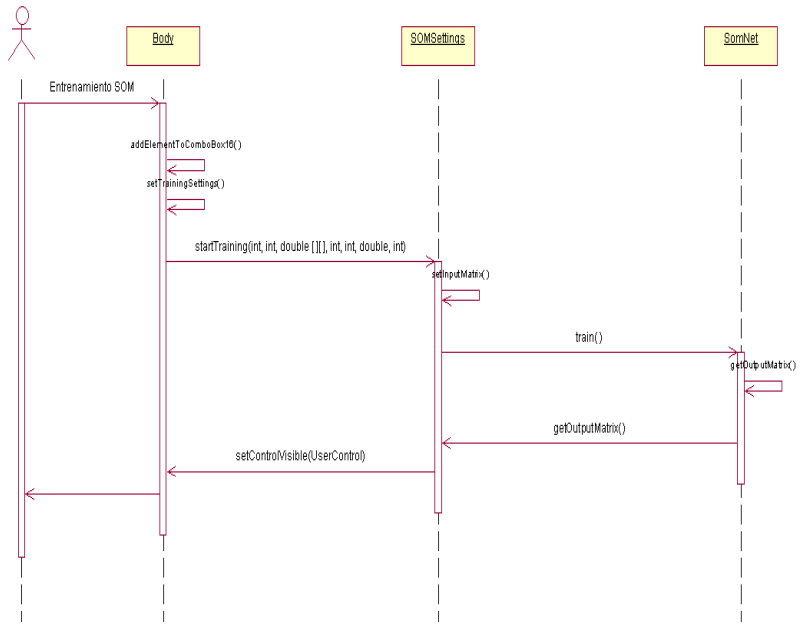


Figura 4.26: Diagrama de Secuencia para Entrenamiento SOM.

4.2.4.9. Diagrama de Secuencia: Visualización.

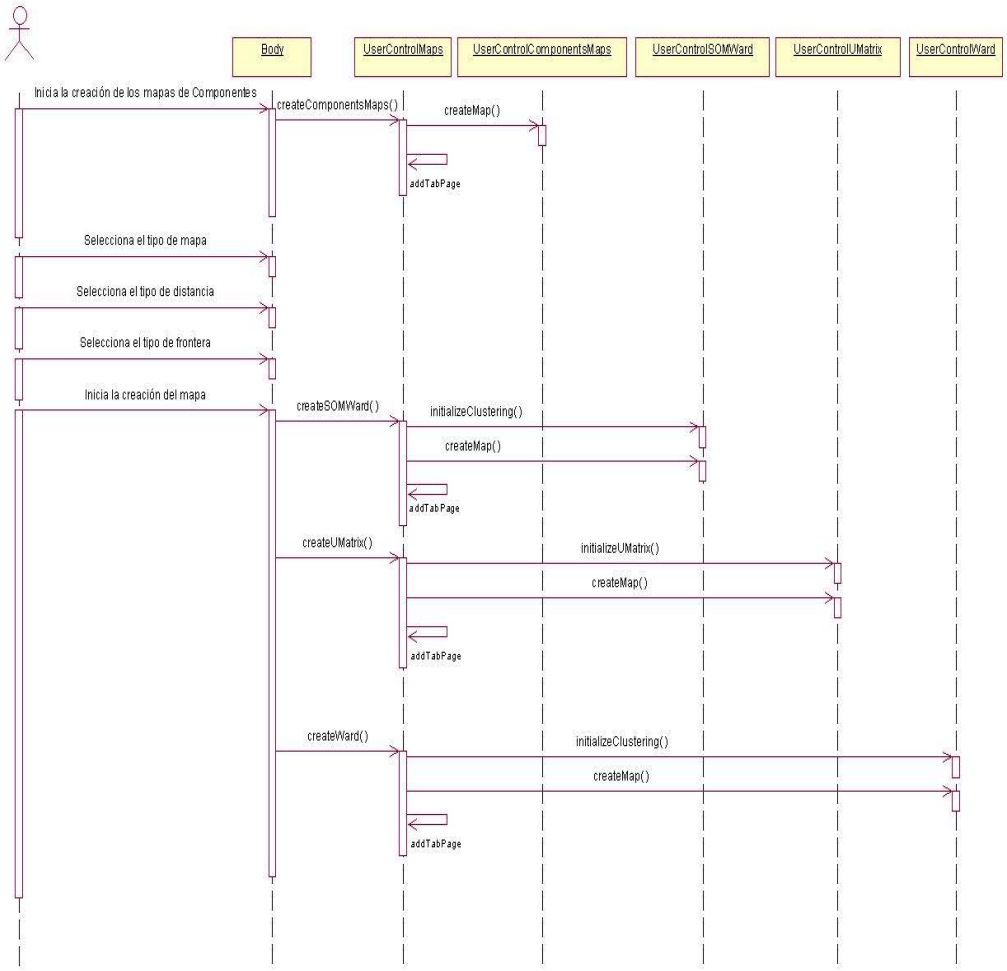


Figura 4.27: Diagrama de Secuencia para Visualización.

4.3. Implementación.

Una vez establecidos los objetivos y las necesidades del sistema, así como los elementos identificados en los diagramas UML, es posible iniciar la codificación del sistema en su totalidad.

En este capítulo mencionamos las 5 funciones básicas de ViBlioSOM® que integra Data SOMinning, en esta sección describimos detalladamente cada una de estas funciones ya imple-

mentadas del sistema.

4.3.1. Adquisición de datos.

En la metodología ViBlioSOM®, el usuario obtenía los datos realizando consultas a MedLine desde el portal de Entrez PubMed, almacenando los resultados en un archivo con formato de texto. Uno de los objetivos del sistema es que las consultas que realice el usuario, se hagan desde la interfaz del sistema; en Data SOMinning, los datos son obtenidos de MedLine a través del sistema de recuperación de información *Entrez Programming Utilities*.

Entrez Programming Utilities, también llamadas **eUtils**, es un conjunto de herramientas las cuales proveen acceso a las diversas bases de datos de MedLine. Para ello se realizan transacciones de petición y respuesta mediante el Protocolo de Transferencia de HiperTexto (HyperText Transfer Protocol, HTTP), operaciones independientes de la interfaz web Entrez PubMed para MedLine. Estos resultados son enviados en formato `XML` (eXtended Markup Language).

Para realizar una búsqueda en Data SOMinnig, el usuario debe seleccionar la base de datos en la cual se realizará dicha búsqueda, ya sea *MeSH* o *PubMed*, e ingresar la frase de consulta o término que se desee. En el caso de PubMed, los registros son obtenidos directamente. Eligiendo MeSH, el sistema envía la frase y el sistema de recuperación eUtils devuelve una lista de términos relacionados con la petición, así como la descripción de cada uno de éstos; Data SOMinning muestra esta lista, de donde el usuario seleccionará el o los términos que sean acorde a su interés (Fig: **??**).

Para la selección de los términos, correspondientes a una misma búsqueda, se proporcionan los operadores AND y OR; con éstos el usuario puede realizar búsquedas donde combine los términos de la lista que haya seleccionado (Fig: 4.12).

Cuando el usuario desea obtener los registros de los términos seleccionados, MedLine a través de las herramientas eUtils envían un archivo `XML` que contiene la información de los artículos. MedLine establece una serie de campos que determinan la información que contiene cada artículo como: autores, título del artículo, afiliación, país, etc. En total, MedLine define 63 campos; con base en las investigaciones realizadas por el grupo de trabajo del Instituto Finlay, consideran a 15 de ellos los más importantes:

1. Autor.
2. Términos MeSH.
3. Año de publicación.
4. Título del artículo.
5. Resumen *Abstract*.
6. Afiliación.
7. Idioma.
8. País de publicación.
9. Número de referencias.
10. Autor Corporativo.
11. Tipo de publicación
12. Páginas.
13. Número de sustancia.
14. Volumen.
15. Fecha de creación.
16. Título de la revista

De cada registro contenido en el archivo `XML`, Data SOMinning recupera estos 15 campos y los muestra al usuario en forma de tabla.

4.3.2. Procesamiento de datos.

En los capítulos anteriores, hemos mencionado la normalización de los datos como un proceso que el usuario realiza comúnmente de forma manual, o a través de archivos de tesauro. Para esta fase en ViBlioSOM®, Procite permite seleccionar los campos a normalizar y realizar estos cambios de forma global; dos de los campos más difíciles de normalizar son el de autor y de afiliación.

Data SOMinnig permite normalizar los registros de forma manual, proporcionando al usuario una lista de los campos de autor y de afiliación. Para normalizar estos campos, el usuario selecciona de esta lista, el o los nombres que considera son los mismos pero escritos de distinta forma. Seleccionados los nombres, el usuario debe ingresar la sintaxis correcta del nombre y presionar el botón correspondiente para realizar los cambios dentro de la tabla de registros (Fig: 4.13).

También, se incluyen tres archivos de tesauro: afiliaciones, autores y países. Los archivos están descritos en formato `XML` y pueden ser modificados de forma manual, ingresando la información directamente en el archivo fuente o desde la interfaz.

El objetivo de describir estos archivos bajo formato `XML`, es la facilidad de recuperar y mostrar la información que contienen, además de ser un formato libre. Si el usuario desea modificar los archivos desde el archivo fuente, no es necesario que él aprenda todo acerca de `XML`, basta con la sintaxis del archivo que consta de 4 etiquetas:

- **<Tessaurus>** define la lista de autores, afiliaciones o países que contiene el archivo.
- **<Field>** es el campo correspondiente al tesauro, es decir <Author>, <Affiliation> o <Country>
- **<Name>** define el nombre del campo.
- **<AlternativeName>** define el nombre alternativo del campo.

Desde la interfaz, los archivos de tesauro son representados como un árbol de dos niveles, donde cada nodo raíz representa la sintaxis correcta del nombre y cada subnodo, las distintas formas con que ese nombre puede aparecer.

Para añadir nuevos nombres, basta con ingresar el texto del nuevo nombre en el TextField “Agregar nuevo elemento” y presionar el botón “Añadir”; para añadir nuevos subnodos, el usuario debe seleccionar el nodo raíz e ingresar el texto del nuevo subnodo en el TextField “Agregar nuevo sinónimo” y presionar el botón “Añadir” (Fig:4.14).

4.3.3. Transformaciones de datos.

Sabemos que cuando la representación de los datos está libre de ambigüedades, nos es posible aplicar distintas transformaciones que reduzcan el número de variables y trabajar con las más peculiares del conjunto total de datos.

En la metodología ViBlioSOM®, la fase de transformación de datos es llevada a cabo por la macro ToolInf; los datos normalizados son almacenados en formato `xls` y deben ser exportados a Excel; las matrices generadas por el usuario se crean en distintas hojas de cálculo.

A diferencia de Excel, en Data SOMinning los datos son obtenidos directamente de la tabla de registros que el usuario normalizó en la fase anterior, sin importar el número de columnas que estos cálculos puedan generar. En esta fase se incluyen distintas funciones de conteo y reducción de dimensión:

- Tabla de Frecuencia.
- Matriz de Clasificación.
- Matriz de Coocurrencia.
- Matriz de Distribución.
- Matriz Condicional.
- Matriz Binaria.
- Matriz de Coeficiente de Empates.
- Matriz de Coeficiente de Jacard.
- Matriz de Coeficiente de Rogers y Tanimoto.
- Matriz de Coeficiente de Sokal y Sneath.
- Matriz de Coeficiente de Gower y Legendre.

Para generar las tablas de frecuencia, sólo es necesario que el usuario seleccione el campo sobre el cual quiere hacer el conteo; los campos por seleccionar son todos aquellos no nulos obtenidos de los registros originales.

La generación de matrices de clasificación, es a partir de alguna tabla de frecuencia previamente calculada. Basta con que el usuario seleccione una de estas tablas para que se cree la matriz correspondiente.

La generación de matrices de coocurrencia, es a partir de dos matrices de clasificación cualesquiera, previamente calculadas. Nuevamente, es suficiente con que el usuario indique cuáles dos desea intersectar para que se calcule la matriz correspondiente.

Para generar las matrices restantes; es decir, aplicar alguna de las transformaciones brindadas por el sistema, es necesario que el usuario haya generado con anterioridad alguna matriz de coocurrencia. El usuario deberá indicar como datos de entrada una matriz de coocurrencia para que se ejecute el procedimiento correspondiente.

Todas las operaciones que el usuario haya generado en esta etapa, aparecerán en pantalla en forma de TabPane. Cada pestaña del TabPane especifica el nombre de la matriz que generó el

usuario, este nombre está compuesto por una letra que indica el tipo de tabla o matriz seguido se encuentra el nombre del campo el cual la originó así como un ID. Por ejemplo: **F[Términos MeSH]1** indicaría la obtención de la primera tabla de frecuencia del campo Términos MeSH, **C[Términos MeSH]1** correspondería a la matriz de clasificicación de la antes mencionada tabla de frecuencia, **COO[Términos MeSH]1** representaría la matriz de coocurrencia tomando como entrada dos veces la misma matriz de clasificación C[Términos MeSH]1. Finalmente **CJ[Términos MeSH]1** correspondería a la matriz de coeficiente de jacard tomando como entrada la matriz de coocurrencia antes mencionada. Véase la figura 4.15.

4.3.4. Entrenamiento SOM.

ViBlioSOM® toma como algoritmo de minería de datos la red neuronal SOM y Viscovery® SOMine® es el sistema de software que implementa el algoritmo. Viscovery® SOMine® permite asignar los valores de distintos parámetros que intervienen durante el entrenamiento de la red; una de las principales características de Viscovery® SOMine® es la retícula hexagonal de la red.

La entrada de datos de Viscovery® SOMine®, son las matrices de datos generadas en Excel; en Data SOMinnig, el usuario selecciona la entrada de datos desde un listado que contiene los nombres de las matrices adecuadas generadas en la fase de Transformaciones. Seleccionada la entrada de datos, el usuario debe asignar la dimensión de la retícula, ésta puede ser cuadrada (`n` x `n`) o rectangular (`n` x `m`); la implementación de retícula de la red es de tipo hexagonal, como se había especificado anteriormente en los objetivos.

Uno de los aspectos importantes durante el entrenamiento es la actualización de los vectores de referencia de las neuronas; en Data SOMinning, el usuario puede definir cuatro parámetros que intervienen en la actualización: función de vecindad, métrica, radio de actualización inicial y radio de actualización intermedio.

El radio inicial e intermedio, definen el radio de actualización de los pesos de las neuronas. El radio de actualización se define como una función que decrece en cada iteración del entrenamiento de forma lineal. El radio inicial es el número de iteraciones en el cual el radio de actualización permanece constante; es decir, en este intervalo de tiempo se actualizan todas las neuronas.

El radio intermedio, es el número de iteraciones donde el radio de actualización disminuye hasta actualizar solamente a las neuronas ganadoras.

Se implementaron cinco distintas funciones de vecindad:

- Gaussiana.
- Umbral.
- Lineal.
- Sigmoidal en (0,1)

- Sigmoidal en (-1,1)

Una de las principales características de Data SOMinning es la implementación de distintas métricas; a diferencia de Viscovery® SOMine®, se implementaron 6 tipos distintos de métricas:

- Euclideana.
- Canberra.
- Manhattan.
- Minkowski.
- Pearson.
- Separación Angular.

Por útlimo, el usuario debe especificar el número de iteraciones del entrenamiento (Fig: 4.16).

4.3.5. Visualización.

Cuando concluye el entrenamiento de la red, el resultado es una matriz que almacena el sistema y sobre la cual se aplicará alguno de los algoritmos de visualización. Los algoritmos son:

- Mapas de Componentes.
- SOM-Ward.
- U-Matrix.
- Ward.

Si el usuario desea generar los mapas de componentes, sólo es necesario presionar el botón con la leyenda correspondiente.

Cuando se selecciona cualquiera de los tres últimos algoritmos, el usuario tiene la posibilidad de definir si desea que se dibujen las fronteras o no. Para el caso en que se requiere dibujar las fronteras, además podrá especificar el método para realizarlo (Single Linkage o Complete Linkage).

A su vez, al usuario le es permitido elegir el tipo de métrica por aplicar en cada uno de los algoritmos de visualización. Estas métricas son las mismas permitidas en el entrenamiento de la red neuronal SOM.

En el caso de los mapas de conglomerados se puede indicar el número de conglomerados que se desean obtener, o en su defecto indicar que el sistema obtenga este número mediante la aplicación del criterio implementado.

Una vez seleccionados los parámetros antes mencionados, basta con iniciar el proceso para la obtención del mapa. Los mapas generados aparecerán en la pantalla en forma de TabPane. Cada pestaña del TabPane indica el tipo de algoritmo y métrica aplicada.

4.3.6. Otras funciones de Data SOMinning.

Dentro de las funciones incluidas en el sistema tenemos el almacenamiento y recuperación de proyectos. Esta opción nos permite guardar en disco duro tanto resultados finales como parciales que formen parte de un caso de uso recientemente creado o modificado. Este almacenamiento contiene los parámetros proporcionados por el usuario, así como los resultados para cada una de las etapas. Lo anterior permite que cada vez que el usuario abra un proyecto, le sea fácilmente continuar las etapas no concluidas o en su defecto poder llevar a cabo una adecuada manipulación de los mapas obtenidos.

Los datos tales como los registros recuperados de MedLine, tablas y matrices, son almacenados en formato `XML`, mientras que las visualizaciones son imágenes en formato `jpeg` (Joint Picture Experts Group). Los parámetros proporcionados por el usuario se encuentran en un conjunto de archivos en texto plano, formato `txt`. Estos formatos permiten una gran portabilidad de los mismos, pudiendo ser éstos analizados en prácticamente cualquier computadora con características mínimas requeridas.

Finalmente el sistema permite la lectura de archivos que contengan registros obtenidos desde el sistema de búsqueda Entrez PubMed, ya sea en formato `XML` o `txt` únicamente. Esto permite flexibilidad al usuario de realizar la tarea de adquisición de datos de manera independiente al software, brindándole ventaja a todos aquellos usuarios que no cuenten con una conexión a Internet deseable.

Capítulo 5

Data SOMinning y su aplicación en la investigación científica.

En el último siglo, hemos presenciado un importante desarrollo tecnológico que ha revolucionado la relación entre las diversas áreas de conocimiento. La *Medicina* y la *Biología*, son claros ejemplos de esta revolución tecnológica: desde el procesamiento digital de imágenes de resonancia magnética hasta el análisis de secuencias genómicas.

Una pieza clave en los avances de la Medicina y la Biología, ha sido sin duda las *Matemáticas*, que se han convertido en una herramienta valiosa e indispensable para la investigación. En la actualidad, la creación de modelos matemáticos en sistemas biológicos, son un elemento importante para diversas áreas de la biomedicina.

Ante esta situación, universidades e institutos han establecido las bases para definir un nuevo perfil de individuos en el área de biomedicina: profesionales en *biomedicina computacional*, capaces de explotar el poder de las herramientas que ofrece el cómputo de alto rendimiento, así como el desarrollo de herramientas de software, en diversas líneas de investigación.

El objetivo de este trabajo, fue desarrollar una herramienta que permita realizar este tipo de análisis bibliométrico; el resultado es Data SOMinning. En los capítulos anteriores, hemos mencionado los aspectos teóricos que intervienen en el sistema, así como la metodología de su construcción, ahora es importante llevar a cabo un caso de estudio en Data SOMinning y analizar los resultados obtenidos.

En este último capítulo, contrastaremos los procedimientos llevados a cabo, así como los mapas generados durante la investigación aplicando la metodología ViBlioSOM®, primeramente utilizando los distintos sistemas de software comerciales y finalmente los obtenidos con el software propio Data SOMinning.

5.1. Matemáticas en Ciencias Biológicas.

En la actualidad, se manejan grandes volúmenes de datos en diversos ámbitos profesionales; sin embargo uno de los retos es conocer y dominar el uso de herramientas que permitan procesar estos datos convirtiéndolos en información que proporcionen conocimiento útil.

Un problema de interés para el grupo del Laboratorio de Dinámica No Lineal, es evaluar la evolución y comportamiento de temas comprendidos en las Ciencias Biológicas desde la perspectiva de las Matemáticas en el intervalo de anõs de 1950 a 2004.

Esta investigación resulta de gran importancia en el sentido de conocer nuevas aplicaciones, métodos y resultados en los que intervienen distintas áreas de las Matemáticas, además de identificar hacia dónde se deben destinar los recursos y esfuerzos con los que cuentan las instituciones ya sean educativas o gubernamentales. De esta manera, podemos reafirmar la importancia de la Bibliometría para el desarrollo tecnológico.

A continuación se describe el proceso realizado al utilizar la metodología ViBlioSOM® con sus dos posibles vertientes de aplicación: distintos sistemas de software comerciales (Procite, Excel y Viscovery® SOMine®) y el sistema Data SOMinning.

Una vez definido nuestro objetivo, continuamos a la segunda fase de ViBlioSOM®, la cual consiste en la adquisición y selección de archivo que contenga el conjunto de datos incial. Para ello recurrimos a MedLine de la NLM. Cabe señalar que la NLM utiliza el *MeSH Vocabulary*, este sirve para indizar toda la literatura biomédica de la base datos. El *MeSH Vocabulary* está organizado en 19 categorías principales y cada categoría se ramifica en series de subcategorías cada vez más concretas o específicas. Un ejemplo de categorías principales lo son **Ciencias Físicas** y **Ciencias Biológicas**, mientras que la subcategoría **Matemáticas** está contenida en Ciencias Físicas.

Es por ello que la investigación consiste en realizar una búsqueda avanzada; es decir, una búsqueda específica controlando lo que se busca. Recuperamos los documentos indizados con términos matemáticos que no estén indizados dentro de los temas estadísticos. Lo anterior debido a que la rama de Estadística representa el 83.98 % de Matemáticas y el número de registros es considerable. También aplicamos límites, específicamente sobre la *Fecha de Publicación* la cual restringimos del 1 de enero de 1950 al 31 de diciembre de 2004. El resultado que se obtuvo es una colección de 116,612 artículos en la consulta a través del portal de Internet para PubMed.

En el primer caso (distintos sistemas de software comerciales) es necesario exportar con Procite, el archivo recuperado del portal de Internet para PubMed y con ello comenzar a trabajar con los datos obtenidos. En este paso se realiza una pequeña selección interesándonos en el campo *Términos MeSH*. Cabe señalar que no se hizo distinción entre el *MeSH Principal* (MeSH Major Topic, MAJR) [1] y los *Términos MeSH* (MeSH Terms, MH) [2]. La tarea antes descrita corresponde a la tercera fase de ViBlioSOM®; es decir, procesamiento de datos.

[1] MAJR, se trata de un término MeSH que refleja una de las materias principales tratadas en el artículo.

[2] El Medical Subject Headings (MeSH), de la NLM es el vocabulario controlado de términos biomédicos que se utiliza para describir el tema de cada artículo de revista en MedLine.

Dentro de la misma fase seguimos ahora con la operación de transformaciones. Por lo que una vez seleccionado el campo deseado iniciamos la creación de 2 tablas de frecuencia que necesitamos. La primera de ellas contiene los siguientes términos de la subcategoría de Matemáticas:

- *Mathematics*
- *Algorithms*
- *Finite Element Analysis*
- *Fourier Analysis*
- *Fractals*
- *Game Theory*
- *Games, Experimental*
- *Mathematical Computing*
- *Decision Support Techniques*
- *Decision Theory*
- *Decision Trees*
- *Neural Networks (Computer)*
- *Nonlinear Dynamics*

En la segunda tabla de frecuencia, contabilizamos todos los términos de la categoría de Ciencias Biológicas, la cual está compuesta por las siguientes 14 subcategorías:

- *Biochemical Phenomena, Metabolism, and Nutrition*
- *Biological Phenomena, Cell Phenomena, and Immunity*
- *Biological Sciences*
- *Chemical and Pharmacologic Phenomena*
- *Circulatory and Respiratory Physiology*
- *Digestive, Oral, and Skin Physiology*
- *Environment and Public Health*
- *Genetic Phenomena*
- *Genetic Processes*

- *Genetic Structures*
- *Health Occupations*
- *Musculoskeletal, Neural, and Ocular Physiology*
- *Physiological Processes*
- *Reproductive and Urinary Physiology*

Recordemos que en esta fase, para el primer caso, hacemos uso de Excel junto con la macro ToolInf. Posteriormente calculamos las matrices de clasificación correspondientes a las tablas de frecuencia del paso anterior. En el caso de utilizar Excel con ToolInf representa varias desventajas, ya que para poder generar la matriz de clasificación del conjunto total de datos es necesario hacerlo por partes. Esto se debe a que en total obtuvimos 1668 componentes y Excel permite hojas de a lo más 250 columnas, teniendo que dividir éstos 1668 términos en 7 hojas de cálculo, obteniendo 7 matrices de clasificaciones parciales.

Otra característica indeseable de ToolInf es que al momento de generar matrices de clasificación, lo hace de manera errónea generando varias columnas vacías. Por otro lado, la macro ToolInf no realiza de manera satisfactoria la comparación de expresiones regulares. Tal es el caso del término *Physiology*, el cual no aparece en la tabla de frecuencia truncada pero en la matriz de clasificación respectiva sí lo hace. Lo anterior se debe a que la expresión: .*Physiology.* es la que en realidad se busca cuando debería ser “Physiology” tal cual. En este caso el metacaracter “.” es interpretado por el motor de búsqueda como cualquier otro caracter excepto los caracteres que representan un salto de línea. Esta alteración influye para la generación de resultados posteriores, lo que nos indica que dicha fase no se excluye de obtener resultados distintos en relación a Data SOMinning.

En el caso del sistema Data SOMinnning la representación de las matrices de clasificación fue modificada, pero esto fue únicamente con la finalidad de mejorar el tiempo de ejecución. Esta modificación no afecta en ningún momento la correcta interpretación de la misma, así como la obtención de resultados posteriores. Inclusive permite que esta matriz pueda ser un conjunto de datos de entrada para la red neuronal artificial SOM.

Como último paso en la etapa de transformaciones creamos la matriz de coocurrencia, uno de los cálculos más pesados del proceso. Satisfactoriamente para dicha tarea, Data SOMinning requiere un tiempo de ejecución menor al de ToolInf. Además que con 3 matrices de clasificación parciales, tendremos que intervenir de manera directa para poder obtener la matriz de coocurrencia deseada. Esta matriz tiene como resultado 13 componentes, correspondientes a la primera tabla de frecuencia, y 1668 datos de entrada, correspondientes a la segunda tabla de frecuencia. A esta matriz de coocurrencia se le aplicó una normalización de acuerdo al criterio del coeficiente de Jacard. Esta matriz de transformación, es la entrada de datos de la red neuronal artificial.

Ahora que ya tenemos listos los datos para la fase de minería de datos ejecutamos Viscovery®

SOMine® para la aplicación del algoritmo Batch Map (variante de la red neuronal SOM). Para los distintos parámetros del entrenamiento de la red se asignaron los siguientes valores:

1. Retícula de 2044 neuronas.
2. Vecindad gaussiana.
3. Razón de cambio de vecindad de 0.5.
4. Métrica euclideana.
5. 40 iteraciones.

Por otro lado, para los distintos parámetros del entrenamiento de la red en Data SOMinning se asignaron los siguientes valores:

1. Retícula cuadrada de 2025 neuronas.
2. Vecindad gaussiana.
3. Métrica Pearson.
4. Radio inicial: 10.
5. Radio final: 22.
6. 40 iteraciones.

Una vez concluido el entrenamiento, en ambos sistemas de manera independiente, continuamos con la elaboración de diversos mapas. Estos mapas serán la manera de visualizar los resultados generados por el entrenamiento y de esta forma poder comparar ambos sistemas. Recordemos que en Data SOMinning se implementó el algoritmo SOM básico, esto con la finalidad de crear una comparación aún más distinguida.

Las visualizaciones creadas son los mapas de componentes, mapas U-Matrix, así como mapas de conglomerados con algoritmos SOM Ward y Ward en los dos sistemas. Estos mapas nos permitirán llevar a cabo una exploración de la auto-organización con la finalidad de dar una interpretación.

El despliegue de los mapas de componentes tiene la particularidad de representar la distribución de los valores de cada variable de los datos asociados a cada neurona en un mapa. La distribución de valores 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 en color azul, los intermedios en verde y amarillo y los valores máximos en rojo.

Los mapas U-Matrix los obtenemos al calcular los promedios de las distancias entre cada neurona y sus vecinas inmediatas, por lo que cada neurona tiene asociada un valor. El conjunto de estos valores se asocia a una escala cromática, para posteriormente asignar un color a cada

neurona. Este tipo de mapa permite visualizar las relaciones de cercanía entre los vectores de referencia de manera global.

Además de la visualización de los mapas de componentes y U-Matrix se pueden obtener mapas de regiones, que representan conglomerados del conjunto de datos. Para realizar esta división los sistemas brindan la posibilidad de utilizar dos algoritmos distintas: SOM Ward y Ward.

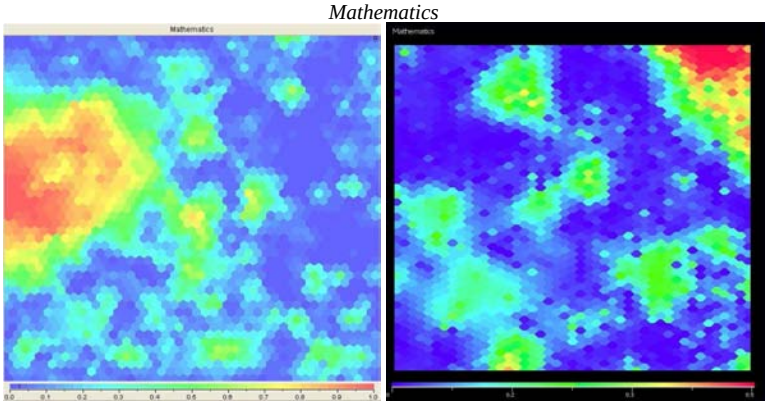
En el punto de partida del algoritmo SOM Ward, cada neurona representa a un conglomerado. En cada paso, dos conglomerados distintos se unen en uno sólo. Los conglomerados seleccionados son aquellos cuya distancia es la mínima de todas las distancias entre conglomerados. Esta distancia toma en cuenta cuando dos conglomerados son adyacentes en el mapa, lo que tiene como consecuencia que sólo se pueden unir conglomerados adyacentes en el mapa.

El algoritmo Ward se diferencia del SOM Ward en que no compara únicamente las distancias entre conglomerados adyacentes, por lo que los mapas no necesariamente muestran regiones conexas. Con los mapas de conglomerados SOM Ward y Ward es posible el establecimiento de relaciones entre las distintas variables.

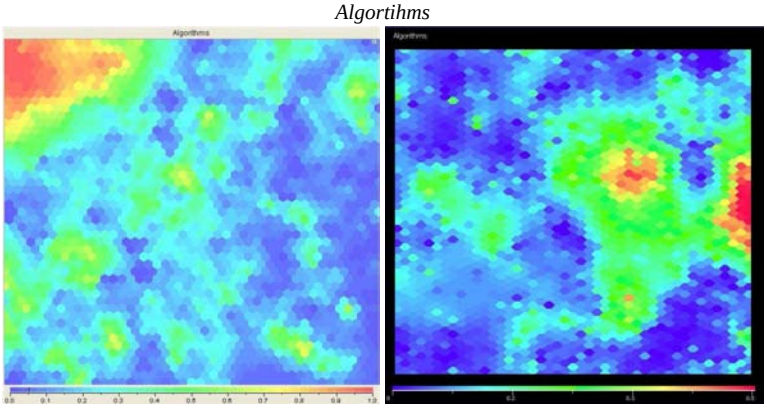
Los mapas U-Matrix, así como los mapas de conglomerados SOM Ward y Ward contienen un descriptor asociado a cada dato. De esta manera es posible adicionar referencias con información a la visualización del conjunto de datos y contar con elementos que faciliten la interpretación de los mapas dentro de un contexto específico. En nuestro caso de estudio solamente se exhiben las etiquetas correspondientes a los términos que conforman la subcategoría Ciencias Biológicas.

A continuación se muestran todos los mapas generados para nuestro caso de estudio ***"Matemáticas en Ciencias Biológicas durante el período 1950-2004"*** mostrando primero los obtenidos con Viscovery® SOMine® continuando con los de Data SOMinning.

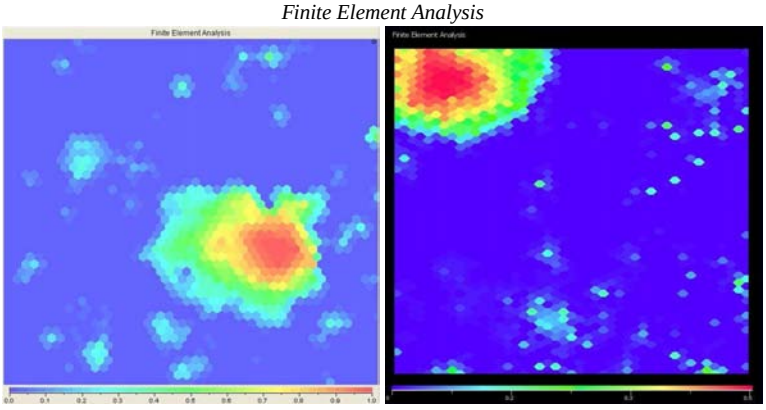
Mathematics



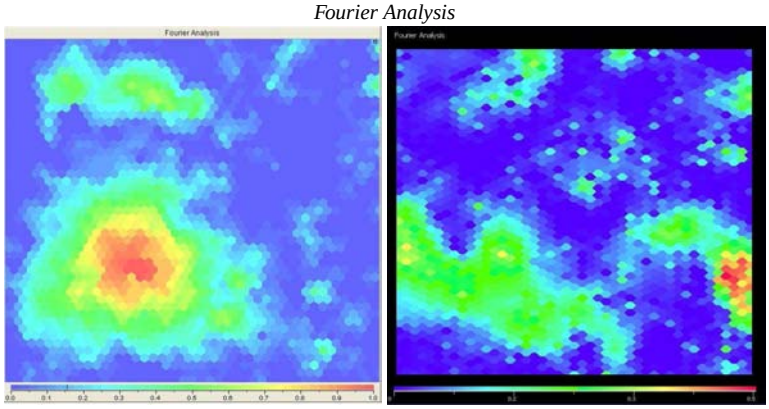
Algortihms



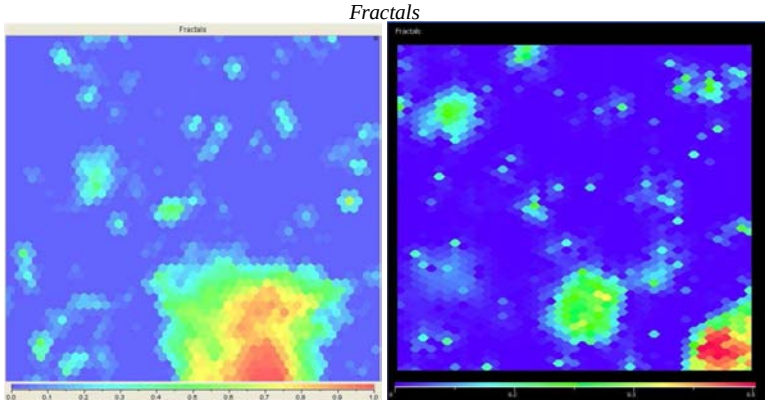
Finite Element Analysis



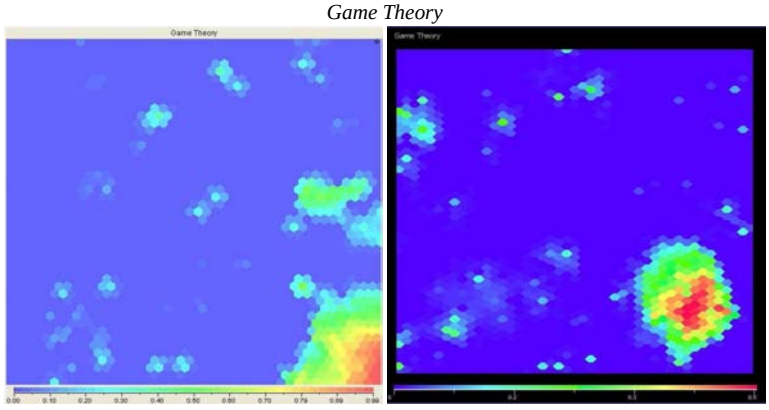
Fourier Analysis



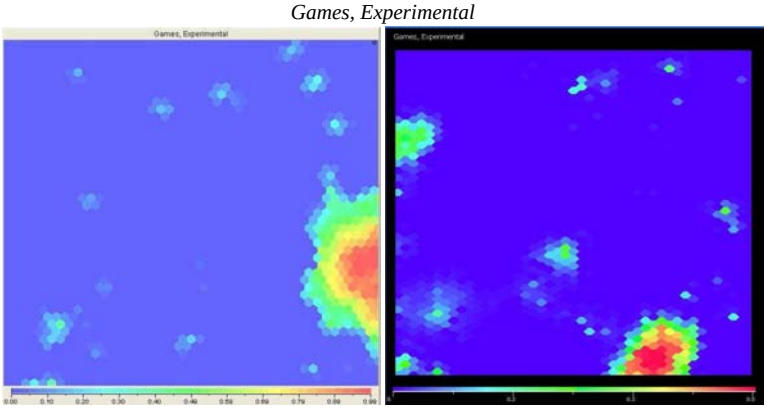
Fractals



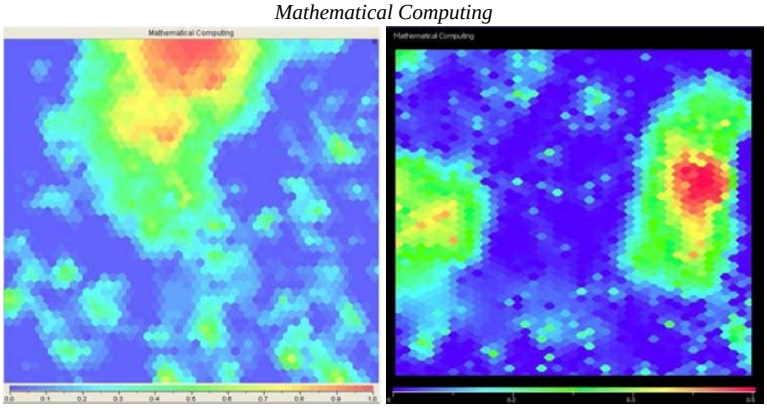
Game Theory



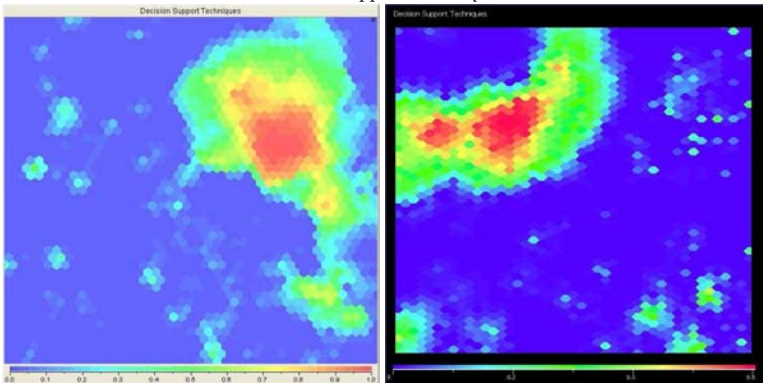
Games, Experimental



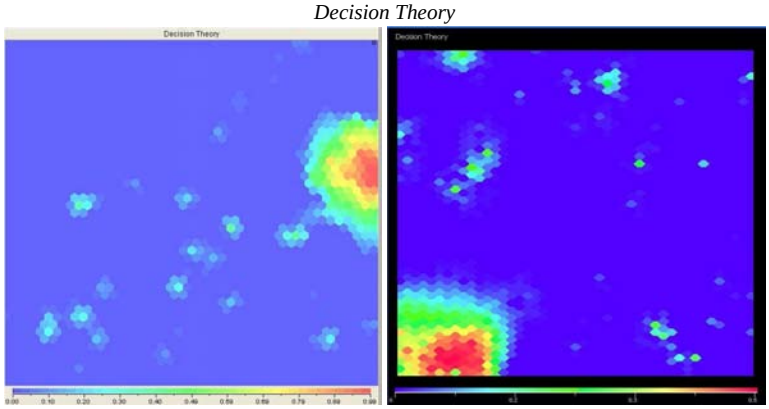
Mathematical Computing



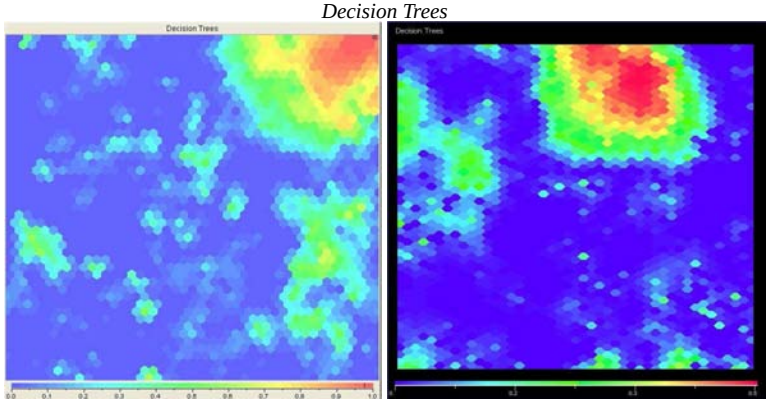
Decision Support Techniques



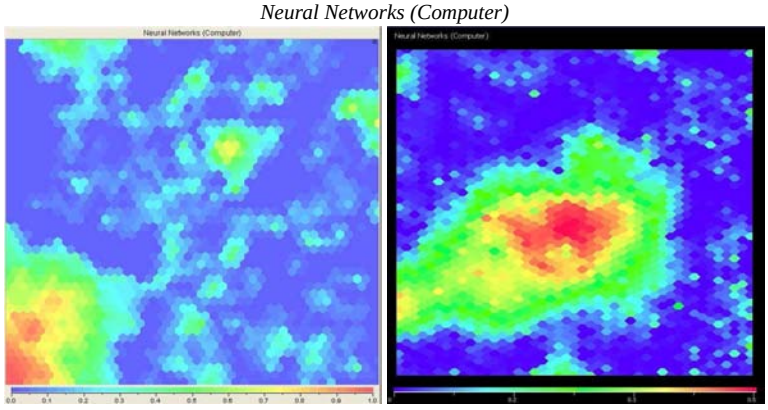
Decision Theory



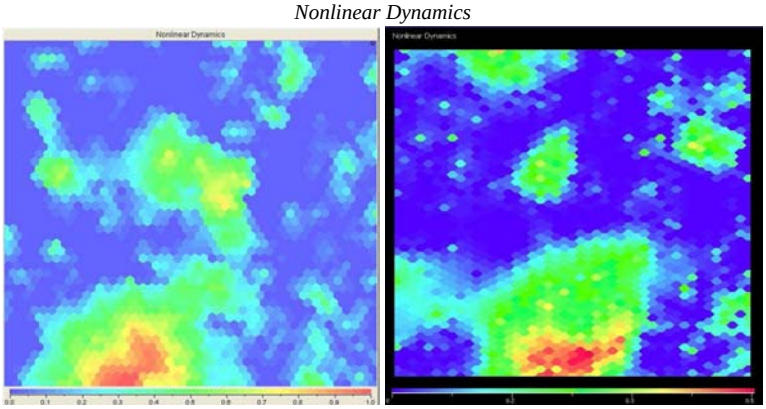
Decision Trees



Neural Networks (Computer)



Nonlinear Dynamics



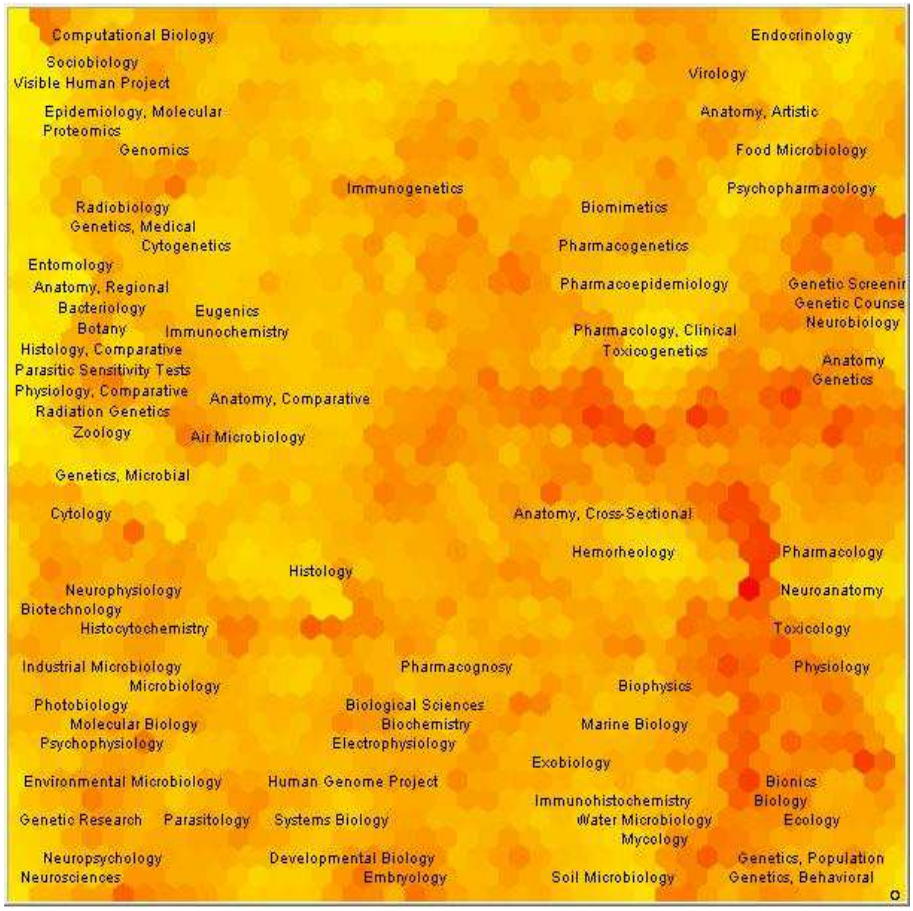


Figura 5.1: Mapa U-Matrix a partir de Viscovery® SOMine®.

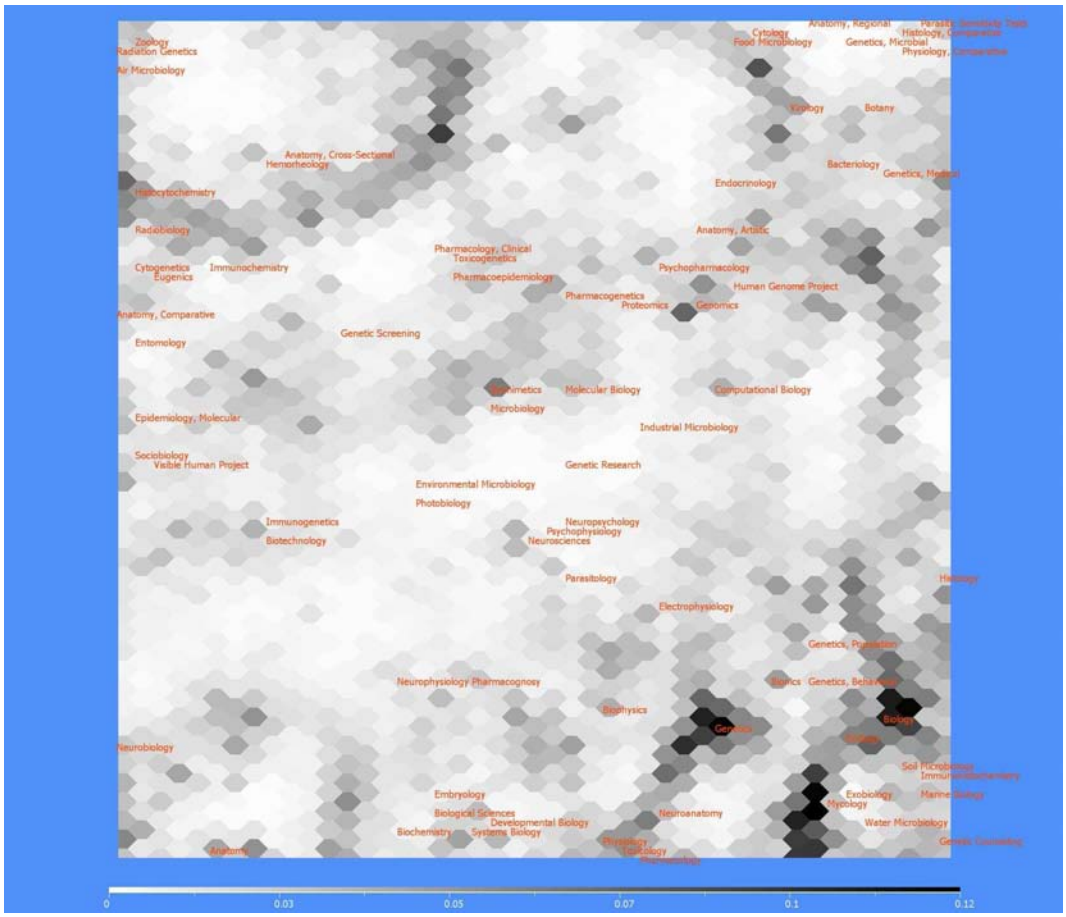


Figura 5.2: Mapa U-Matrix a partir de Data SOMinning.

Figura 5.3: Mapa de Conglomerados aplicando SOM Ward a partir de Viscovery® SOMine®.

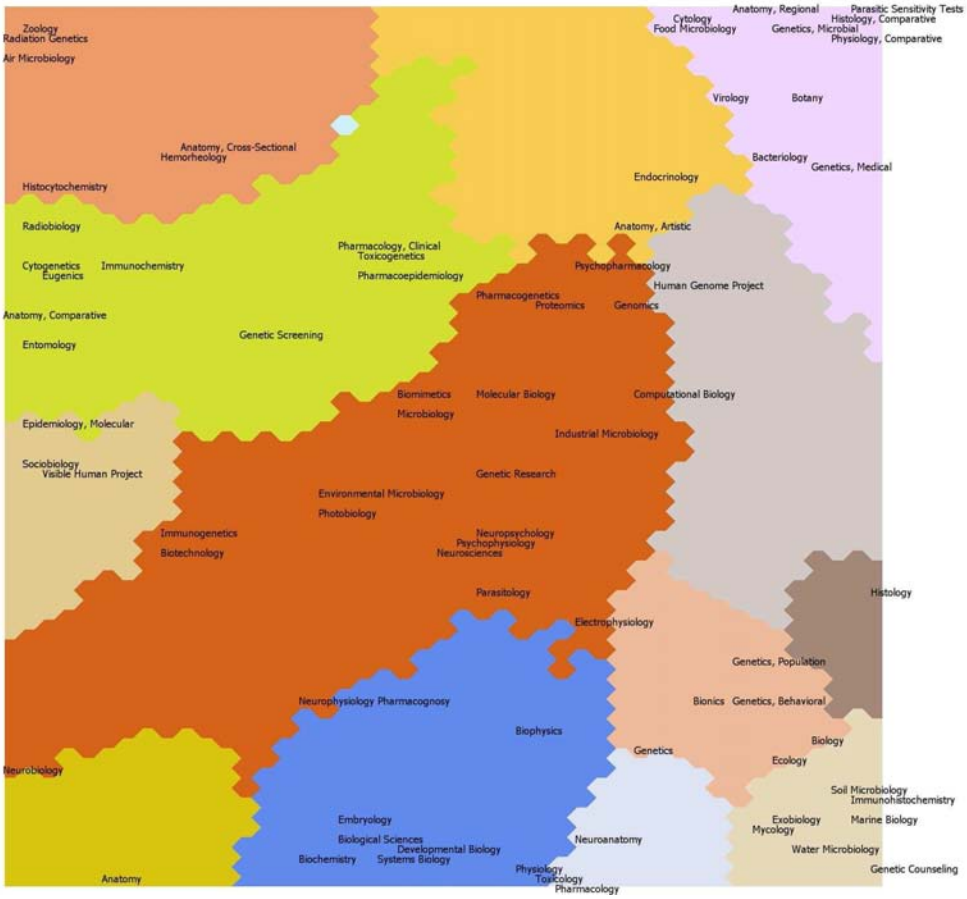


Figura 5.4: Mapa de Conglomerados aplicando SOM Ward a partir de Data SOMinning.

Figura 5.5: Mapa de Conglomerados aplicando Ward a partir de Viscovery® SOMine®.

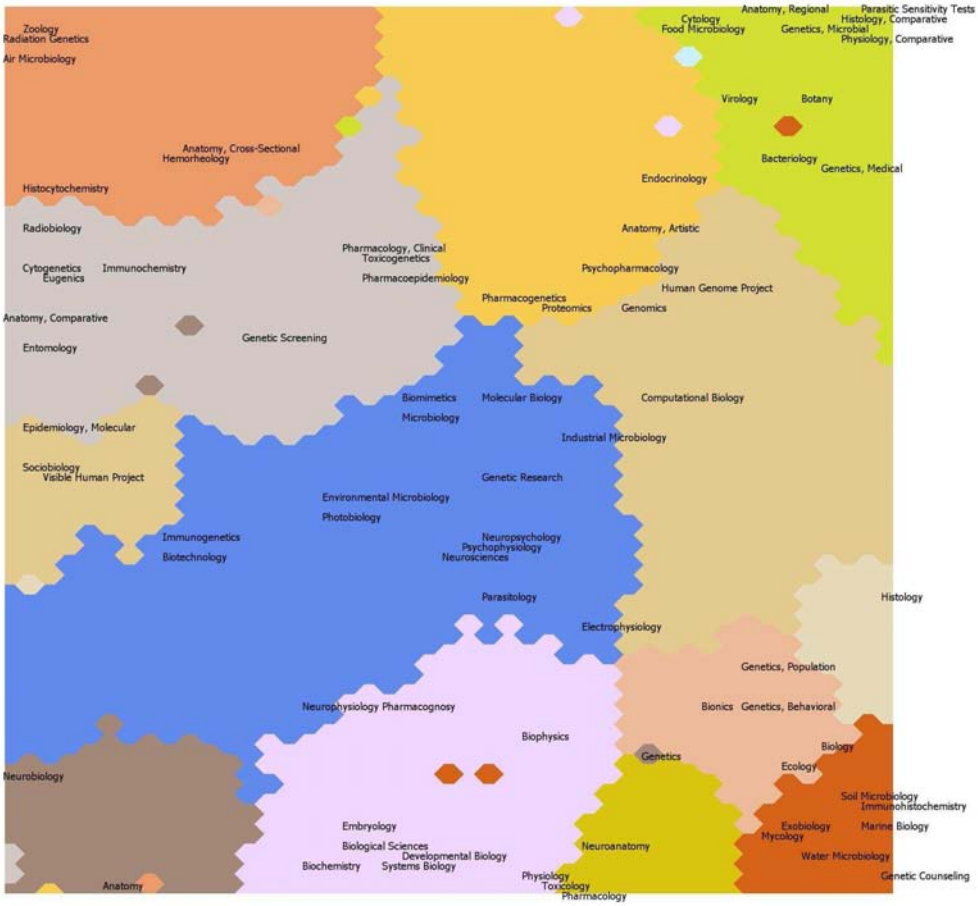


Figura 5.6: Mapa de Conglomerados aplicando Ward a partir de Data SOMinning.

A partir del análisis de los dos conjuntos de mapas podemos llegar a resultados semejantes. Cabe mencionar que Viscovery® SOMine® arrojó mapas de 13 clusters tanto para el algoritmo SOM-Ward como Ward. En el caso de Data SOMinning se obtuvieron mapas de 14 y 13 clusters para los algoritmos SOM-Ward y Ward respectivamente. Lo anterior aplicando el criterio, implementado en el sistema, para plasmar el número de conglomerados que den una mejor representación de la clasificación de los datos. Este criterio hace una división de clusters de tal manera que tenga una disimilaridad mayor entre clusters, así como también la mayor similaridad entre los elementos que componen cada cluster.

Al utilizar Data SOMinning se aplicaron las distintas métricas tanto para entrenamiento como para la creación de visualizaciones, llegando a la conclusión de que la transformación de coeficiente de Jacard se acopla de manera positiva con la métrica Pearson.

De los mapas podemos apreciar regiones que abarcan solamente un término, o ninguno, así como otros que abarcan un número considerable de ellos. Para dar conclusiones respecto a la interpretación de los mapas, nos enfocamos en los dos componentes de mayor interés por nuestro grupo de investigación que son: *Nonlinear Dynamics* y *Neural Networks (Computer)*.

A partir de la interpretación de los mapas podemos percibir la estrecha relación de la dinámica no lineal con la Bioquímica, Sistemas Biológicos y la Electrofisiología. A su vez apreciamos que las redes neuronales artificiales mantienen una relación de aplicación en el campo de la Neurofisiología, Biotecnología, Microbiología, Biología Molecular y Neurología.

Las aplicaciones actuales apuntan hacia los modelos biológicos y muy vinculados éstos a las redes neuronales artificiales. Una evidencia de estos progresos es que los propios investigadores, que trabajan en Medicina con redes neuronales, se han agrupado en una Sociedad Científica Mundial: ANNIMAB (Artificial Neuronal Network in Medicine and Biology).

De forma general, se ha podido apreciar que la Biomedicina y las Matemáticas se complementan en su propio desarrollo. Tal es el caso del área de Dinámica No Lineal, término de la subcategoría Matemáticas, el cual sirve de soporte a la Biomedicina para el desarrollo de su tecnología, como soporte de análisis epidemiológico y mejores respuestas clínicas. Un ejemplo, son las investigaciones sobre el establecimiento de patrones comunes en pacientes sometidos a cámara gamma con talio radioactivo para el diagnóstico de insuficiencia coronaria (Universidad de Viena) [DdC].

Así podemos apreciar la importancia de la visualización de los datos, resultados de la aplicación de indicadores bibliométricos, para descubrir un nuevo conocimiento relevante para la gestión de los proyectos y el re-direccionamiento de líneas de investigación.

Conclusiones

Durante los últimos años, hemos sido testigos de un importante proceso evolutivo de los sistemas de software, el cual ha sido trascendental para los usuarios de sistemas de cómputo. Día a día, son desarrollados enormes volúmenes de sistemas de software "a gran escala", que implementan desde las mejores técnicas de la inteligencia artificial, hasta los más avanzados algoritmos de procesamiento digital de imágenes.

Al mismo tiempo y de manera natural, el avance mismo de la ciencia ha proporcionado las condiciones ideológicas, físicas y económicas para esta evolución. Este avance ha permitido que hoy en día, el *desarrollo de software científico* sea parte fundamental para la investigación.

Es así como instituciones privadas, universidades e institutos destinan cada vez más, importantes cantidades de recursos económicos a la contratación y apoyo de investigadores y científicos. Cabe señalar, que tanto universidades como institutos, han sido pieza clave en la generación de nuevos profesionales, capaces de desarrollar este nuevo tipo de software.

A lo largo del presente trabajo, hemos mencionado la importancia y el impacto del desarrollo y la aplicación de herramientas de software científico, en distintas áreas del conocimiento. De esta forma, llegamos a esta sección donde presentamos las conclusiones obtenidas durante el desarrollo de este trabajo.

Software científico como una categoría de software.

El desarrollo de un sistema de software científico no sólo depende de la habilidad y experiencia que los programadores posean; un software de esta índole va más allá.

Es necesario que el o los programadores tengan los fundamentos teóricos, se apropien y dominen los algoritmos que se implementarán, además de tener la capacidad de crear y/o diseñar nuevas soluciones para su aplicación.

Es fundamental que el equipo de desarrollo tenga un periodo de entrenamiento, que va desde la investigación de sistemas de software desarrollados previamente, adquisición de bibliografía, hasta cursos impartidos por investigadores y/o especialistas en el dominio de aplicación.

Es necesario establecer el concepto de un *Proceso de Desarrollo de Software Científico*; los procesos comunes para el desarrollo de software no contemplan este periodo de entrenamiento que requiere el equipo de desarrollo para un proyecto de software científico y que tiene un

impacto directo en la asignación de roles, tareas y tiempos de implementación al equipo de desarrollo.

Ventajas de Data SOMining.

Hay que resaltar que la ventaja primordial que pueda proveer este sistema es gracias al sustento que brinda la eficiente e innovadora metodología ViBlioSOM®, metodología en la cual se basó el desarrollo de Data SOMinning. Por ejemplo, ViBlioSOM® utiliza la red neuronal SOM, la cual ha probado ser de gran utilidad para resolver problemas de minería de datos. Ésta ha sido útil particularmente en la organización creativa de información, el descubrimiento de conocimiento y la visualización de información.

La metodología ViBlioSOM®, y por ende Data SOMinning, es muy útil para realizar análisis de correlación entre variables o datos complejos y en la clasificación de información. Las ventajas alcanzadas con este método consisten en que ha permitido organizar visualmente la información bibliométrica y de esta manera percibir la estructura del conjunto de datos y profundizar en su análisis.

Gracias a la automatización obtenida al aplicar ViBlioSOM® con Data SOMinning se permite enriquecer el procesamiento, visualización y análisis de los indicadores bibliométricos, con una metodología propia. Este método puede ser aplicado a cualquier campo del saber y tiene un vínculo muy estrecho con los procesos de inteligencia empresarial, vigilancia científico-tecnológica, gestión del conocimiento y evaluación de proyectos. Igualmente, el método puede ser aplicado en servicios bibliotecarios e informativos y en observatorios de ciencia y tecnología.

La interfaz de Data SOMinnig contiene todas las funciones utilizadas dentro de la metodología ViBlioSOM®, desde la adquisición de datos a partir de MedLine hasta la visualización de mapas de conglomerados de los datos entrenados mediante la red SOM. De esta manera el usuario no requiere de algún otro tipo de software, incrementando de manera significativa su productividad.

Las tablas generadas por el usuario son almacenadas en archivo con formato `XML`, por lo que el usuario puede hacer uso de la información generada en el sistemaen otra aplicación e incluso modificar la información que contienen los archivos de forma manual. Hasta el momento se han realizado pruebas que han generado matrices de hasta 441 renglones y 441 columnas; cabe recordar que Excel sólo permite 256 columnas por hoja de cálculo, lo cual significa una seria limitación.

Los archivos `XML` de tesauro tienen un formato sencillo para el usuario, de esta forma, es posible que se editen estos archivos desde la interfaz o desde el archivo fuente. Estos archivos de tesauro se incrementan de forma dinámica conforme a las actualizaciones realizadas por los usuarios, de esta forma dichos archivos son altamente reutilizables para experimentos posteriores decrementando el tiempo de uso y aumentando la eficiencia.

Los parámetros de entrenamiento de la red SOM son claros para el usuario, además de ofrecer más opciones en cuanto a la métrica y la función de vecindad a utilizar.

Las visualizaciones generadas pueden ser almacenadas en archivo con formato `jpeg`, lo cual garantiza su manipulación a partir de cualquier aplicación para procesar gráficos, sin la necesidad de contar con el sistema de software Data SOMinning.

La posibilidad de almacenar resultados parciales o finales como parte de un proyecto de Data SOMinning, da la oportunidad de abortar el sistema sin la necesidad de rehacer los cálculos hasta ese momento obtenidos.

Desventajas de Data SOMining.

Como se mencionó anteriormente, Data SOMining está implementado en C# bajo la plataforma de desarrollo de .NET, por lo tanto, el sistema sólo puede ser ejecutado en sistemas operativos *Microsoft Windows XP* (el sistema no ha sido probado con versiones anteriores del sistema operativo). Ésto representa una desventaja para aquellos usuarios que utilizan algún otro sistema operativo y que requieran de utilizar el sistema.

Cabe mencionar que es necesario adquirir la licencia de uso de Visual Studio .NET, debido a que contiene todas las bibliotecas disponibles del lenguaje, así como versiones estables del compilador. Una posible opción fue utilizar el compilador de Mono para C#, sin embargo, éste aún no está liberado por completo, además de estar en fase de desarrollo de algunas bibliotecas como las de los componentes gráficos, las cuales representan un gran beneficio para este tipo de aplicaciones.

Hasta el momento, Data SOMining sólo puede leer y recuperar la información de archivos `txt` y `XML` con el formato definido por PubMed. El usuario únicamente puede acceder a estos archivos originales vía Internet a través del portal PubMed o por medio de Data SOMinning. Sin embargo, de esta forma es posible acceder a todos los registros contenidos en la base de datos MedLine.

Data SOMinning se encuentra en periodo de evaluación. Actualmente el grupo del Instituto Finlay que colabora con el grupo del laboratorio se encuentra realizando diversas pruebas; cabe señalar, que son usuarios que han utilizado la metodología ViBlioSOM®.

Trabajo a futuro.

Uno de los dos principales objetivos es añadir en el módulo de adquisición de datos, otras bases de datos que permitan el acceso a través de la interfaz. Al mismo tiempo, integrar nuevos módulos de lectura y recuperación de registros de estas nuevas bases de datos. Este punto es de gran interés, ya que al añadir nuevas fuentes de datos, el campo de aplicación de Data SOMinnig se vuelve más amplio.

Es importante obtener la opinión del usuario en cuanto el diseño dinámico de la interfaz, ya que éste es un factor primordial para el mismo.

Hasta el momento, la generación de los mapas se realiza con las bibliotecas nativas de

C#. En este sentido, habrá que investigar más acerca de nuevas bibliotecas y herramientas de visualización que aporten mayor interactividad con el usuario.

Sería de suma importancia evaluar la opción de migrar Data SOMinning bajo plataforma Unix Like, debido a las diversas ventajas de administración de recursos que ofrecen dichos sistemas. Para ello, podemos pensar en dos posibles opciones:

1. Utilizar el compilador de Mono para C# y adaptar las clases ya desarrolladas a las bibliotecas disponibles del compilador en el mejor de los casos, ya que existen bibliotecas para C# en .NET, que aún no son desarrolladas en su totalidad para Mono. Para este caso, las clases deberán ser reimplementadas con bibliotecas nativas para ambos compiladores.

2. Desarrollar Data SOMinning en otro lenguaje multiplataforma (Java, C++ o C).

Ambas consideraciones tienen un impacto negativo para el desarrollo, pues parte del código implementado sería desechado, sin mencionar el costo en tiempo y recursos que tendría el hecho de volver a implementar el sistema en otro lenguaje.

La concepción de Data SOMinning como un sistema de software multiplataforma, es una tarea importante. Bajo una plataforma de software libre, es posible eliminar costos de licencias de ambientes de desarrollo y compiladores, de implantación, mantenimiento, seguridad e interoperabilidad; sería posible llevar a cabo un proceso de corrección de errores y soporte, más dinámico y eficaz.

Una de las mejores opciónes para llevar a cabo este objetivo, es orientar las versiones posteriores de Data SOMinning al *desarrollo de componentes*, con el fin de construir un framework para minería de datos disponible para cualquier plataforma.

De esta forma, el código de esta primera versión de Data SOMinning, puede ser revisado para identificar partes del código que pueda ser reutilizado. Al mismo tiempo, la construcción de un framework, mitiga de forma considerable los costos de mantenimiento y escalabilidad.

Otro de los objetivos es añadir nuevos métodos y algoritmos matemáticos para la minería de datos. De inicio puede pensarse en las distintas variantes de la red neuronal SOM. Con esto se intentará abarcar un mayor número de áreas en las cuales sea de gran apoyo el uso de dicha herramienta. A su vez, se podrán realizar comparaciones entre los resultados obtenidos de los distintos algoritmos y de esta manera poder decidir cuál de ellos es el mejor para cada uno de los casos de estudio.

Bibliografía

[AA02] R. J. Arencibia and J. A. Araújo. Informetría, bibliometría y cienciometría: aspectos teórico-prácticos. *ACIMED*, 2002.

[Are03] G. Arellano. Integración del contexto técnico y tecnológico al proceso de desarrollo para la generación de software con calidad. Master's thesis, Posgrado en Ciencia e Ingeniería de la Computación, Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas, UNAM, 2003.

[CGMdlE+05] H. Carrillo, M. V. Guzmán, N. Martínez de la Escalera, J. L. Jiménez, E. Valencia, and E. Villaseñor. *ViBlioSOM: Aplicaciones en MedLine*. Laboratorio de Dinámica No Lineal, Facultad de Ciencias, UNAM, 2005.

[Cha04] O. G. Chaviano. Algunas consideraciones teórico-conceptuales sobre las disciplinas métricas. *ACIMED*, 2004.

[DdC] Medicina II de la Universidad de Viena. Departamento de Cardiología. The interpretation of thallium-201 scintigrams (heart scans) with respect to coronary artery disease.

[FPSS96a] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Knowledge discovery and data mining: An overview. *American Association for Artificial Intelligence.*, 1996.

[FPSS96b] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Knowledge discovery and data mining: Towards a unifying framework. *Proceedings of the Second International Conference on Knowlege Discovery and Data Mining*, 1996.

[HM00] J. R. Hilera and V. Martínez. *Redes Neuronales Artificiales: Fundamentos, modelos y aplicaciones*. RA-MA editorial, 2000.

[Hol96] Jaakko Hollmén. Process modeling using the self-organizing map. Master's thesis, Helsinki University of Technology, 1996.

[Hum00] H. Humphrey. *Introduction to Team Software Process: SEI Series in Software Engineering*. Addison Wesley, 2000.

[Juh00] Vesanto Juha. *Using SOM in Data Mining*. PhD thesis, Helsinki University of Technology, 2000.

[LPM96] S. Lesteven, P. Poinçot, and F. Murtagh. Neural networks and information extraction in astronomical information retrieval. *Vistas in Astronomy*, 1996.

[MC03] CA. Macías Chapula. Papel de la informetría y la cienciometría y su perspectiva nacional e internacional. `http://www.infomed.sld.cu/revistas/aci/vol9s01/sci06100.htm`, 2003.

[MS02] B. Martín and A. Sanz. *Redes Neuronales y Sistemas Difusos*. RA-MA editorial, segunda edition, 2002.

[OR99] J. Ong and S. Raza. Data mining using self-organizing kohonen maps: A technique for effective data clustering & visualisation. *International Conference on Artificial Intelligence*, 1999.

[Rau96] Rojas Raul. *Neural Networks a systematic introduction*. Springer-Verlag New York, Inc., 1996.

[Sam97] Kaski Samuel. *Data exploration using Self-Organizing Maps*. PhD thesis, Helsinki University of Technology, 1997.

[SGC02] G. Sotolongo, M. V. Guzmán, and H. Carrillo. Vibliosom: Visualización de información bibliométrica mediante el mapeo autoorganizado. *Revista Española de Documentación Científica*, 2002.

[Spi] E Spinak. *Diccionario enciclopédico de Bibliometría, Cienciometría e Informetría*. UNESCO - CII/II.

[SSG] G. Sotolongo, C. A. Suárez, and M. V. Guzmán. Modular bibliometrics information system with propietary software. In *Proceedings of the Seventh International Society for Scientometrics and Informetrics. Universidad de Colima, México.*

[SSGC] G. Sotolongo, C. A. Suárez, M. V. Guzmán, and H. Carrillo. Mining informetrics data with self-organizing maps. In *Proceedings of the 8th International Society for Scientometrics and Informetrics. Sydney, Australia.*

[Teu98] Kohonen Teuvo. Self-organizing maps. *Neurocomputing*, 1998.

[Teu00] Kohonen Teuvo. *The Self-Organizing Map*. Springer-Verlag, third edition, 2000.

[Tim97] Honkela Timo. *Self-Organizing Maps in natural language processing*. PhD thesis, Helsinki University of Technology, 1997.

[VA00] J. Vesanto and E. Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 2000.

[VHAP00] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas. Som toolbox for matlab. Technical report, 2000.

[WDA+02] J. Wang, J. Delabie, H. Asheim, E. Smeland, and O. Myklebost. Clustering of the som easily reveals distinct gene expression patterns: results of reanalysis of lymphoma study. *BioMed Central Bioinformatics*, 2002.

[WI97] S.M. Weiss and N. Indurkhya. *Predictive Data Mining: A practical guide*. Morgan Kaufmann, 1997.

[WP98] Wright and Peggy. El descubrimiento del conocimiento en las bases de datos: herramientas y técnicas. Technical report, 1998.