UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

" APLICACIÓN DE TÉCNICAS DE MINERÍA DE DATOS PARA UN SISTEMA DE SOPORTE DE DECISIONES "

T E S I S

PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACIÓN

Q U E P R E S E N T A :

CARLOS FLORES CEBALLOS

ASESOR DE TESIS: ING. SALVADOR PEREZ VIRAMONTES

México D. F.

Octubre 2003

Agradecimientos

A Dios

Por haberme permitido tener el entendimiento, paciencia y tenacidad para lograr esta meta, una de las más grandes hasta ahora. Y le ruego que el camino que falta por recorrer sea aún mejor.

A mis papás

Ya que sin su apoyo, sus consejos y sus bendiciones me hubiera sido infinitamente más difícil este camino. Gracias porque nunca colocaron límites para mis estudios, y cada vez pedían algo mejor que lo anterior. Espero estén orgullosos de sus tres hijos, porque nosotros sí lo estamos de ustedes.

A mis hermanos

Porque aunque nunca hemos tenido la oportunidad de hablar de esto, el ejemplo y los estándares que impusieron en mi vida, como ejemplos a seguir y mejorar, fueron muy importantes, como gran empuje para continuar tal cual ustedes lo han hecho.

A ti Lyzzette

Aunque nunca lo he dicho, eres la inspiración más grande en mi vida para seguir esforzándome día con día y así ser mejor persona de lo que fui ayer. Gracias por tu apoyo incondicional en este tiempo y sabes que también cuentas conmigo para cualquier cosa que necesites. Espero que todas las promesas y sueños, tanto individuales como en pareja se cumplan siempre y de la mejor manera. Recuerda que te amo y será *2VDS.

A las personas que he conocido en los últimos años y que me han ayudado

- Thelma Álvarez Murillo, gracias por ser mi amiga desde hace diez años, por haberme ayudado cuando te necesite y saber que tan solo esperas lo mismo de mi parte, ser incondicional en esta amistad.

- Laura Liyen Galicia Peñalosa, gracias por haberme permitido ser parte de esa gran familia de la Dirección de Sistemas y así poder aprovechar todas las oportunidades que vienen integradas.

- Salvador Pérez Viramontes, gracias por todo el tiempo de análisis, estudio, búsqueda y mejora de todo este trabajo que culmina una gran etapa de mi vida.

- Martha Elena Ramírez Bustos, gracias por haber estado pendiente de cada una de las acciones de mis estudios en estos últimos años y siempre haber estado con una sonrisa y tiempo suficiente para escucharme.

- Claudia Reyes, por darme la original y única oportunidad de ser mi amiga desde hace ya algunos años, espero que todas tus metas las logres como hasta ahora.

- Luis Vázquez González, por toda la paciencia que haz tenido para conmigo, además de toda la ayuda prestada sin pedir más que buenos resultados a cambio. Gracias por tu experiencia compartida.

- Semiramis Zaldivar Granada, por los innumerables consejos, apoyos y comentarios, así como el seguimiento a mi desarrollo y desempeño profesional, que sin duda han sido y serán una gran herramienta para mi crecimiento profesional y personal. Con toda sinceridad te repito que eres mi amiga. Gracias.

Índice general

Capítulo		
	Introducción	6
I	**La minería de datos**	**7**
	Data Warehousing	8
	Información oculta en los datos	10
	¿Qué es y que no es la MD?	11
	Minería de datos frente a OLAP y DSS	14
	¿Qué se puede esperar?	16
	Objetivo de la minería de datos	20
	Metodología de la minería de datos	24
II	**Data Warehouse (DW)**	**33**
	Introducción al concepto de data warehouse	35
	Sistemas de información	36
	Características de un data warehouse	37
	Estructura de los datos del data warehouse	43
	Arquitectura de un data warehouse	48
	Transformación de datos y metadata	51
	Medios de almacenamiento para información antigua	52
	Usos del data warehouse	53
	Excepciones en el data warehouse	54
	Consideraciones previas al desarrollo de un DW	55
	Sistemas de gestión de bases de datos	58
	Consideraciones para la implementación de un DW	63
	Software en un data warehouse	64
	Elección de herramientas de análisis de datos	68
III	**La preparación de los datos (DP)**	**70**
	Entendimiento de problemas	72
	Algunas formas de preparación de datos	74
	Salida de la preparación de datos	74
	Modelos activos y modelos pasivos	75
	La naturaleza del mundo y su impacto en los datos	81
	Tipos de mediciones	82
	Rol de las columnas en la minería de datos	86
	Datos para la minería de datos	87
	Series de tiempo	90
IV	**Análisis de las técnicas de la minería de datos**	**91**
	Análisis de la canasta de mercado (MBA)	94
	Aplicación de la técnica de MBA	95
	Generación de reglas de asociación	98

Razonamiento basado en la memoria (MBR)..... 102
¿Cómo trabaja el MBR?..... 102
Función de distancia..... 103
Función de combinación..... 105
Detección automática de clusters (ACD)..... 107
Método de las k-medias..... 110
Algoritmo de aglomeración..... 112
Análisis de asociación(LA)..... 115
Teoría de grafos..... 115
Árboles de decisión(DT)..... 120
CART..... 123
C4.5..... 125
CHAID..... 128
Redes neuronales artificiales (ANN)..... 131
Estructura de una red neuronal..... 133
Elección de datos para el entrenamiento de la red..... 140
Interpretación de los resultados..... 142
Algoritmos genéticos (GA)..... 145
Clases de algoritmos genéticos..... 146
Elementos de un algoritmo genético..... 148
Operadores genéticos..... 149
Diferencias entre AG y otros métodos de optimización..... 151

V Programas y código fuente **155**

Herramientas comerciales..... 155
Programa de las k-medias (PERL)..... 156
Programa de árboles de decisión (JAVA)..... 160
Programa de matriz de co-ocurrencia (PERL)..... 165
Consultas SQL, cláusula SELECT (SQL3)..... 168
Ejemplo y algoritmo de redes neuronales..... 174
Ejemplo y algoritmo de algoritmos genéticos..... 178
Ejemplo de razonamiento basado en memoria..... 182

Apéndices

Conclusiones y comentarios..... 187
Bibliografía..... 190
Glosario..... 191

Introducción

En este trabajo se intenta mostrar los aspectos más importantes de la minería de datos como una solución ante los problemas más comunes que afronta cualquier organización, cuando sus datos almacenados no han sido utilizados como una fuente invaluable de información oculta, por lo que, tan solo se ha limitado a su almacenamiento pero no a su manipulación como posible fuente de retroalimentación.

Los alcances de esta tesis son:

- La explicación de las bondades de la minería de datos, como una posible solución cuando no existe información suficiente que sirva como soporte para la toma de decisiones de la organización.
- Un análisis de cada una de las siete técnicas de la minería de datos propuestas; detallando las circunstancias, consideraciones y resultados involucrados antes, durante y después de la implementación de cada técnica.
- La construcción de una serie de puntos para el correcto uso y exploración de un data warehouse. Proponiendo al data warehouse, como el almacén y administrador óptimo de los datos que serán utilizados en la minería de datos.
- El análisis de las causas y consecuencias de la existencia de datos "sucios" en la base de datos; además, también se analizan las posibles soluciones existentes para cada caso mencionado.
- Proponer una serie de programas o ejemplos, para la implementación de cada de las técnicas de la minería de datos, los cuales fueron diseñados y creados por el autor de esta tesis, bajo la plataforma Unix y el uso de los paquetes WEKA de la Universidad de Nueva Zelanda.

Dadas las limitantes y poblemáticas encontradas en la elaboración de esta tesis, no fue posible la realización completa de una minería de datos, por lo que tan solo se logró hacer la implementación de los programas en datos aislados.

Capítulo I

LA MINERÍA DE DATOS

Con la denominada sociedad de la información se está produciendo un fenómeno curioso. Día con día se multiplica la cantidad de datos almacenados. Sin embargo, contrariamente a lo que pudiera esperarse, esta explosión de datos no supone un aumento de nuestro conocimiento, puesto que resulta imposible procesarlos con los métodos clásicos. La mayoría de las empresas multinacionales generan más información en una semana que cualquiera persona pudiera leer en toda su vida, e incluso las pequeñas empresas generan un volumen de datos que no son capaces de manejar. De modo que actualmente nos enfrentamos a la paradoja de que, cuantos más datos están disponibles, menos información tenemos.

Para superar este problema, en los últimos años han surgido una clase de técnicas que facilitan el procesamiento avanzado de los datos y permiten un análisis detallado de los mismos en forma automática. La idea clave es que los datos contienen más información oculta que la que se ve a simple vista.

Los datos, origen de la información

Hoy en día, y está claro que se trata de una tendencia válida para los próximos años, el almacenamiento de la información es algo sencillo y barato. Los sistemas informáticos cada vez tienen una capacidad mayor, y lo que ahora es normal encontrar en una computadora personal, quedara anticuado dentro de unos meses.

Este incremento de los sistemas de almacenamiento tiene un comportamiento que es realmente interesante: es poco costoso guardar datos del funcionamiento de los procesos, o de los sistemas de venta, o de los clientes, etc., por lo que las bases de datos (en el sentido más amplio del término) crecen hasta límites insospechados.

Cuando se decide iniciar el proceso de almacenamiento de datos, se suele hacer con la intención de analizarlos posteriormente. Sin embargo, cuando llega ese momento, el análisis que se realiza suele ser bastante superficial y guiado por los resultados que se esperaban encontrar al analizarlos. Lo normal es utilizar algún paquete estadístico para localizar correlaciones entre variables, establecer medias y varianzas e intentar modelar de esta forma la información.

Sin embargo, en la montaña de datos existe información que no puede ser encontrada con los procedimientos habituales de trabajo. La minería de datos ayuda a dar un paso más en ese análisis sacando a las relaciones ocultas entre los datos: información desconocida que pueda ayudar a gestionar mejor el negocio o proceso.

Estructura de los datos

Para poder analizar los datos fiablemente, es necesario que exista una cierta estructuración y coherencia entre los mismos. Si el responsable de almacenamiento de la

información ha sido siempre la misma persona, es posible que una parte de este problema este resuelto. Sin embargo, en general no se da esa situación, sino que, más bien al contrario, son muchas las personas que en distintos departamentos y a lo largo del tiempo han ido creando archivos con diferentes tipos de datos.

Surge entonces la necesidad de conjuntar los distintos archivos y bases de datos de manera que se puedan utilizar para extraer conclusiones. Se enfrentarán diferentes tipos de problemas:

- Diferentes tipos de datos representando el mismo concepto: un ejemplo que ha provocado uno de los mayores problemas informáticos es la representación de la fecha, donde el año se puede guardar con 2 ó 4 dígitos.
- Diferentes claves para representar el mismo elemento: un mismo cliente puede ser representado por un código de cliente propio o por su NIP.
- Diferentes niveles de precisión al representar un dato: los números reales no siempre se almacenan de la misma forma, y es posible que esto genere algún problema.
- Entre otros más, sin mencionar la incompatibilidad del formato con el que se pueden guardar los datos almacenados.

Con esto se puede ver que la situación no es sencilla, y se agrava cuando los diferentes archivos se encuentran en sistemas informáticos y soportes diferentes.

Es cierto que cada una de estas fuentes de datos puede ser manejada por separado. Seguro que hay quien opina que los datos están en diferentes archivos porque representan informaciones y procesos distintos, y que no tiene sentido estructurar la información más allá de lo que ya está. Y es posible que si así se hace se encuentre información útil. Pero no es menos cierto que se está quitando a la base de datos misma la posibilidad de descubrir un conocimiento que va más allá de cada una de las secciones del negocio: un conocimiento que representa la interacción entre diferentes procesos, que es, precisamente, donde se encuentra la información más valiosa.

Data Warehousing

El mecanismo más habitual para estructurar la información de un negocio es hacer uso de un *Data Warehouse*. Las definiciones más habituales de este concepto son:

- Almacén de datos. Plataforma que concentra la información de interés de toda la empresa.
- Sistema que permite el almacenamiento en un único entorno de la información histórica e integrada proveniente de los distintos sistemas de la empresa que refleja los indicadores clave asociados a los negocios de la misma.
- Sistema de información orientado a la toma de decisiones empresariales que, almacenando de manera integrada la información relevante del

negocio, permite la realización de consultas complejas con tiempos de respuesta cortos.

- Sistema orientado a dar información en términos de negocio en vez de datos en términos operacionales.

Como se puede apreciar, las palabras más empleadas son: información de interés, negocio e integración. De su conjunto se puede expresar que el data warehouse es un almacén estructurado de la información clave del negocio, que integra datos provenientes de todos los departamentos, sistemas, etc., y que permite analizar el funcionamiento de la compañía para tomar decisiones sobre su gestión.

No se trata de una simple agregación de las diferentes bases de datos. Es importante destacar que hay algunas diferencias de concepto a éstas y a su forma de uso.

Una base de datos operativa almacena la información de un sector del negocio, se actualiza a medida que llegan datos que deban ser almacenados y se opera mediante los cuatro mecanismos clásicos "Añadir-Eliminar-Modificar-Imprimir".

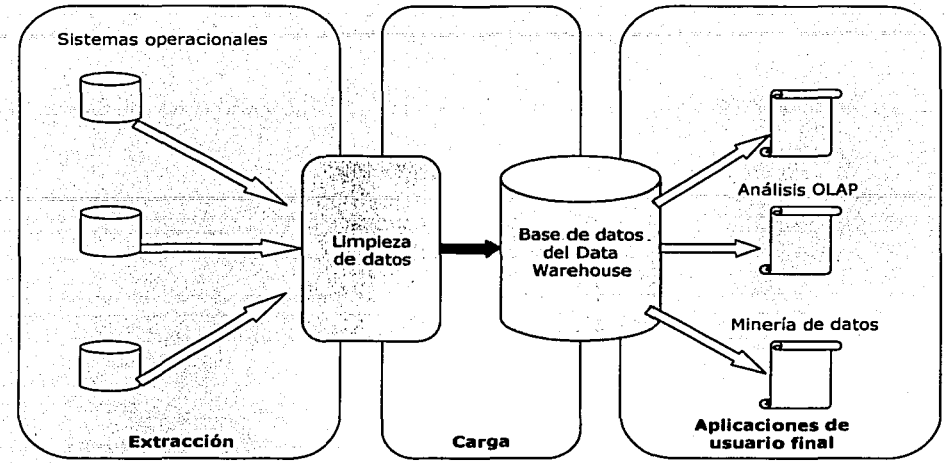
Clásicamente se orienta hacia la elaboración de informes periódicos, suele manejar pequeños volúmenes de datos y su entorno está dimensionado para muchas transacciones (gran cantidad de actualizaciones).

En cuanto al data warehouse, su actualización se realiza a intervalos regulares (típicamente una al día) dentro de un proceso controlado, y tras realizar un preproceso de los datos que se van a almacenar. Su orientación es hacia la consulta del estado del negocio.

Características del data warehouse:

- Se ofrece información bajo demanda (análisis libre mediante el uso de herramientas de generación de informes que usan el Data warehouse).
- Refleja el modelo de negocio, frente al modelo de proceso.
- Almacena grandes volúmenes de datos (información histórica e integración de datos de múltiples aplicaciones). Está dimensionado para consultas largas y elaboradas.
- Actualizaciones controladas y no eliminación de datos (el Data Warehouse contiene toda la historia de la compañía).

La estructura de esta gran base de datos es multidimensional, con diferentes puntos de vista que reflejan los distintos aspectos del negocio. Así los responsables de los productos pueden analizar su evolución a lo largo del tiempo en diferentes sectores y localización geográfica. Sobre los mismos datos, los responsables de grandes cuentas pueden obtener información sobre los tipos de productos que se han vendido, por regiones, a lo largo del tiempo. En la siguiente figura se muestra la estructura de un data warehouse.



Estructura de un data warehouse

Una técnica muy usada en un Data Warehouse es el cubo de datos, del que se pueden extraer diferentes "rodajas" o puntos de vista, se pueden analizar una parte concreta, o estudiar el conjunto global.

Cuando se tiene una estructura de Data Warehouse, pero se adapta sólo a un sector de la empresa, o para un fin concreto, se denomina un Data Mart. Los Data Marts puede extraerse del Data Warehouse de la empresa, aunque también es posible que el Data Warehouse se construya a partir de los Data Marts que se hayan ido diseñando e implantado en los diferentes departamentos. Este segundo enfoque es el que se utiliza cuando se comienza por aplicar estas técnicas en las áreas del negocio y no en su globalidad.

Información oculta en los datos

Si se va almacenando información relevante del negocio en un sistema que acumula y acumula datos sin parar, un análisis razonable puede permitir descubrir tendencias, localizar grupos de datos con comportamiento homogéneo, establecer relaciones, etc.

Esta información está oculta en los datos y será necesario utilizar todas las técnicas al alcance para obtenerla. El objetivo que se plantea es localizar relaciones entre atributos del Data Warehouse. Estas relaciones podrían ser del tipo:

- Para una tienda: Más del 60% de las personas que adquieren queso fresco también compran algún tipo de mermelada.
- Para una compañía aérea: Muchos usuarios que hacen vuelos en menos de tres días a Berlín alquilan un coche en el aeropuerto.

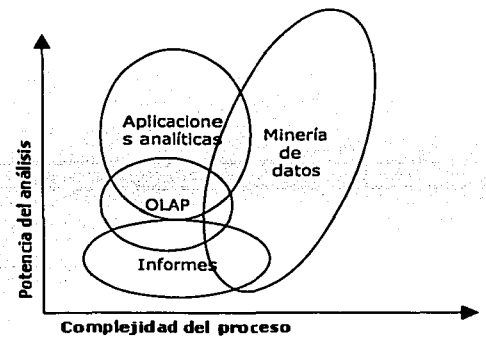
- Para una compañía de telefonía: Durante el mes siguiente al lanzamiento de una campaña de descuento en llamadas internacionales por parte de una compañía de la competencia, los pequeños clientes redujeron su consumo en este sector, mientras que los grandes lo mantuvieron.

Esta información puede ser extraída haciendo uso de diversas técnicas y ninguna de ellas debe ser depreciada, sino agregada al resto para obtener mejores resultados. Sin embargo, en este trabajo se centra la atención en la minería de datos y en las ventajas que puede aportar.

¿Qué es y que no es la minería de datos?

La minería de datos puede definirse como la *extracción no trivial de información implícita, previamente desconocida y potencialmente útil, a partir de los datos.* Para conseguirlo hace uso de diferentes tecnologías que resuelven problemas típicos de agrupamiento automático, clasificación, asociación de atributos y detección de patrones secuenciales. La minería de datos es, en principio, una fase dentro de un proceso global denominado *descubrimiento de conocimiento en bases de datos* (Knowledge Discovery In Databases, KDD), aunque finalmente haya adquirido el significado de todo el proceso para ligar la fase de extracción con la fase de conocimiento.

Es habitual que los expertos en estadística confundan la minería de datos con un análisis estadístico de éstos (afirmaciones de este tipo pueden encontrarse empresas dedicadas al procesamiento estadístico que venden sus productos como herramientas de minería de datos).



La minería de datos en comparación con otras técnicas de soporte de toma de decisiones.

La diferencia fundamental entre ambas técnicas es muy clara: para conseguir una afirmación como la que ha sido utilizada en el ejemplo anterior (más del 60% de las personas que adquieren queso fresco también compran algún tipo de mermelada)

utilizando un paquete estadístico, es necesario conocer a priori que existe una relación entre el queso fresco y la mermelada, y lo que se realiza con el entorno estadístico es una cuantificación de dicha relación.

En el caso de la minería de datos el proceso es muy distinto: la consulta que se realiza a la base de datos (al Data Warehouse) busca relaciones entre parejas de productos que son adquiridos por una misma persona en una misma compra. De esa información, el sistema deduce, junto a otras muchas, la información anterior. Como se puede ver, en este proceso se realiza un acto de descubrimiento de conocimiento real, puesto que no es necesario ni siquiera sospechar la existencia de una relación entre estos dos productos para encontrarla.

Definición, caracterización y estructura del problema

La evolución de la tecnología ha facilitado y automatizado en gran medida las tareas de análisis de información. Cada paso en esta evolución se apoya en los anteriores y cada un o de ellos ha supuesto un avance significativo para el usuario, que ha visto cómo cada progreso le abría nuevas posibilidades de análisis y aumentaba el nivel de abstracción de las consultas.

Para decidir cuál es la técnica más adecuada para una determinada situación, es necesario distinguir el tipo de información que se desea extraer de los datos. Según su nivel de abstracción, el conocimiento contenido en los datos puede clasificarse en distintas categorías y requiere una técnica más o menos avanzada para su recuperación.

Conocimiento evidente. Información fácilmente obtenible con una simple consulta (SQL). Un ejemplo de este tipo de conocimiento es una pregunta como "¿Cuáles fueron las ventas en España el pasado marzo?" o "¿Cuál es la edad media de los clientes?"

Conocimiento multi-dimensional. El siguiente nivel de abstracción consiste en considerar los datos con una cierta estructura. Por ejemplo, en vez de considerar cada transacción individualmente, las ventas de una compañía puede organizarse en función del tiempo y de la zona geográfica, y analizarse con diferentes niveles de detalle (país, región, localidad, ...).

Técnicamente se trata de reinterpretar una tabla con *n* atributos independientes como un espacio *n-dimensional*, lo que permite detectar algunas regularidades difíciles de observar con la representación monodimensional clásica. Este tipo de información es la que analizan las herramientas OLAP, que resuelven de forma automática cuestiones como "¿Cuáles fueron las ventas en España el pasado marzo?" Aumentar el nivel de detalle: mostrar las de Madrid.

Conocimiento oculto. Información no evidente, desconocida a priori y potencialmente útil, que puede recuperarse mediante técnicas de minería de datos, como reconocimiento de regularidades. Esta información es de gran valor, puesto que no se conocía y se trata de un descubrimiento real de nuevo conocimiento, del que antes no se

tenía idea, y que abre una nueva visión del problema. Un ejemplo de este tipo sería "¿Qué tipo de clientes se tienen? ¿Cuál es el perfil típico de cada clase de usuario?"

Las técnicas disponibles para extraer la información contenida en los datos son muy variadas y cada una de ellas es complementaria del resto, no excluyentes entre sí. Cada técnica resuelve problemas de determinadas características y, para extraer todo el comportamiento oculto, en general será necesario utilizar una combinación de varias.

La mayor parte de la información de interés contenida en una base de datos, aproximadamente el 80%, corresponde a conocimiento superficial, fácilmente recuperable mediante consultas sencillas con SQL. El 20% restante corresponde a conocimiento oculto que requiere técnicas más avanzadas de análisis para su recuperación. Estas cifras pueden dar la falsa impresión de que la cantidad de información recuperable mediante técnicas de minería de datos es despreciable. Sin embargo, se trata precisamente de información que puede resultar de vital importancia para la empresa y que no se puede desdeñar.

Básicamente, y como se ha comentado, la clave que diferencia la minería de datos respecto de las técnicas clásicas es que el análisis que realiza es exploratorio, no corroborativo. Se trata de descubrir conocimiento nuevo, no de confirmar o desmentir hipótesis. Con cualquiera de las otras técnicas es necesario tener una idea concreta de lo que se esta buscando y, por lo tanto, la información que se obtiene con ellas esta condicionada a la idea preconcebida con que se aborde el problema. Con la minería de datos es el sistema y no el usuario el que encuentra las hipótesis, además de comprobar su validez.

La minería de datos, esencialmente, permite obtener a partir de los datos un *modelo* del problema que se analiza, bien sean las ventas de un artículo para mejorar la campaña de marketing, las características técnicas de un producto en control de calidad o un proceso industrial cuyo control se desea optimizar, por citar algunos ejemplos. El modelo obtenido permitirá simular el comportamiento del sistema real y obtener conclusiones aplicables en el día a día.

Uso de los resultados de la minería de datos

La minería de datos descubre relaciones en los datos, pero eso es sólo el principio. Son las personas, no las técnicas de minería de datos, las que toman las decisiones. El factor más importante en minería de datos es el conocimiento y la experiencia de dichas personas. Armadas con mejor información, pueden aplicar su creatividad y su propio criterio para tomar decisiones más acertadas y obtener mejores resultados.

Por muy buenos que sean los resultados obtenidos en un proyecto de minería de datos son totalmente inútiles si no se aplican en la práctica correctamente. Así, es inútil que se consiga un clasificador que diferencie perfectamente diversos tipos de clientes si no se tiene en cuenta dicha información en una campaña de marketing. O descubrir la influencia de una determinada variable en el rendimiento de un proceso si después no se

controla consecutivamente su valor. Las conclusiones de la minería de datos no son valiosas por si mismas, sino en la medida en que se apliquen para obtener resultados.

Es importante recordar que los responsables de dicha puesta en práctica no serán generalmente expertos en minería de datos. Un factor clave en el éxito de estos proyectos es presentar los resultados de una forma clara e inteligible, haciendo hincapié en la información realmente útil, teniendo siempre en cuenta sus destinatarios. Es así mismo fundamental, justificar adecuadamente dichas conclusiones, puesto que otro problema muy generalizado es la desconfianza que frecuentemente suscitan los sistemas automáticos. A menudo, es necesario un cambio de mentalidad para convencer a las personas involucradas del interés, utilidad y fiabilidad de la información obtenida gracias a la minería de datos.

Estas dificultades pueden ser superadas en gran medida si los responsables de la aplicación del proyecto han participado activamente en su desarrollo. Será mucha más sencillo convencer a una persona de la validez de las conclusiones obtenidas si ella misma ha aportado su conocimiento del proceso en estudio, o de su utilidad si fue el promotor del análisis. La colaboración de todos los usuarios implicados es fundamental para el éxito de un proyecto de minería de datos.

Minería de datos frente a OLAP y DSS

Los sistemas de ayuda a la decisión (DSS) son herramientas sobre las que se apoyan los responsables de una empresa, directivos y gestores, en la toma de decisiones. Para ello utilizan:

- Un Data Warehouse, en el que se almacena la información de interés para la empresa y,
- Herramientas de análisis multidimensional. (OLAP).

OLAP (On-Line Analytica Processing) se define como análisis rápido de información multidimensional compartida. El término OLAP aparece en contraposición al concepto tradicional OLTP (On-Line Transaccional Processing), que designa el procesamiento *operacional* de los datos, orientado a conseguir la máxima eficacia y rapidez en las transacciones (actualizaciones) individuales de los datos, y no a su análisis de forma agregada.

Las herramientas OLAP permiten navegar a través de los datos almacenados en el Data Warehouse y analizarlos dinámicamente desde una perspectiva multidimensional, es decir, considerando unas variables en relación con otras y no de forma independiente entre si y permitiendo enfocar el análisis desde distintos puntos de vista. Esta visión multidimensional de los datos puede visualizarse como un "cubo de Rubik", que puede girarse para examinarlo desde distintos puntos de vista, y del que se pueden seleccionar distintas "rodajas" o "cubos" dependiendo de los aspectos de interés para el análisis.

Los DSS permiten al responsable de la toma de decisiones consultar y utilizar de manera rápida y económica las enormes cantidades de datos operacionales y de mercado

que se generan en una empresa. Gracias al análisis OLAP, pueden verificarse hipótesis y resolverse consultas complejas. Además, en el curso del análisis, la interpretación de los datos puede dar lugar a nuevas ideas y enfoques del problema, sugiriendo nuevas posibilidades del análisis.

Sin embargo, el análisis OLAP depende de un usuario que plantee una consulta o hipótesis. Es el usuario el que lo dirige y, por tanto, el análisis queda limitado por las ideas preconcebidas que aquel pueda tener.

La minería de datos constituye un paso más en el análisis de los datos de la empresa para apoyar la toma de decisiones. No se trata de una técnica que sustituya los DSS ni el análisis OLAP, sino que los complementa, permitiendo realizar un análisis más avanzado de los datos y extraer más información de ellos.

Utilizando la minería de datos es el propio sistema el que descubre nuevas hipótesis y relaciones. De este modo, el conocimiento obtenido con estas técnicas no queda limitado por la visión que el usuario tiene del problema.

Las diferencias entre la minería de datos y OLAP radican esencialmente en que el enfoque desde el que se aborda el análisis con cada una de ellas es completamente distinto. Fundamentalmente:

- El análisis que realizan las herramientas OLAP es dirigido por el usuario, deductivo, parte de una hipótesis o de una pregunta del usuario y se analizan los datos para resolver esa consulta concreta. Por el contrario, la minería de datos permite razonar de forma inductiva a partir de los datos para llegar a una hipótesis general que modele el problema.
- Además, las aplicaciones OLAP trabajan generalmente con datos agregados, para obtener una visión global del negocio. Por el contrario, la minería de datos trabaja con datos individuales, concretos, descubriendo las regularidades y patrones que presentan entre si, generalizando a partir de ellos.

	OLAP	**MD**
Razonamiento	Deductivo	Inductivo
Trabaja con datos	Agregados	Concretos / individuales

Un ejemplo clarificara la diferencia entre ambas técnicas:
Una pregunta típica de un sistema OLAP/DSS seria: "El año pasado, ¿se compraron más camionetas en Durango o en Zacatecas?" La respuesta del sistema seria del tipo "En Durango se compraron 12,000 camionetas, mientras que, durante el mismo intervalo, en Zacatecas se compraron 10,000". Obviamente es una información interesante y útil, pero restringida por las hipótesis realizadas a priori.

En cambio, un problema típico para resolver utilizando minería de datos sería:
"Hallar un modelo que determine las características más relevantes de las personas que compran camionetas". A partir de los datos del pasado, el sistema de minería de datos

proporcionaría una respuesta del tipo: "Depende de la época del año y la situación geográfica. En invierno, los habitantes de Zacatecas que pertenecen a un cierto grupo de edad y nivel de ingresos probablemente comprarán más camionetas que la gente de las mismas características en Chihuahua".

Como puede verse, se trata de problemas distintos, de modo que según los objetivos perseguidos deberá utilizarse una técnica u otra. Además, puesto que sus conclusiones son complementarias, en general será conveniente combinar ambas para obtener los mejores resultados.

¿Qué se puede esperar?

La finalidad de cualquier proyecto de minería de datos puede resumirse en uno de estos dos:

- Ahorrar mejorando la eficacia de sus actividades, o bien,
- Ganar descubriendo nuevas fuentes de beneficios.

¿Cómo se llega a estos objetivos?. A partir de un conjunto de datos y un conjunto de técnicas se puede llegar a unas determinadas conclusiones. Pero, ¿cómo se traducen los resultados de un proyecto de minería de datos en beneficios tangibles para la empresa?. Básicamente, esos resultados suponen una mejora de la información disponible y será al aplicar dicha información cuando se obtengan los beneficios.

Los campos en los que pueden utilizarse estas técnicas son extremadamente variados: prácticamente en cualquier situación en la que se disponga de un conjunto de datos. A continuación se comentan algunas de las áreas más comunes en las que se ha aplicado frecuentemente la minería de datos, pero se trata simplemente de algunos ejemplos. En casi cualquier caso que se pueda imaginar es probable que la minería de datos pueda aportar importantes beneficios.

¿Parece una exageración?, tal vez no tanto. A modo de curiosidad: 28 de los 29 equipos que participan activamente en la liga de baloncesto profesional americana (NBA) utilizan técnicas de minería de datos para detectar patrones de comportamiento y relaciones entre variables del juego (por ejemplo, detectar que el jugador X realiza 90% de sus tiros de campo cuando el jugador Y juega de base), de forma que estas técnicas ofrecen nuevas perspectivas para modificar las tácticas de juego a fin de mejorar el rendimiento del equipo. Un análisis tradicional podría indicar que un jugador consigue el 70% de sus puntos en tiros de media distancia desde el lateral derecho.

En general; disponer de un modelo que permita simular el comportamiento y/o predecir la evolución de un sistema, un proceso, las ventas de un producto, etc., de forma suficientemente precisa supone una clara ventaja competitiva, permitiendo adelantarse y aprovechar oportunidades, así como prevenir problemas. Algunas de las aplicaciones más comunes son la que a continuación se enumeran.

Marketing

Este es uno de los campos donde los éxitos de la minería de datos son más conocidos. Cuanto más precisa sea la información que se tenga sobre los clientes, mayores posibilidades se tendrán de aumentar los ingresos y rentabilizar al máximo las acciones. El objetivo fundamental puede resumirse en determinar quién comprará qué, cuándo y dónde.

- Targeting: Se puede aumentar espectacularmente el porcentaje de respuesta a una campaña de marketing si se dirige a los objetivos adecuados. La minería de datos permite detectar entre los clientes potenciales, los que presentan una mayor probabilidad de responder a la campaña y dirigirla a ellos específicamente, con lo cual se consigue reducir drásticamente los costos.
- Fidelización de clientes: Conseguir un nuevo cliente o recuperar uno perdido resulta mucho más costoso que mantener uno que ya lo es. De ahí la rentabilidad de las campañas de fidelización de clientes, que detectan aquellos que parece más probable que se vayan a perder, permitiendo llevar a cabo iniciativas que eviten dicha perdida.
- La minería de datos también permite detectar nuevas oportunidades de mercado, comparando hábitos de consumo de diferentes clientes, por ejemplo, o determinando la ubicación más conveniente para un determinado negocio.

Predicción

Conocer a priori cómo evolucionará una variable en el futuro constituye una información muy valiosa y supone una indudable ventaja competitiva. Se trata de una herramienta de evidente interés tanto desde el punto de vista comercial, como en gestión o control de proceso.

A partir de los datos históricos almacenados y utilizando técnicas de minería de datos pueden elaborarse modelos que permitan estimar con precisión la evolución de una variable en el futuro. Disponer de esta información con tiempo suficiente permite adecuar la respuesta de forma óptima. Esto resulta útil en los campos más diversos:

- Detección de oportunidades.
- Prevención de problemas.
- Gestión optima del personal.
- Optimización de stocks.

Reducción de riesgos

La minería de datos permite construir sistemas de evaluación automática de riesgos, basados en la experiencia previa. Estos sistemas resultan de gran utilidad cuando la cantidad de casos a evaluar es excesiva para su procesamiento manual. El empleo de

técnicas de minería de datos ha aumentado la eficacia y fiabilidad de dichos sistemas, logrando un comportamiento más similar al de los expertos humanos.

Detección de fraudes

Aplicando técnicas de minería de datos, pueden obtenerse modelos que permitan descubrir posibles fraudes, basándose en la detección de comportamientos anómalos, en comparación con los datos registrados anteriormente.

Se puede encontrar aplicaciones concretas en operadores de telefonía o empresas de gestión de tarjetas de crédito. Estas compañías analizan el uso que los clientes hacen de sus servicios y puede localizar, de manera muy rápida, un uso fraudulento de los mismos.

Control de calidad

Existen numerosos ejemplos en los que se han aplicado técnicas de minería de datos para desarrollar sistemas automáticos de control de calidad. Estos sistemas suponen un considerable ahorro en el proceso productivo, puesto que facilitan:

- *Detección más precisa de productos defectuosos*. A menudo el control de calidad se realiza de forma manual y, por tanto, depende de una evaluación subjetiva por parte del personal responsable del mismo. El principal problema de este método es que el criterio de calidad no es estable sino que depende de la persona que realiza el análisis. La minería de datos permite desarrollar sistemas automáticos de control de calidad que discriminan los productos defectuosos con un alto grado de precisión y fiabilidad, según un criterio objetivo. Esto no solo evita el problema mencionado, además, al aumentar la exactitud de la evaluación se ahorran los costes derivados de las clasificaciones erróneas: productos defectuosos que se consideraron correctos por error y productos correctos, desechados por un exceso de precaución.

- *Localización precoz de defectos*. El control de calidad no sólo debe realizarse al final del proceso. Cuanto antes se detecte un fallo, menor será su impacto. Además de las ventajas de los sistemas automáticos, en este caso existe un problema añadido. A menudo no resulta fácil medir la variable que determina la calidad del producto en tiempo real o en la cadena de producción. En estos casos, es imprescindible utilizar técnicas de minería de datos para descubrir posibles relaciones que permitan detectar los fallos utilizando las variables disponibles durante el proceso.

- *Identificación de causas de fallos*. La minería de datos no solo resulta útil para discriminar los productos defectuosos. También ayuda a determinar los

fallos más frecuentes así como identificar las causas de los mismos. Esto permite adoptar medidas para evitarlos en el futuro.

- *Análisis no destructivo.* A menudo, para obtener la información que se necesita, hay que realizar un análisis destructivo. Un ejemplo típico es la evaluación de la resistencia de un material, medida que se establece forzándolo hasta que se rompe. Utilizando minería de datos es posible estimar con bastante exactitud el valor de este tipo de parámetros en función de otras características que sí pueden medirse sin destruir el producto. Esto permite controlar la calidad de todos los productos fabricados y no solo de una pequeña muestra, ya que no se destruyen con el examen.

Procesos industriales

Otra aplicación básica de la minería de datos en el entorno industrial, además del control de calidad, es el control de procesos. Estas técnicas permiten explotar la información disponible sobre un sistema o proceso y utilizar los modelos desarrollados (bien de un sistema o proceso global, o bien de una parte concreta del mismo) para:

- *Automatizar y optimizar el control del proceso.* En muchos sistemas se conoce el proceso suficientemente como para diseñar e implantar controladores a partir de análisis matemático del proceso. En otras ocasiones, esto no es posible, bien por que el proceso es enormemente complejo, bien porque no se dispone de todas las variables. En estas circunstancias, técnicas de minería de datos pueden ayudar a establecer relaciones entre las variables, y así diseñar los controladores adecuados.

- *Optimizar su rendimiento.* Los propios sistemas de aprendizaje pueden ser utilizados para adaptar los mecanismos de control de forma permanente, en función de los datos del proceso que se vaya recibiendo. De esta forma es posible optimizar el rendimiento del proceso, adaptando los controladores, en cada momento, a la situación de la planta.

- *Implantar programas de mantenimiento predictivo.* Uno de los problemas de todo equipo de mantenimiento de un proceso es establecer el calendario de reparaciones. Las reparaciones, limpiezas y ajustes programados suponen en muchos casos para el proceso productivo, con las consiguientes perdidas, no sólo de lo que se deba de producir, sino de los costes de parada y arranque de la cadena. Un análisis profundo de los datos de que se disponga puede hacer una planificación óptima de estas paradas, de manera que se minimice su impacto.

Objetivo de la minería de datos

La minería de datos es el proceso de exploración y análisis, por procesos semiautomáticos o automáticos de grandes volúmenes de información con el propósito de descubrir patrones y reglas ocultos.

Dadas las necesidades de cualquier disciplina por el incremento explosivo de la información almacenada, la minería de datos sirve para poder explotar la información con la que se cuenta en datos históricos, esto es, todos los registros que se han almacenado en un medio electrónico, para aprovechar oportunidades y poder descubrir errores.

El contexto de la minería de datos en los negocios

La minería de datos (extracción de patrones significativos y reglas a partir de grandes volúmenes de información) puede usarse en cualquier campo donde existen grandes cantidades de información y exista algo por aprender.

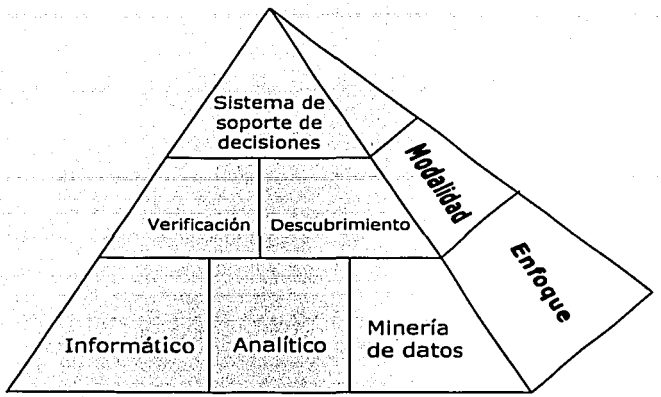
En los negocios, hay una definición explicita de que es lo que se quiere aprender, obtener un conocimiento resultante para obtener más ingresos en las áreas donde no han sido explotadas o no se ha descubierto su potencial.

En sentido académico, el conocimiento es considerado que tiene un valor intrínseco con una aplicación en cualquier parte. En el contexto de los negocios sin embargo, el conocimiento puede ser valorado en dos formas: puede incrementar los beneficios por una reducción de costos, o puede incrementar los beneficios por un aumento de los ingresos. Actualmente, hay una tercera forma: puede incrementar los precios existentes por mantener la promesa de incrementar los beneficios por medio de cualquiera de estos mecanismos.

La minería de datos es una herramienta de búsqueda

En una de las cosas en la que la minería de datos puede disminuir los costos es en el inicio del ciclo de vida de producción, durante la búsqueda y el desarrollo. La industria farmacéutica provee un buen ejemplo. Globalmente, esta industria gasta 13 billones de dólares en búsqueda y desarrollo de medicamento. El riesgo de minimizar costos sacrificando la calidad del producto, seria catastrófico por todos los componentes químicos que afectarían al organismo humano al ser ingeridos, causando grandes problemas en el ámbito mundial.

La figura que se encuentra en la siguiente pagina, muestra la forma en la que se estructura una plataforma de soporte de decisiones en una empresa y el papel de la minería de datos en ella.



Plataforma de soporte de decisiones

La minería de datos sirve para trabajar con seis **actividades principales** sobre la información:

- Clasificación
- Estimación
- Predicción
- Agrupación por afinidad o reglas de asociación
- Clustering
- Descripción y visualización

Las tres primeras tareas (clasificación, estimación y predicción) son ejemplos de minería de datos dirigida. En la minería de datos dirigida, la meta es usar los datos disponibles para construir un modelo que describa una variable particular de interés en términos del resto de los datos disponibles.

En la minería de datos no dirigida (agrupación por afinidad o reglas de asociación, clustering, descripción y visualización), no hay variables que se tomen como objetivo, la meta es establecer alguna relación entre todas las variables.

Clasificación

La clasificación consiste en examinar las características de un objeto asignándolo a una clase predefinida. La clasificación consiste en actualizar cada registro llenando un campo con una "marca" que lo identifica como integrante de una cierta clase.

La tarea de la clasificación se caracteriza por una bien construida definición de las clases, y un conjunto consistente de ejemplos preclasificados para entrenamiento. El objetivo es construir un modelo que pueda ser aplicado a datos sin clasificar para clasificarlos.

Ejemplos de las tareas de la clasificación incluyen:

- Clasificación de clientes de bajo, medio y alto riesgo en aplicaciones de crédito.
- Clasificación de cuando una línea telefónica casera es utilizada para acceso a Internet.
- Clasificación de clientes a segmentos predefinidos.

En todos estos ejemplos, hay un número limitado de clases bien conocidas y se espera poder asignar cualquier registro dentro de cualquiera de esas clases.

Estimación

Se usa para calcular un valor aproximado de una variable continua o discreta: autorización de tarjeta de crédito, hipotecas o autofinanciamiento.

Con ciertos datos de entrada, se puede usar la estimación para alcanzar con una variable desconocida algo como réditos ó balances de tarjetas de crédito. En la práctica, la estimación es utilizada para mejorar los resultados arrojados por la clasificación.

Un ejemplo es cuando un banco desea decidir si ofrecer o no un crédito para casa a un sector de clientes, ejecuta un modelo que dará a cada cliente una calificación entre 0 y 1, esto es un estimado de la probabilidad de que cada cliente responderá positivamente a la propuesta. Este enfoque tiene la ventaja, de que los registros individuales pueden ser ordenados por probabilidad de aceptación o rechazo ante la propuesta bancaria.

La tarea de la clasificación ahora comienza a establecer el punto de confianza. Cualquiera de los clientes con una marca mayor o igual al punto de confianza marcado recibirá la oferta. Otros ejemplos son:

- Estimar el número de niños en una familia.
- Estimación del total de huéspedes en una casa.

Frecuentemente la clasificación y la estimación se utilizan juntas, como cuando la minería de datos es usada para predecir quienes realizarán una transferencia y entonces se estima el volumen de la transferencia.

Predicción

Cualquier predicción puede ser directamente una clasificación o estimación. La diferencia es el énfasis.

Cuando la minería de datos es usada para clasificar el uso primario de una línea telefónica casera (para llamadas normales o acceso a Internet) o detectar transacciones fraudulentas en una tarjeta de crédito, no se espera regresar para verificar si la clasificación fue correcta.

La clasificación puede ser o no correcta, pero la incertidumbre se elimina solo para completar el conocimiento: en el mundo real, las acciones relevantes han sido posicionadas en un lugar. El teléfono es o no usado para conexiones a Internet. Las transacciones de la tarjeta de crédito son o no fraudulentas.

Las tareas de predicción se sienten diferentes porque los registros son clasificados según algún comportamiento futuro predecido o un valor estimado. Con la predicción, la

única forma de checar la predicción de la clasificación es esperar y ver. Ejemplos de las tareas de la predicción son:

- Predecir el volumen del saldo que será transferido de una tarjeta de crédito propuesta.
- Predicción de cual de los clientes se quedará con la empresa los próximos 6 meses.
- Predicción de cual de los clientes ordenará un servicio de valor agregado, como una conferencia tripartita o correo de voz.

Cualquiera de las técnicas usadas como clasificación o estimación puede ser adaptado para usar ejemplos de entrenamiento donde el valor de la variable a predecir es conocido, junto con datos históricos del ejemplo.

Los datos históricos son usados para construir un modelo que explique el comportamiento observado. Cuando este modelo es aplicado a las entradas actuales, los resultados son predicción de un comportamiento futuro.

Agrupación por afinidad o reglas de asociación

La tarea de la agrupación por afinidad es poder determinar cuales cosas van juntas. El ejemplo prototipo es determinar que cosas van juntas en un carro de compras de un cliente en un supermercado. Las ventas en "cadena" pueden usar agrupación por afinidad para planear un arreglo de elementos en venta o ser puestos en un catalogo juntos para ser vistos juntos por el cliente para su adquisición. La agrupación por afinidad puede entonces ser usada para identificar las oportunidades de ventas cruzadas y poder diseñar paquetes atractivos o grupos de productos y servicios.

Clustering

Clustering es la tarea de segmentar un grupo diverso en un número de grupos similares o clusters. Lo que distingue al clustering de la clasificación, es que el clustering no depende de clases predefinidas.

En el clustering, no hay clases predefinidas y no hay ejemplos. Los registros son agrupados juntos basándose en su similitud. Esto es realizado por la minería de datos para determinar un significado y así se obtienen los clusters resultantes. Un cluster en particular de síntomas que presenta una persona enferma, podría indicar que tipo de enfermedad es la que contrajo. Similarmente, los clusters de clientes que compran videos y discos de música, indican ser miembros de diferentes grupos culturales.

El clustering se utiliza como el primer paso de alguna otra forma de minería de datos o modelado. Por ejemplo, el clustering puede ser el primer paso en una tarea de segmentación de mercado. Intentando crear una serie de reglas para responder a las preguntas "¿qué clientes responderán mejor a un tipo de promoción?," primero se deberá segmentar a los clientes basándose en clusters de gente con hábitos de compra similares, y entonces la siguiente técnica determinará que tipo de promoción trabajará en cada cluster.

Descripción y visualización

Algunas veces el propósito de la minería de datos es simplemente describir que es lo que pasará en una complicada base de datos para incrementar el entendimiento de la gente, productos o procesos que crean los datos.

Una buena descripción del comportamiento puede ofrecer una buena explicación. Una buena descripción sugiere por donde empezar a buscar una explicación.

La visualización de los datos es una poderosa forma de describir la minería de datos. Pero no siempre es fácil construir visualizaciones significativas, pero las correctas pueden trabajar con cientos de reglas de asociación.

METODOLOGÍA DE LA MINERÍA DE DATOS

El proceso de la minería de datos se divide en cuatro etapas:

1. Identificación del problema.
2. Análisis de los datos.
3. Tomar una acción.
4. Medición de los resultados.

La primera y tercera etapas se construyen principalmente en los principios del negocio. Para que la minería de datos sea exitosa, estos resultados del negocio deben ser direccionados apropiadamente. La empresa debe identificar fuentes de donde extraer mejor información para obtener un mejor conocimiento sobre que acción tomar.

La identificación del problema, es responsabilidad acerca de las expectativas que se tienen por parte de la empresa que esta solicitando la minería de datos, es un factor principal para poder determinar que rumbo deberá tomar la minería de datos. Esto es, se desea solo conocer el mercado, se desea segmentar a los clientes o se desea construir una serie de estrategias de mercadotecnia.

En el análisis de datos, es donde entra la minería de datos y donde la información resultante es revisada para producir conocimiento.

Existen dos formas básicas de realizar una minería de datos.

La primera es probando hipótesis, es un comportamiento de arriba a abajo, que aprueba o no ideas ya concebidas.

La segunda es obtener conocimiento descubierto, es un comportamiento de abajo a arriba que inicia con los datos e intenta decir algo que no se conoce acerca del negocio.

En la metodología de arriba abajo, es posible obtener explicaciones a través de observar el comportamiento y analizar los datos. En la otra metodología, los datos sugieren nuevas hipótesis a ser probadas.

Prueba de hipótesis

La prueba de una hipótesis es una posible explicación propuesta que puede ser probada. Probándola, la validez es mostrada al analizar datos que pueden simplemente ser recolectados por observación o generados directamente por un experimento. Para poder hacer esta prueba, se deben hacer ciertos pasos de ese proceso. El método se compone por los siguientes pasos:

a) Generación de buenas ideas (hipótesis).
b) Determinar que datos serán usados para que la hipótesis sea probada.
c) Localizar los datos.
d) Preparar los datos para su análisis.
e) Construir modelos computacionales sobre los datos.
f) Evaluar los modelos computacionales para confirmar o rechazar la hipótesis.

a) Generación de buenas ideas (hipótesis). La clave para este paso se inicia con un flujo de ideas que estén enfocadas a la resolución del problema, intentando conseguir la mejor salida del problema.

b) Determinar que datos serán usados para probar la hipótesis. Una vez que se ha seleccionado la hipótesis se debe evaluar con detalle. Algunas hipótesis pueden ser probadas por simples consultas a bases de datos existentes. Otras requieren información de sistemas operacionales o desde fuentes externas de datos. Para cada hipótesis se debe realizar una lista de requerimientos.

c) Obtener los datos. La minería de datos requiere datos, que supuestamente, ya existen en el data warehouse y están estructurados. Pero la realidad no es esa, ya que se cuenta con multiplataformas, hay incompatibilidad y no es lo que se requiere. Por lo que se debe hacer que la información este en un formato accesible, en computadoras que estén disponibles y listas para su uso. En las cuales debe residir toda la información del negocio.

d) Preparar los datos para su análisis. Es muy común que cuando los datos ya han sido recolectados y almacenados no estén en formato o condiciones para llevar a cabo la minería de datos. La transformación de los datos dependerá directamente de la herramienta que se utilice para llevar a cabo la minería de datos, los casos más comunes de estas transformaciones son:

- Sumarización.
- Arquitecturas de computadoras incompatibles.
- Codificación inconsistente de los datos.
- Datos textuales.
- Valores perdidos.

Sumarización. ¿Cuál es el correcto nivel de detalle? La respuesta depende del análisis que será realizado. Como regla general, mientas existan más datos, se requerirán mayor potencial de cómputo, mientras que si se aplica una sumarización (condensación de los datos) se obtienen resultados menos detallados pero con

menor cantidad de recursos empleados para su análisis. Unas de las razones para sumarizar los datos son:

- Los datos pueden reflejar detalles que simplemente no son requeridos, además de que no es posible tratar todos los datos por su gran volumen. Esto sucede mucho en la técnica de análisis de canasta de mercado.
- Cuando los datos no son suficientes para llegar a un nivel de detalle, entonces, se hacen mezclas de los mismos datos para poder obtener grupos con ciertas características requeridas para el problema en particular.

Arquitecturas de computadoras incompatibles. Existen muchos formatos en los que los datos son almacenados por distintas herramientas, por lo que no siempre es posible que sean leídos los datos a partir de una única herramienta cuando estos no fueron preprocesados para tener un tipo estándar. Es por eso que existen muchas utilerías en el mercado que se dedican a realizar estas transformaciones.

Codificación inconsistente de los datos. Cuando la información del mismo tipo es recolectada desde varias fuentes, éstas representan los mismos datos de diferentes formas. Si estas diferencias no son detectadas y corregidas, se agregan errores que se reflejan en resultados incorrectos.

Datos textuales. Los datos textuales representan muchos problemas, ya que no siempre son capturados con un formato "riguroso" y se pueden tener cualquier cantidad de cadenas con la misma información representada de distintas formas, por lo que una cadena "si ", puede ser diferente a " sí", o " si " o "sí", claro esta que esto dependerá de la herramienta que se utilice para la minería de datos. Pero en general si no se tiene una plantilla para recibir esa información en un campo de texto, no siempre será información fácil de recabar. Además, los datos que se requiere para hacer ciertos análisis detallados, no se puede dejar que el usuario los codifique libremente.

Valores perdidos. En la práctica diaria, hay muchas perdidas de valores cuando se recaba información. Cuando se esta trabajando una minería de datos, se desea obtener resultados siempre, aún cuando se tengan valores nulos o inexistentes. En algunos casos, algunos campos sin valor puede que se completen con ciertas condiciones, como el salario, se puede basar en algún otro dato, o simplemente colocar uno promedio, pero el teléfono, no es posible colocar otro que no sea el correcto; para esto se pueden utilizar las redes neuronales.

e) Construcción de modelos computacionales. Antes de poder arrancar con un modelo que se lleve a cabo e implique gastos para el negocio, es necesario poder probar esa idea, de tal forma en la que se corran los menores riesgos para la empresa, siendo la opción optima para esto el modelado computacional. Pero antes de este paso, se requiere la concepción de la idea, ajustada a las necesidades empresariales.

f) Evaluación de modelos computacionales. Finalmente, después de la construcción del modelo, se llevará a cabo la prueba sobre las hipótesis; dependiendo de la naturaleza de las mismas y del modelo que las respalde, será si regresa como salida un solo valor o una colección de reglas de asociación generadas por la técnica de análisis de canasta de mercado o determinar una correlación por un modelo de regresión.

Conocimiento descubierto

El conocimiento descubierto es el resultado de la minería de datos. El poder obtener un conocimiento a partir de la información contenida en una base de datos, es muy importante para poder conocer aquellos patrones que aun no se descubrían.

Se puede dividir este tipo de conocimiento descubierto como dirigido o no dirigido.

El conocimiento descubierto dirigido es tratar de explicar un valor objetivo de un dato a partir de todos los demás existentes. Esto se hace seleccionando un valor específico y se entenderá como se estima, clasifica o predice ese valor.

En el conocimiento descubierto no dirigido no existe ese valor objetivo. Tan solo se obtiene como respuesta patrones que pueden ser significativos. En este caso no se tiene una necesidad específica que requiera respuesta por parte de la minería de datos. El conocimiento descubierto no dirigido reconoce relaciones en los datos.

El conocimiento descubierto dirigido explica esas relaciones donde fueron encontradas. El no dirigido puede ser utilizado en clustering y agrupación por afinidad, pero la mayor parte del conocimiento descubierto se utiliza para obtener respuestas específicas (en la forma directa), para saber "algo interesante" del negocio.

El conocimiento descubierto dirigido. Está orientado a una meta. Es una predicción específica de un dato o clases a partir de relaciones encontradas, declaradas y exploradas. Es un proceso de buscar patrones interesantes en los datos que expliquen eventos pasados para ayudar a predecir eventos futuros. Los pasos en el proceso para descubrir conocimiento dirigido son:

1. Identificar las fuentes de datos preclasificados.
2. Preparar datos para el análisis.
3. Construir y probar un modelo computacional.
4. Evaluar el modelo computacional.

Identificar las fuentes de datos preclasificados. El conocimiento descubierto se basa en la premisa de que las respuestas de esas preguntas serán encontradas por un tratamiento directo en los datos del pasado. Entonces, el primer requerimiento para un conocimiento descubierto exitoso son buenos datos.
La fuente ideal de datos es un data warehouse empresarial. Esto asegura consistencia, ausencia de duplicidad y compatibilidad en los datos, ya que estos han sido previamente verificados y se ha eliminado cualquier anomalía, otorgando así, el nivel correcto de agregación.
De no existir el data warehouse, las bases de datos de uso actual están optimizadas para cumplir con su tarea inicial, procesar transacciones de forma rápida y efectiva, pero sin tener las opciones de sumarización ni poder conservar datos históricos a grandes volúmenes.
Es necesario poder preclasificar los datos, ya que se utilizan datos pasados para modelar el futuro, pudiendo construir modelos computacionales de cada segmento detectado en la preclasificación de los datos.

Preparar los datos para el análisis. Esto es poder tener las fuentes externas de los distintos datos en el mismo formato, ya que de no ser así no se logrará la integración de toda la información.
Algunas herramientas de minería de datos no son capaces de hacer ciertas consideraciones sobre algunos datos inexistentes o con un formato no estándar. Por lo que es necesario estandarizar todos los campos existentes de cada uno de los registros.
Los datos además, deberán ser divididos para construir un modelo inicial (datos de entrenamiento), probados para ajustar el modelo inicial para hacerlo más general y más adecuado (datos de prueba), y datos evaluados para verificar la efectividad de los modelos construidos (datos de evaluación). La cantidad de datos que se requieren dependen del algoritmo a utilizar, la complejidad de los datos y la frecuencia de obtención de salidas.

Construir y probar un modelo computacional. Los detalles de este paso varían de técnica a técnica, pero en términos generales se debe conseguir que la variable objetivo sea expresada en términos de las variables de entrada, mostrando una relación entre el dato que se quiere estimar, clasificar o predecir y los otros datos contenidos en la base de datos.
Existe un problema que es causado por un sobre ajuste (overfitting), por lo que se llegan a conclusiones erróneas de patrones, esto también será determinado directamente por la técnica utilizada. Se recomienda tener datos disponibles para un entrenamiento que no llegue a afectar a la organización.
El overfitting es un sobre ajuste, esto es, esta apunto de "memorizar" el modelo los datos, por lo que será muy difícil para los registros poder ser clasificados con este

modelo. Una comparación sería con el juego de baloncesto, en el que el aro de la canasta es tan justo (cerrado) que solo un tipo de pelota y en una cierta posición podrá atravesarlo.
El underfitting es un bajo ajusto, esto es, en el juego de baloncesto se representaría por tener un aro de la canasta tan grande, que desde cualquier ángulo y cualquier tipo de pelota sería posible atravesarlo.

Evaluar el modelo computacional. Aunque no se sabe aun el comportamiento del modelo computacional en datos que no sean de prueba, se espera que con la eliminación de reglas o relaciones que dependían enteramente del conjunto de prueba, sea posible optimizar el desempeño del modelo en estos datos.

El conocimiento descubierto no dirigido. En este proceso se espera que las herramientas de minería de datos encuentren patrones o estructuras significativas dentro de los datos. Este es usado comúnmente en el análisis de canasta de mercado para responder la pregunta "¿qué productos se venden juntos?", También en el clustering para agrupar aquellos registros que tengan algo en común. Es usual que este proceso sea el preludio para futuras investigaciones con técnicas más directas. Los pasos en el proceso de conocimiento descubierto no dirigido son:

1. Identificar las fuentes de datos.
2. Preparación de los datos para análisis.
3. Construcción y entrenamiento del modelo computacional.
4. Evaluar el modelo computacional.
5. Aplicar el modelo computacional a nuevos datos.
6. Identificar potenciales datos objetivo para aplicar descubrimiento de conocimiento dirigido.
7. Generar nuevas hipótesis para probarlas.

Los pasos del 1 al 5 son exactamente iguales que el conocimiento descubierto dirigido, por lo que los dos nuevos son los que diferencian a este método del anterior.

Identificar potenciales datos objetivo para conocimiento descubierto dirigido. Es realmente bueno este método para poder reconocer ciertos patrones que servirán como futuras "preguntas" a ser resueltas por otros métodos, como el análisis de canasta de mercado (ya que este es especialista para resolver las preguntas porque, quien y cuando sobre las ventas de aquellos productos que tienen alguna relación en su venta) o el conocimiento descubierto dirigido.

Generar nuevas hipótesis para probarlas. Una vez que se ha logrado segmentar la población de los datos, se deberá hacer un análisis de los nuevos segmentos, ya

que aun no se ha completado el ciclo para poder tener respuestas concisas sobre algún comportamiento del negocio.

En resumen, se han mencionado tres metodologías de la minería de datos para poder realizar el estudio sobre los datos.

1. **La prueba de hipótesis,** que se resume en 8 pasos, los cuales son:
   - Generación de las hipótesis (ideas que intentan resolver la problemática actual).
   - Definir los datos necesarios para probar la hipótesis (esto es, saber conque tipo de información es que la hipótesis trabajará para regresar algún valor).
   - Localizar los datos (identificar las fuentes de datos para poder extraerlos a una base de datos de prueba y así trabajar con una copia de los mismos).
   - Preparación de los datos para el análisis (existen muchas formas de tratar los datos de forma tal que en el procesamiento no exista duplicidad de información, redundancia, inconsistencia o incompatibilidad en el formato).
   - Diseño de modelos computacionales y consultas a bases de datos para probar la hipótesis con los datos.
   - Evaluación de los resultados de las consultas y los modelos computacionales.
   - Evaluación de las acciones tomadas (esto es, medir los resultados finales de las acciones llevadas a cabo a partir de las hipótesis evaluadas y llevadas a la práctica).
   - Reiniciar el proceso de la minería de datos tomando ventaja de los nuevos datos generados a partir de las acciones tomadas (siendo esto un paso adelante, ya que los nuevos datos se acercan aún más al resultado final que se persigue por estar enfocados a la resolución del problema.

2. **El conocimiento descubierto dirigido,** el cual tiene como pasos a seguir para obtener un resultado, los siguientes:
   - Identificar las fuentes de datos preclasificados.
   - Preparación de los datos para su análisis.
   - Selección apropiada de una técnica de conocimiento descubierto basado en las características de los datos y de las metas perseguidas.
   - Dividir los datos en tres partes, para entrenamiento del modelo, para prueba del modelo y para evaluación del modelo.
   - Uso del conjunto de datos de entrenamiento para construir el modelo computacional.

- Tomar el modelo para aplicarlo al conjunto de datos de prueba, con el modelo ahora ajustado se aplica en el conjunto de datos de evaluación.
- Tomar el modelo sin errores para aplicárselo al conjunto de datos de evaluación.
- Tomar una acción basada en los resultados de la minería de datos.
- Medir los efectos de las acciones tomadas.
- Reiniciar el proceso de la minería de datos tomando ventaja de los nuevos datos generados en las acciones anteriores.

3. **El conocimiento descubierto no dirigido,** el cual es utilizado como preludio de otras técnicas de la minería de datos, teniendo las siguientes actividades:

- Identificar las fuentes de datos disponibles.
- Preparar los datos para el análisis.
- Seleccionar una técnica apropiada para conocimiento descubierto no dirigido, basada en las características de los datos y de las metas perseguidas.
- Usar la técnica seleccionada para descubrir estructuras ocultas en los datos.
- Identificar los posibles objetivos para aplicar el conocimiento descubierto dirigido y generar nuevas hipótesis para ser probadas.

Medición de la efectividad de la minería de datos

¿Pero como saber si es que ya se logró la meta buscada con la minería de datos? Para saber esto es necesario tener bien definido el problema a resolver. Dada la naturaleza de la minería de datos, los resultados que se arrojan se pueden englobar en tres grandes ramas:

- Ganar mayor conocimiento sobre el comportamiento del ente analizado.
- Descubrir patrones importantes de los datos.
- Aprender algo interesante del negocio.

Se puede decir que las metas que persigue la minería de datos son descriptivas o predictivas. En una meta descriptiva se obtiene entendimiento, explicación o conocimiento descubierto, lo cual es rápidamente visualizado como resultado. En una meta predictiva, se debe tener un muy buen modelo (suficientemente descriptivo) para poder justificar la respuesta.

Obviamente la medición de la precisión de un modelo predictivo o de clasificación es con respecto al Número de registros que han sido ordenados erróneamente. En el caso de la medición de un modelo descriptivo, se hace por medio de la *longitud mínima de descripción, MDL*, que es el número de veces que requiere el modelo para decodificar una regla y las excepciones a esa regla. Mientras menor sea ese número, mejor es la regla.

Para el caso de la medición de un modelo de estimación, se realiza por medio de la desviación estándar aplicada a las diferencias de los valores estimados y los valores reales.

$$\text{Desviación estándar } \sigma = \sqrt{\frac{\sum (valor\_real - valor\_estimado)^2}{numero\_total\_valores}}$$

La importancia de la desviación estándar es que los resultados que entrega son en las mismas unidades de los valores que se están trabajando, teniendo esto como resultado una mejor comprensión de los resultados arrojados. Es una medida de dispersión muy utilizada, y es la raíz cuadrada de la varianza.

Básicamente se puede separar dos grandes tendencias de la minería de datos, la minería de datos *dirigida (arriba-abajo)*, usada cuando se sabe que es lo que se está buscando.

Y la minería de datos no dirigida *(abajo-arriba)*, usada para buscar patrones en los datos dejando que el usuario determine si le interesan o no. Se dice que los datos hablan por sí mismos.

Los modelos construidos por la minería de datos considerados como *predictivos*, pueden resolver algunas cuestiones como las siguientes, ya que asignan calificaciones y niveles de confianza para alguna salida relevante:

- ¿Quién puede responder a una oferta dada, basada en la historia de campañas de mercadotecnia anteriores?
- ¿Cuál es el tratamiento médico adecuado, basado en la experiencia?
- ¿Cuál máquina es más propensa a que falle a continuación?
- ¿Cuál cliente está próximo a cambiarse con la competencia en los próximos seis meses?
- ¿Cuál de las transacciones registradas es un fraude, según las experiencias pasadas?

La meta en las predicciones es aprender del pasado y aprender de tal forma en la que el conocimiento pueda ser aplicado en el futuro. El mejor modelo no es el cual obtenga la mejor calificación cuando se está construyendo, sino el que su desempeño con datos nunca vistos sea el mejor.

Los resultados de la minería de datos no dirigida, se implanta para una exploración de los datos. Se resuelven las preguntas como:

- ¿Qué hay en los datos?
- ¿Cómo se ven los datos?
- ¿Existen patrones inusuales en los datos?
- ¿Qué segmentación de clientes sugieren los datos?

Capítulo II

DATA WAREHOUSE

Desde que se inició la era de la computadora, las organizaciones han usado los datos desde sus sistemas operacionales para atender sus necesidades de información. Algunas proporcionan acceso directo a la información contenida dentro de las aplicaciones operacionales. Otras, han extraído los datos desde sus bases de datos operacionales para combinarlos de varias formas no estructuradas, en su intento por atender a los usuarios en sus necesidades de información.

La gestión administrativa reconoce que una forma de elevar su eficiencia está en hacer el mejor uso de los recursos de información que ya existen dentro de la organización. Sin embargo, a pesar de que esto se viene intentando desde hace muchos años, no se tiene todavía un uso efectivo de los mismos.

La mayoría de las organizaciones hacen lo posible por conseguir buena información, pero el logro de ese objetivo depende fundamentalmente de su arquitectura actual, tanto de hardware como de software.

El **data warehouse**, provee un ambiente para que las organizaciones hagan un mejor uso de la información que está siendo utilizada por diversas aplicaciones operacionales.

Un data warehouse es una colección de datos en la cual se encuentra integrada la información de la Institución y que se usa como soporte para el proceso de toma de decisiones gerenciales.

Reunir los elementos de datos apropiados desde diversas fuentes de aplicación en un ambiente integral centralizado, simplifica el problema de acceso a la información y en consecuencia, acelera el proceso de análisis, consultas y el menor tiempo de uso de la información.

Las aplicaciones para soporte de decisiones basadas en un data warehouse, pueden hacer más práctica y fácil la explotación de datos para una mayor eficacia del negocio, que no se logra cuando se usan sólo los datos que provienen de las aplicaciones operacionales.

Un data warehouse se crea al extraer datos desde una o más bases de datos de aplicaciones operacionales. La base de datos extraída es transformada para eliminar inconsistencias y resumir si es necesario y luego, cargada en el data warehouse. El proceso de transformar, crear el detalle de tiempo variante, resumir y combinar los extractos de datos, ayudan a crear el ambiente para el acceso a la información Institucional. Este nuevo enfoque ayuda a las personas individuales, en todos los niveles de la empresa, a efectuar su toma de decisiones con más responsabilidad.

La innovación de la Tecnología de Información dentro de un ambiente data warehousing, puede permitir a cualquier organización hacer un uso mejor de los datos, como un ingrediente clave para un proceso de toma de decisiones más efectivo.

Las organizaciones tienen que aprovechar sus recursos de información para crear la información de la operación del negocio, pero deben considerarse las estrategias tecnológicas necesarias para la implementación de una arquitectura completa del data warehouse.

En sentido general, el data warehouse es utilizado en la minería de datos como sostén de toda la información y datos, ya que proporciona un ambiente operacional óptimo al contener datos históricos, resumidos y detallados, combinados con alta disponibilidad, integridad y limpieza de los datos ahí contenidos.

INTRODUCCIÓN AL CONCEPTO DE DATA WAREHOUSING

Data Warehousing es un aspecto importante de la arquitectura para los sistemas de información. Soporta el procesamiento informático al proveer una plataforma sólida, a partir de los datos históricos para hacer el análisis. Facilita la integración de sistemas de aplicación no integrados. Organiza y almacena los datos que se necesitan para el procesamiento analítico e informático sobre una amplia perspectiva de tiempo.

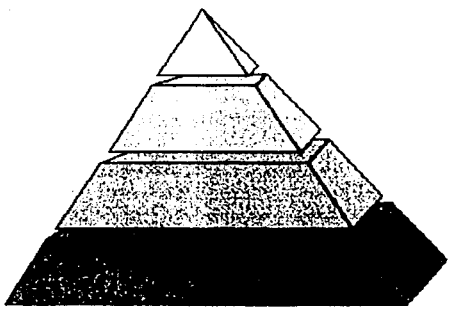
Un Data Warehouse o Depósito de Datos es una colección de datos orientado a temas, integrado, no volátil, de tiempo variante, que se usa para el soporte del proceso de toma de decisiones gerenciales.

Se puede caracterizar un data warehouse haciendo un contraste de cómo los datos de un negocio almacenados en un data warehouse, difieren de los datos operacionales usados por las aplicaciones de producción.

Base de Datos Operacional	Data Warehouse
Datos Operacionales	Datos del negocio para Información
Orientado a la aplicación	Orientado al tema
Sólo datos actuales	Datos Actuales y históricos
Sólo datos detallados	Datos Detallados y resumidos
Cambia continuamente	Estable

Diferentes tipos de información

El ingreso de datos en el data warehouse viene desde el ambiente operacional en casi todos los casos. El data warehouse es siempre un almacén de datos transformados y separados físicamente de la aplicación donde se encontraron los datos en el ambiente operacional.



Capas en las que se encuentra dividido un data warehouse

SISTEMAS DE INFORMACIÓN

Los sistemas de información se han dividido de acuerdo al siguiente esquema:

- **Sistemas Estratégicos (soporte de decisión)**, orientados a soportar la toma de decisiones, facilitan la labor de la dirección, son sistemas sin carga periódica de trabajo, es decir, su utilización no es predecible.

  Destacan entre estos sistemas: los Sistemas de Información Gerencial (MIS), Sistemas de Información Ejecutivos (EIS), Sistemas de Información Georeferencial (GIS), Sistemas de Simulación de Negocios (BIS y que en la práctica son sistemas expertos o de Inteligencia Artificial-AI).

- **Sistemas Tácticos**, diseñados para soportar las actividades de coordinación de actividades y manejo de documentación, facilitan la gestión de la información por parte de los niveles intermedios de la organización.

- **Sistemas Técnico-Operacionales,** se usan periódicamente y cubren el núcleo de operaciones cotidianas tradicionales de captura masiva de datos (Data Entry) y servicios básicos de tratamiento de datos, con tareas predefinidas (contabilidad, facturación, almacén, presupuesto, personal y otros sistemas administrativos). Estos sistemas están evolucionando con la irrupción de sensores, autómatas, sistemas multimedia, bases de datos relacionales más avanzadas.

- **Sistemas Interinstitucionales**, es consecuencia del desarrollo organizacional orientado a un mercado de carácter global, el cual obliga a pensar e implementar estructuras de comunicación más estrechas entre la organización y el mercado, todo esto a partir de la generalización de las redes informáticas de alcance nacional y global (INTERNET).

  La tecnología data warehouse basa sus conceptos y diferencias *entre* dos tipos fundamentales de sistemas de información en todas las organizaciones: *los sistemas técnico-operacionales y los sistemas de soporte de decisiones*. Este último es la base de un data warehouse.

- **Sistemas de Soporte de Decisiones**, hay otras funciones dentro de la empresa que tienen que ver con el planeamiento, previsión y administración de la organización. Estas funciones son también críticas para la supervivencia de la organización, especialmente en nuestro mundo de rápidos cambios.

  Las funciones como "planificación de marketing", "planeamiento de ingeniería" y "análisis financiero", requieren además, de sistemas de información que los soporte. Pero estas funciones son diferentes de las operacionales, así como los tipos de sistemas y la información requerida también son diferentes.

Mientras las necesidades de los datos operacionales se enfocan normalmente hacia una sola área, los datos para el soporte de decisiones, con frecuencia, toma un número de áreas diferentes y necesita grandes cantidades de datos operacionales relacionadas.
Son estos sistemas sobre los se basa la tecnología data warehousing.

CARACTERÍSTICAS DE UN DATA WAREHOUSE

Entre las principales se tiene:

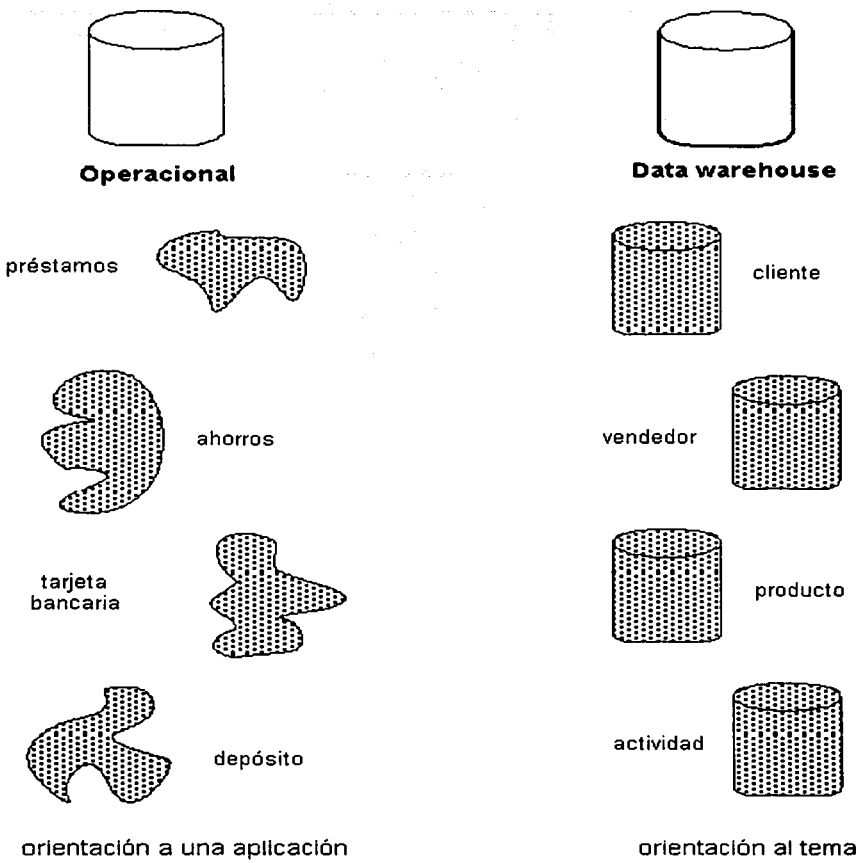
1. Orientado al tema
2. Integrado
3. De tiempo variante
4. No volátil

1. Orientado a Temas

En el data warehouse la información se clasifica en base a los aspectos que son de interés para la empresa.

El ambiente operacional se diseña alrededor de las aplicaciones. Por ejemplo, una aplicación de ingreso de órdenes puede acceder a los datos sobre clientes, productos y cuentas. La base de datos combina estos elementos en una estructura que acomoda las necesidades de la aplicación.

En el ambiente data warehousing se organiza alrededor de temas, tales como cliente, vendedor, producto y actividad. Por ejemplo, para un fabricante, éstos pueden ser clientes, productos, proveedores y vendedores. Para una universidad pueden ser estudiantes, clases y profesores. Para un hospital pueden ser pacientes, personal médico, medicamentos, etc.



El data warehouse tiene una fuerte orientación al tema

En el data warehouse se excluyen los datos que no serán usados por el proceso de sistemas de soporte de decisiones. Los datos de las orientadas a las aplicaciones, satisfacen los requerimientos funcionales y de proceso, y pueden ser usados o no por el analista de soporte de decisiones.

2. Integración.

El aspecto más importante del ambiente data warehousing es que los datos encontrados en su interior estén siempre integrados.

La integración de datos se muestra de muchas formas: en convenciones de nombres consistentes, en la medida uniforme de variables, en la codificación de estructuras consistentes, en atributos físicos de los datos consistentes, fuentes múltiples, entre otros.

El contraste de la integración encontrada en el data warehouse con la carencia de integración del ambiente de aplicaciones, se muestran en la figura, titulada Integración (página 40).

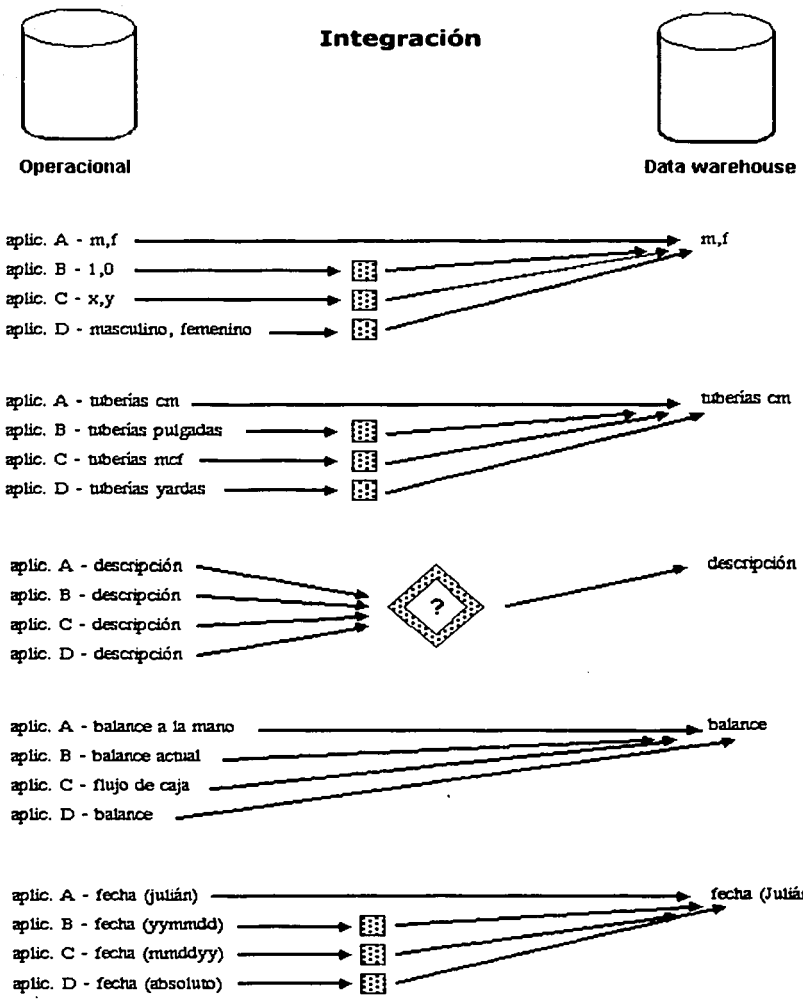
A través de los años, los diseñadores de las diferentes aplicaciones han tomado sus propias decisiones sobre cómo se debería construir una aplicación. Los estilos y diseños personalizados se muestran de diferentes formas.

Se diferencian en la codificación, en las estructuras claves, en sus características físicas, en las convenciones de nomenclatura y otros. La figura mencionada también muestra algunas de las diferencias más importantes en las formas en que se diseñan las aplicaciones.

a) ***Codificación***. Los diseñadores de aplicaciones codifican el campo GÉNERO en varias formas. Un diseñador representa GÉNERO como una "M" y una "F", otros como un "1" y un "0", otros como una "X" y una "Y" e inclusive, como "masculino" y "femenino".
   No importa mucho cómo el GENERO llega al data warehouse. Probablemente "M" y "F" sean tan buenas como cualquier otra representación. Lo importante es que sea de cualquier fuente de donde venga, el GENERO debe llegar al data warehouse en un estado integrado uniforme.
   Por lo tanto, cuando el GENERO se carga en el data warehouse desde una aplicación, donde ha sido representado en formato "M" y "F", los datos deben convertirse al formato del data warehouse.

b) ***Medida de atributos***. Los diseñadores de aplicaciones miden las unidades de longitud en una variedad de formas. Un diseñador almacena los datos de tuberías en centímetros, otros en pulgadas, otros en millones de pies cúbicos por segundo y otros en yardas.
   Al dar medidas a los atributos, la conversión se traduce en las diversas unidades de medida usadas en las diferentes bases de datos para transformarlas en una medida estándar común.
   Cualquiera que sea la fuente, cuando la información de la tubería llegue al data warehouse necesitará ser medida de la misma forma.

c) ***Convenciones de nomenclatura.*** El mismo elemento es frecuentemente referido por nombres diferentes en las diversas aplicaciones. El proceso de transformación asegura que se utilice.



Cuando los datos se mueven al data warehouse desde las aplicaciones orientadas al ambiente operacional, los datos se integran antes de entrar al depósito.

d) ***Múltiples Fuentes.-*** El mismo elemento puede derivarse desde múltiples fuentes. En este caso, el proceso de transformación debe asegurar que la fuente apropiada sea usada, documentada y movida al depósito.

Tal como se mostró en la figura, la integración afecta casi todos los aspectos de diseño - las características físicas de los datos, la disyuntiva de tener más de una de fuente de datos, el problema de estándares de denominación inconsistentes, formatos de fecha inconsistentes, entre otros. Cualquiera que sea la forma del diseño, el resultado es el mismo, la información necesita ser almacenada en el data warehouse en un modelo globalmente aceptable y singular, aun cuando los sistemas operacionales subyacentes almacenen los datos de manera diferente.

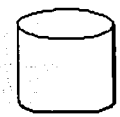
Cuando el analista de sistema de soporte de decisiones observe el data warehouse, su enfoque deberá estar en el uso de los datos que se encuentren en el depósito, antes que preguntarse sobre la confiabilidad o consistencia de los datos.

3. De Tiempo Variante

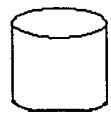
Toda la información del data warehouse es requerida en algún momento en el ambiente operacional. La información se requiere al momento de acceder. Cuando se accede a una unidad de información, se espera que los valores requeridos se obtengan en el momento de acceso.

La información en el data warehouse es solicitada en cualquier momento (es decir, no "ahora mismo"), los datos encontrados en el depósito se llaman de *tiempo variante*.

Los datos históricos son de poco uso en el procesamiento operacional. Los datos del data warehouse por el contrario, deben incluir los datos históricos para usarse en la identificación y evaluación de tendencias.



De tiempo variante



Operacional

Data warehouse

Valor actual de los datos:
- Horizonte de tiempo: 60-90 días
- La clave puede, como no, tener un elemento de tiempo
- Los datos pueden ser actualizados

Datos instantáneos:
- Horizonte de tiempo:5-10 años
- La clave contiene un elemento de tiempo
- Una vez que el snapshot se realice, el registro no puede ser actualizado

El tiempo variante se muestra de varias maneras:

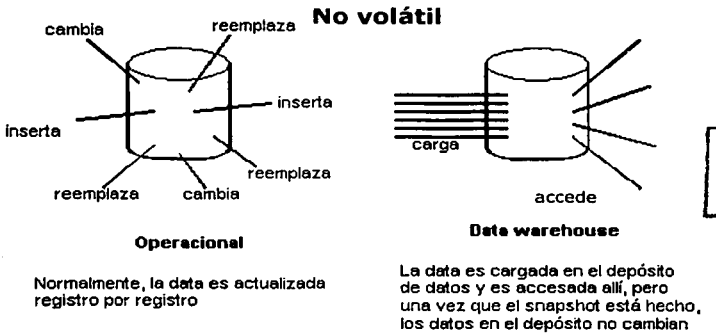
a) La información representa los datos sobre un horizonte largo de tiempo - desde cinco a diez años. El horizonte de tiempo representado para el ambiente operacional es mucho más corto - desde valores actuales hasta sesenta a noventa días.

b) Cada estructura clave en el data warehouse contiene, implícita o explícitamente, un elemento de tiempo como día, semana, mes, etc. que está casi siempre junto con la clave concatenada, encontrada en el data warehouse.

c) La información del data warehouse, una vez registrada correctamente, no puede ser actualizada. La información del data warehouse es, para todos los propósitos prácticos, una serie larga de "snapshots" (vistas instantáneas).
Por supuesto, si los snapshots de los datos se han tomado incorrectamente, entonces pueden ser cambiados. Asumiendo que los snapshots se han tomado adecuadamente, ellos no son alterados una vez almacenados. En algunos casos puede ser no ético, e incluso ilegal, alterar los snapshots en el data warehouse. Los datos operacionales, siendo requeridos a partir del momento de acceso, pueden actualizarse de acuerdo a la necesidad.

4. No Volátil

La información es útil sólo cuando es estable. Los datos operacionales cambian de momento a momento. La perspectiva más grande, esencial para el análisis y la toma de decisiones, requiere una base de datos estable.



En la figura anterior se muestra que la actualización (insertar, borrar y modificar), se hace regularmente en el ambiente operacional sobre una base de registro por registro. Pero la manipulación básica de los datos que ocurre en el data warehouse es mucho más simple. Hay dos únicos tipos de operaciones: la carga inicial de datos y el acceso a los mismos. No hay actualización de datos (en el sentido general de actualización) en el depósito, como una parte normal de procesamiento. El data warehouse si se actualiza periódicamente.

Hay una diferencia básica, entre el procesamiento operacional y del data warehouse. Ser precavido para actualizar las anomalías no es un factor en el data

warehouse, ya que no se hace la actualización de datos. En el nivel físico de diseño, se pueden tomar libertades para optimizar el acceso a los datos, particularmente al usar la normalización y desnormalización física.

La tecnología permite realizar backup y recuperación, transacciones e integridad de los datos, detección y solución al "deadlock" que es más complejo. En el data warehouse no es necesario el procesamiento.

La fuente de casi toda la información del data warehouse es el ambiente operacional. A simple vista, se puede pensar que hay redundancia masiva de datos entre los dos ambientes.

Los datos se filtran cuando pasan desde el ambiente operacional al de depósito. Existen muchos datos que nunca salen del ambiente operacional. Sólo los datos que realmente se necesitan ingresarán al ambiente de data warehouse.

El horizonte de tiempo de los datos es muy diferente de un ambiente al otro. La información en el ambiente operacional es más reciente con respecto a la del data warehouse. Desde la perspectiva de los horizontes de tiempo únicos, hay poca superposición entre los ambientes operacional y de data warehouse.

El data warehouse contiene un resumen de la información que no se encuentra en el ambiente operacional.

Los datos experimentan una transformación fundamental cuando pasa al data warehouse. La mayor parte de los datos se alteran significativamente al ser seleccionados y movidos al data warehouse.

ESTRUCTURA DE LOS DATOS DEL DATA WAREHOUSE

Los data warehouses tienen una estructura distinta. Hay niveles diferentes de esquematización y detalle que delimitan el data warehouse. La estructura de un data warehouse se muestra en la figura.

En la figura, se muestran los diferentes tipos de datos contenidos en el data warehouse:

1. Detalle de datos actuales.
2. Detalle de datos antiguos.
3. Datos ligeramente resumidos.
4. Datos completamente resumidos.
5. Meta data.

1. ***Datos actuales detallados.-*** En gran parte, el interés más importante radica en el detalle de los datos actuales, debido a que:
    - Refleja las ocurrencias más recientes, las cuales son de gran interés.
    - Es voluminoso, ya que se almacena al más bajo nivel de granularidad.

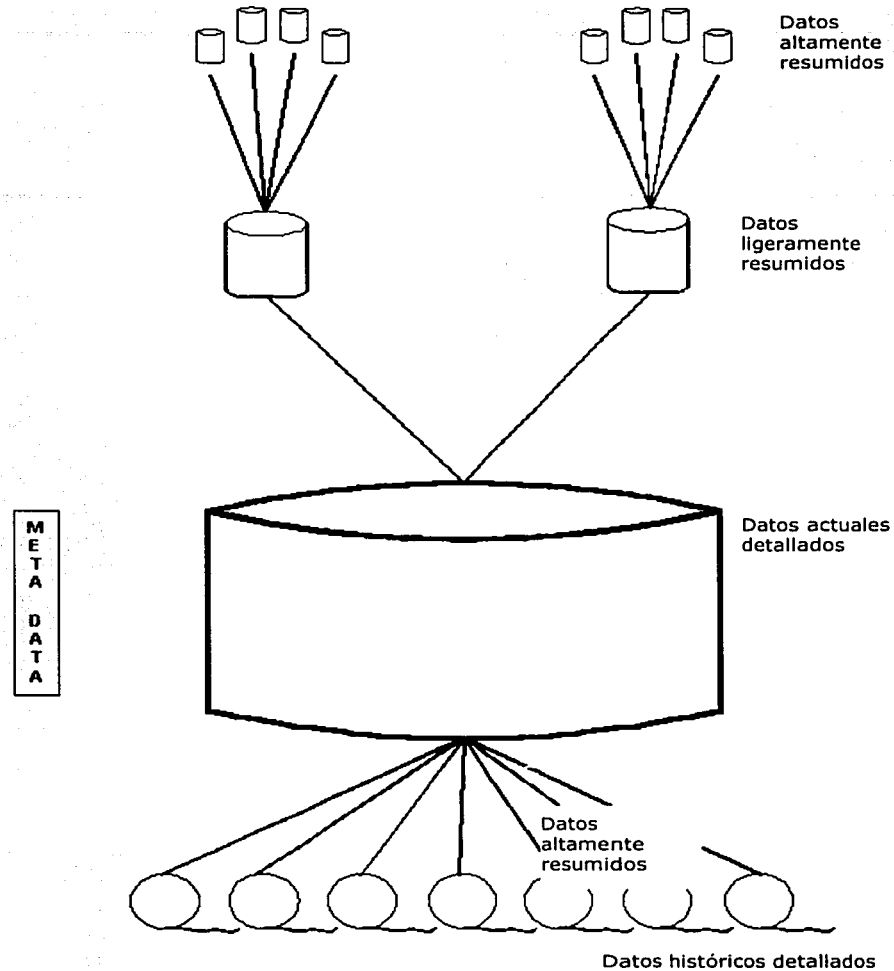
- Casi siempre se almacena en disco, el cual es de fácil acceso, aunque su administración sea costosa y compleja.

2. ***Datos antiguos detallados.-*** Los datos antiguos son aquellos que se almacenan sobre alguna forma de almacenamiento masivo. No son frecuentemente accedidos y se almacenan a un nivel de detalle, consistente con los datos detallados actuales. Mientras no sea prioritario el almacenamiento en un medio de almacenaje alterno, a causa del gran volumen de datos unido al acceso no frecuente de los mismos, es poco usual utilizar el disco duro como medio de almacenamiento.

3. ***Datos ligeramente resumidos.-*** Los datos ligeramente resumidos son aquellos que provienen desde un bajo nivel de detalle encontrado al nivel de detalle actual. Este nivel del data warehouse casi siempre se almacena en disco duro. Los puntos en los que se basa el diseñador para construirlo son:
   - Qué la unidad de tiempo, se encuentre sobre la esquematización hecha.
   - Qué atributos tendrán, los datos ligeramente resumidos.

4. ***Datos completamente resumidos.-*** El siguiente nivel de datos encontrado en el data warehouse es el de los datos completamente resumidos. Estos datos son compactos y difícilmente accesibles.
   A veces se encuentra en el ambiente de data warehouse y en otros, fuera del límite de la tecnología que ampara al data warehouse. (De todos modos, los datos completamente resumidos son parte del data warehouse sin considerar donde se alojan los datos físicamente.)

5. ***Metadata.*** La metadata son datos en una jerarquía diferente al de otros datos del data warehouse, debido a que su contenido no es tomado directamente desde el ambiente operacional.



Estructura de los datos en un Data Warehouse

La metadata juega un rol especial y muy importante en el data warehouse y es utilizada como:

a) Un directorio para ayudar al analista a ubicar los contenidos del data warehouse.

b) Una guía para el mapeo de datos de cómo se transforma, del ambiente operacional al de data warehouse.

c) Una guía de los algoritmos usados para la esquematización entre el detalle de datos actual, con los datos ligeramente resumidos y éstos, con los datos completamente resumidos, etc.

La metadata juega un papel mucho más importante en un ambiente data warehousing que en un operacional clásico.

Ejemplo:

Las de ventas pasadas, antes de 2003. Todas las de ventas desde que se inició la colección de los datos son almacenadas a nivel de detalle de datos históricos.

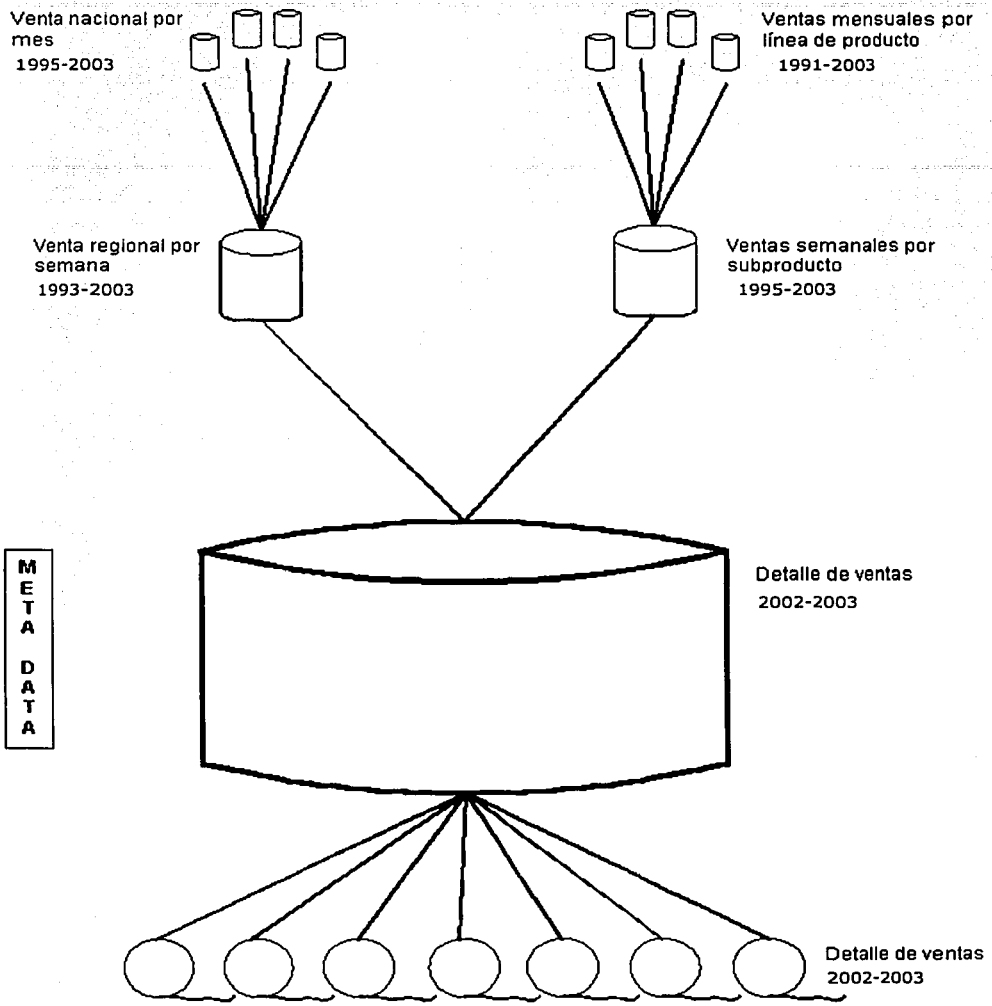
Los datos actuales detallados contienen información del 2003. En general, los datos detallados de ventas no se ubican en el nivel de datos detallados actuales del data warehouse hasta que haya pasado, cierto tiempo para que la información de ventas llegue a estar disponible para el data warehouse.

Habría un retraso de tiempo de por lo menos veinticuatro horas, entre el tiempo en que en el ambiente operacional se haya hecho un nuevo ingreso de la venta y el momento cuando la información de la venta haya ingresado al data warehouse. El detalle de las ventas se resume semanalmente por línea de subproducto y por región, para producir un almacenamiento de datos ligeramente resumidos.

La metadata contiene (al menos):

- La estructura de los datos.
- Los algoritmos usados para la esquematización.
- El mapeo desde el ambiente operacional al data warehouse.

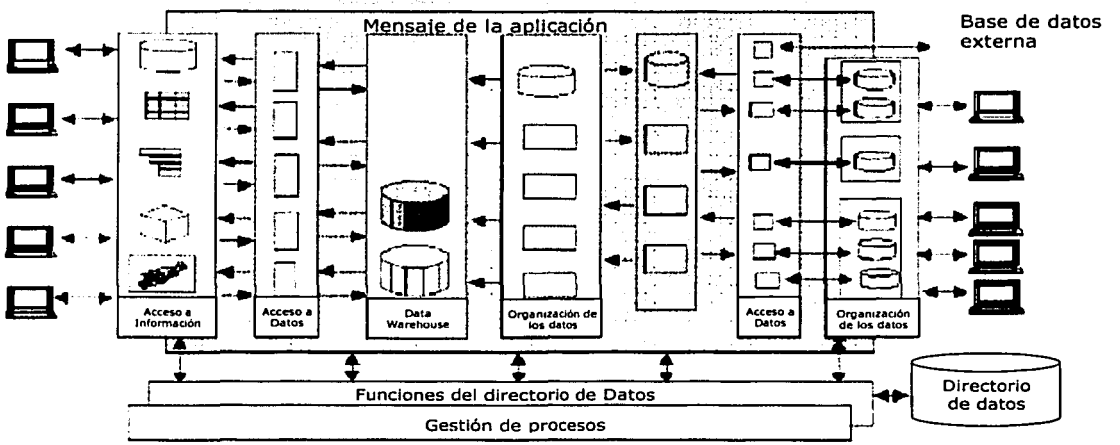
La información adicional que no se esquematiza es almacenada en el data warehouse. En muchas ocasiones, allí se hará el análisis y se producirá un tipo de resumen. El único tipo de esquematización que se almacena permanentemente en el data warehouse, es el de los datos que son usados frecuentemente.



Niveles de esquematización en un data warehouse

ARQUITECTURA DE UN DATA WAREHOUSE

Una de las razones por las que el desarrollo de un data warehouse crece rápidamente, es que realmente es una tecnología muy entendible. De hecho, el data warehousing puede representar mejor la estructura amplia de una empresa para administrar los datos dentro de la organización. A fin de comprender cómo se relacionan todos los componentes involucrados en una estrategia de ambiente data warehousing, es esencial tener una Arquitectura Data Warehouse.



Arquitectura Data Warehouse

Una arquitectura Data Warehouse (Data Warehouse Architecture - DWA) es una forma de representar la estructura total de datos, comunicación, procesamiento y presentación, para los usuarios finales.

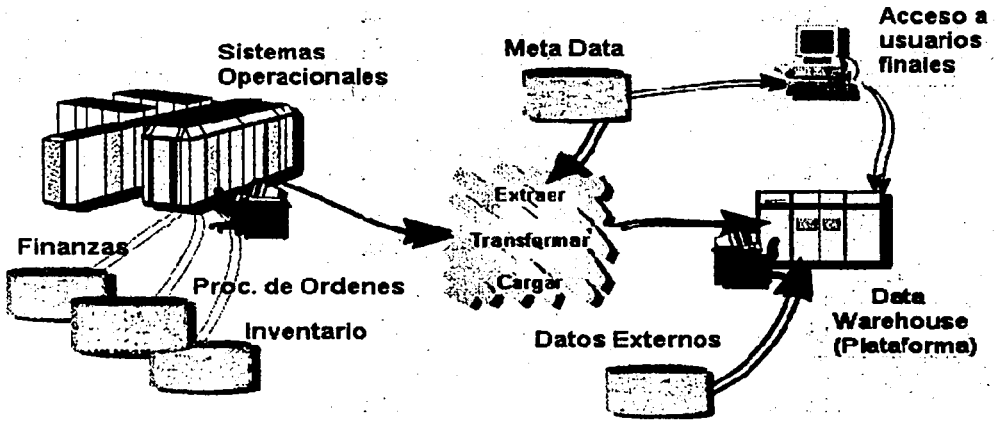
La arquitectura se constituye de un número de partes interconectadas:

- Base de datos operacional / Nivel de base de datos externo.
- Nivel de acceso a la información. Representa las herramientas que el usuario final usa día con día. Por ejemplo Excel, Lotus 1-2-3, Focus, SAS, etc.
- Nivel de acceso a los datos. El lenguaje de datos común que ha surgido es SQL para el desarrollo de una serie de *filtros* de acceso a datos, tales como EDA/SQL para acceder a casi todos los DBMS.
- Nivel de directorio de datos (Metadata). Idealmente, los usuarios finales deberían acceder a los datos del data warehouse sin tener que conocer dónde residen los datos o la forma en que se han almacenado.

- Nivel de gestión de proceso. Programación de diversas tareas que deben realizarse para construir y mantener el data warehouse, así como la información del directorio de datos.
- Nivel de mensaje de la aplicación. Transporte de información alrededor de la red de la empresa.
- Nivel de data warehouse. Es donde residen los datos actuales, usados principalmente para usos estratégicos, como una vista lógica o virtual de datos
- Nivel de organización de datos. Incluye todos los procesos necesarios como seleccionar, editar, resumir, combinar y cargar datos en el depósito y acceder a la información desde bases de datos operacionales y/o externas.

Procesos en un Data Warehouse

En la siguiente figura se muestran algunos de los tipos de operaciones que se efectúan dentro de un ambiente data warehousing.



a) Acceso a datos de Sistemas Operacionales

Los datos son la fuente principal de datos para el data warehouse.

b) Extracción, Transformación y Carga de los Datos.

Se requieren herramientas para extraer datos desde bases de datos y/o archivos operacionales, transformar los datos antes de cargarlos en el data warehouse.

c) Creación del *Metadata.*

La metadata (datos acerca de datos) describe los contenidos del data warehouse. Consiste de definiciones de los elementos de datos en el depósito, sistema(s) de(los) elemento(s) fuente. Como la base de datos, se integra y transforma antes de ser almacenada en información similar.

d) Acceso de usuario final.

Los usuarios acceden al data warehouse por medio de herramientas de productividad basadas en GUI (Graphic User Interface - Interfase Gráfica de Usuario).

Estos pueden incluir software de consultas, generadores de reportes, procesamiento analítico en línea, herramientas data/visual mining, etc., dependiendo de los tipos de usuarios y sus requerimientos particulares. Una sola herramienta no satisface todos los requerimientos, por lo que es necesaria la integración de una serie de herramientas.

e) Elegir la plataforma del data warehouse.

La plataforma para el data warehouse es casi siempre un servidor de base de datos relacional. Cuando se manipulan volúmenes muy grandes de datos puede requerirse una configuración en bloque de servidores UNIX con multiprocesador simétrico (SMP) o un servidor con procesador paralelo masivo (MPP) especializado.

f) Obtener Datos Externos.

El alcance del data warehouse puede extenderse por la capacidad de acceder a los datos externa. Por ejemplo, los datos accesibles por medio de servicios en línea (tales como Portales, oficinas gubernamentales, organizaciones internacionales, etc.) o vía Internet, pueden estar disponibles a los usuarios del data warehouse.

Evolución del Data Warehouse

Construir un data warehouse es una tarea grande. No es recomendable emprender el desarrollo del data warehouse de la empresa como un proyecto cualquiera. Más bien, se recomienda que los requerimientos de una serie de fases se desarrollen e implementen en modelos consecutivos que permitan un proceso de implementación más gradual e iterativo.

No existe ninguna organización que haya triunfado en el desarrollo de su data warehouse empresarial, en un sólo paso. Muchas, sin embargo, lo han logrado luego de un desarrollo paso a paso. Los pasos previos evolucionan conjuntamente con la materia que está siendo agregada.

Los datos en el data warehouse no son volátiles, es un depósito de datos de sólo lectura (en general). Sin embargo, pueden añadirse nuevos elementos sobre una base regular para que el contenido siga la evolución de los datos en la base de datos fuente.

Uno de los desafíos de mantener un data warehouse, es idear métodos para identificar datos nuevos o modificados en las bases de datos operacionales. Algunas maneras para identificar estos datos incluyen insertar fecha/hora en los registros de base de datos y entonces crear copias de registros actualizados y copiar información de los registros de transacción y/o base de datos diarias.

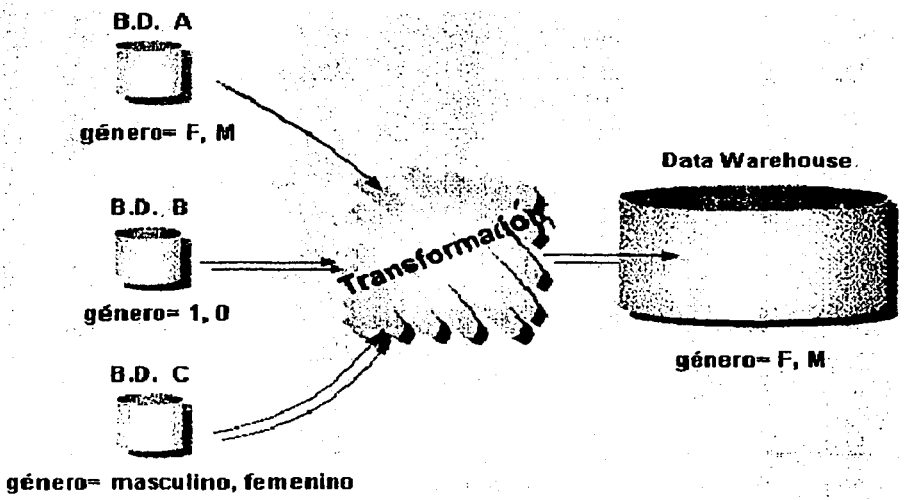
Estos elementos de datos nuevos y/o modificados son extraídos, integrados, transformados y agregados al data warehouse en pasos periódicos programados. Como se añaden las nuevas ocurrencias de datos, los datos antiguos son eliminados.

TRANSFORMACIÓN DE DATOS Y METADATA

Transformación de Datos

Uno de los desafíos de cualquier implementación de data warehouse, es el problema de transformar los datos. La transformación se encarga de las inconsistencias en los formatos de datos y la codificación, que pueden existir dentro de una base de datos única y que casi siempre existen cuando múltiples bases de datos contribuyen al data warehouse.

En la siguiente figura se ilustra una forma de inconsistencia, en la cual el género se codifica de manera diferente en tres bases de datos diferentes. Los procesos de transformación de datos se desarrollan para corregir estas inconsistencias.



La transformación de datos también se encarga de las inconsistencias en el contenido de datos. Una vez que se toma la decisión sobre qué reglas de transformación serán establecidas, deben crearse e incluirse las definiciones en las rutinas de transformación.

Se requiere una planificación cuidadosa y detallada para transformar datos inconsistentes en conjuntos de datos conciliables y consistentes para cargarlos en el data warehouse.

La transformación de los datos se basa en la metadata.

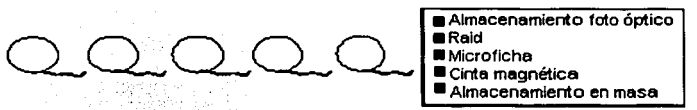
Típicamente, la metadata incluye los siguientes elementos:

- Las estructuras de datos que dan una visión de los datos al administrador de datos.
- Las definiciones del sistema de registro desde el cual se construye el data warehouse.
- Las especificaciones de transformaciones de datos que ocurren tal como la fuente de datos se replica al data warehouse.
- El modelo de datos del data warehouse (es decir, los elementos de datos y sus relaciones).
- Un registro de cuando los nuevos elementos de datos se agregan al data warehouse y cuando los elementos de datos antiguos se eliminan o se resumen.
- Los niveles de sumarización, el método de sumarización y las tablas de registros de su data warehouse.
- Algunas implementaciones de la metadata también incluyen definiciones de la(s) vista(s) presentada(s) a los usuarios.

MEDIOS DE ALMACENAMIENTO PARA INFORMACIÓN ANTIGUA

El símbolo mostrado en la siguiente figura para medios de almacenamiento de información antigua es la cinta magnética, que puede usarse para almacenar este tipo de información. De hecho hay una amplia variedad de medios de almacenamiento que deben considerarse para almacenar datos más antiguos. En la figura se muestran algunos de esos medios.

Dependiendo del volumen de información, la frecuencia de acceso, el costo de los medios y el tipo de acceso, es probable que otros medios de almacenamiento sirvan a las necesidades del nivel de detalle más antiguo en el data warehouse.



Los medios de almacenamiento para la porción voluminosa del data warehouse puede ser de una amplia variedad de tipos de almacenamiento.

USOS DEL DATA WAREHOUSE

Los datos operacionales y los datos del data warehouse son accedidos por usuarios que usan los datos de maneras diferentes.

Uso de Base de Datos Operacionales	Uso de Data Warehouse
Muchos usuarios concurrentes.	Pocos usuarios concurrentes.
Consultas predefinidas y actualizables.	Consultas complejas, frecuentemente no anticipadas.
Cantidades pequeñas de datos detallados.	Cantidades grandes de datos históricos detallados para evaluar tendencias y relaciones.
Requerimientos de respuesta inmediata.	Requerimientos de respuesta no críticos.
Datos "simples" (a partir de una sola fuente).	Requerimientos de respuesta no críticos.

Por lo general, los diferentes niveles de datos dentro del data warehouse reciben diferentes usos. A más alto nivel de esquematización, se tiene mayor uso de los datos.

En la siguiente figura se muestra que hay mayor uso de los datos completamente resumidos, a diferencia de la información antigua que apenas es usada.

Hay una buena razón para mover una organización al paradigma sugerido en la siguiente figura, la utilización del recurso. Los datos más resumidos, se pueden capturar en forma más rápida y eficiente.

Si en una tarea se encuentra mucho procesamiento a niveles de detalle del data warehouse, entonces se consumirán muchos recursos. Es mejor hacer el procesamiento a niveles más altos de esquematización como sea posible.

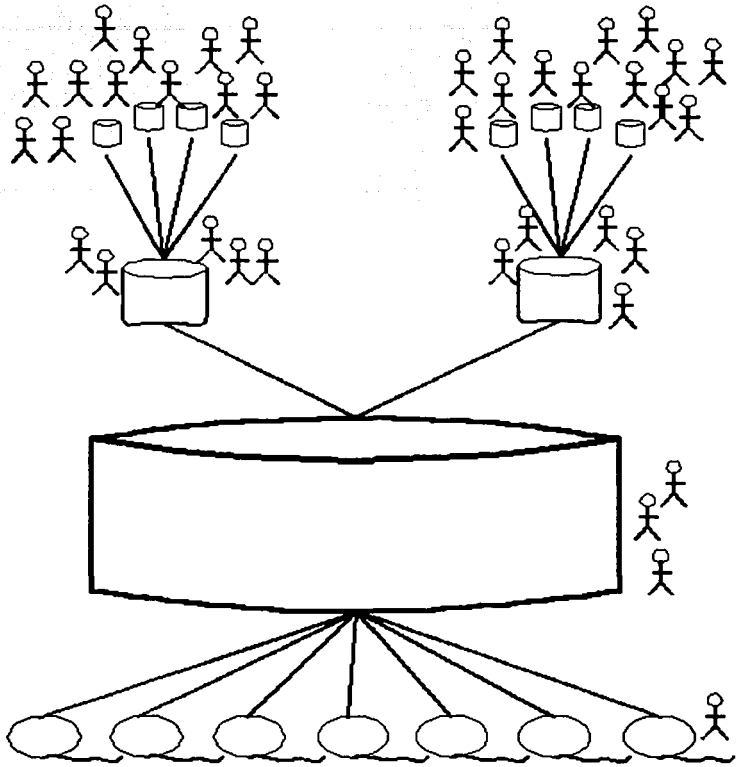
Para muchas tareas, el analista de sistemas de soporte de decisiones usa la información a nivel de detalle en un pre-data warehouse. El acceso a la información de detalle se consigue de muchas maneras, aun cuando estén disponibles otros niveles de esquematización.

Una de las actividades del diseñador del data warehouse es el de desconectar al usuario del sistema de soporte de decisiones del uso constante de datos a nivel de detalle más bajo.

Hay dos opciones para manejar la situación de data en el diseño:

- Instalar un sistema chargeback, donde el usuario final pague por los recursos consumidos.

- Señalar que el mejor tiempo de respuesta puede obtenerse cuando se trabaja con los datos a un nivel alto de esquematización, a diferencia de un mal tiempo de respuesta que resulta de trabajar con los datos a un nivel bajo de detalle.



A más altos niveles de sumarización, más uso de los datos

EXCEPCIONES EN EL DATA WAREHOUSE

Los ***datos resumidos públicos***, que son los datos que han sido calculados fuera del data warehouse pero son usados a través de la corporación. Los datos resumidos públicos se almacenan y administran en el data warehouse, aunque su cálculo se haya hecho fuera de él.

Un ejemplo clásico de datos resumidos públicos es el archivamiento trimestral hecho por cada compañía pública. Los contadores trabajan para producir cantidades como rentas trimestrales, gastos trimestrales, ganancias trimestrales y otros. El trabajo hecho por los contadores está fuera del data warehouse. Sin embargo, esas cantidades

referenciales producidas por ellos se usan ampliamente dentro de la corporación para marketing, ventas, etc. Una vez que se haya hecho el archivo, los datos se almacenan en el data warehouse.

Otra excepción no considerada en esta tesis son los ***datos externos***.

Un tipo de datos a veces encontrado en un data warehouse es ***el detalle de los datos permanentes***, que resulta de la necesidad de una corporación para almacenar permanentemente los datos a un nivel detallado por razones éticas o legales.

Si una corporación expone a sus trabajadores a sustancias peligrosas hay una necesidad de detalle de datos permanentes Si una corporación produce un producto que involucra la seguridad pública, tal como la construcción de las partes de aviones, hay una necesidad de datos permanentes. Si una corporación se compromete con contratos peligrosos, hay una necesidad de detalle de datos permanentes.

La organización simplemente no puede dejar los detalles porque en años futuros, en el caso de una demanda, una notificación, un edificio en disputa, etc., se incrementaría la exposición de la compañía. Por lo tanto hay un único tipo de datos en el data warehouse conocido como detalle de datos permanentes.

El detalle de datos permanentes comparte muchas de las mismas consideraciones como otro data warehouse, excepto que:

- El medio donde se almacenan los datos debe ser tan seguro como sea posible.
- Los datos deben permitir ser restaurados.
- Los datos necesitan un tratamiento especial en su indexación, ya que de otra manera los datos pueden no ser accesibles aunque se haya almacenado con mucha seguridad.

CONSIDERACIONES PREVIAS AL DESARROLLO DE UN DATA WAREHOUSE

Hay muchas formas para desarrollar data warehouses como tantas organizaciones existen. Sin embargo, hay un número de dimensiones diferentes que necesitan ser consideradas:

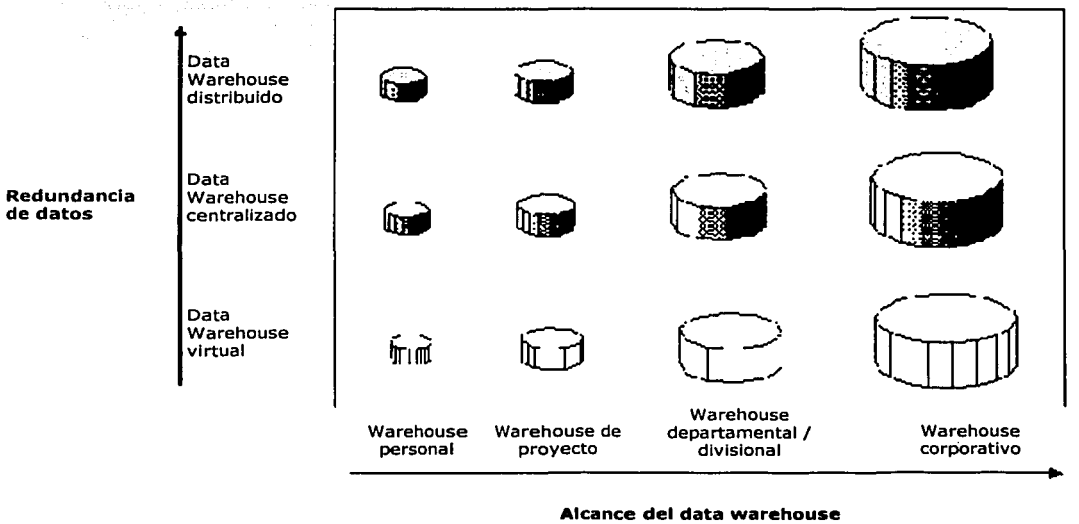
- Alcance de un data warehouse.
- Redundancia de datos.
- Tipo de usuario final

La siguiente figura muestra un esquema bidimensional para analizar las opciones básicas. La dimensión horizontal indica el alcance del depósito y la vertical muestra la cantidad de datos redundantes que deben almacenarse y mantenerse.

ALCANCE DEL DATA WAREHOUSE

El alcance de un data warehouse puede ser tan amplio como toda la información estratégica de la empresa desde su inicio, o puede ser tan limitado como un data warehouse personal para un solo gerente durante un año.

En la práctica, el mayor valor del data warehouse es para la empresa, y lo más caro y consumidor de tiempo es crearlo y mantenerlo. Como consecuencia de ello, la mayoría de las organizaciones comienzan con data warehouses funcionales, departamentales o divisionales y luego los expanden como usuarios que proveen retroalimentación.



Redundancia de los datos

Hay tres niveles esenciales de redundancia de datos que las empresas deberían considerar en sus opciones de data warehouse:

- Data warehouses "virtual" o "Point to Point".
- Data warehouses "centrales".
- Data warehouses "distribuidos".

No se puede pensar en un único enfoque. Cada opción adapta un conjunto específico de requerimientos y una buena estrategia de almacenamiento de datos, lo constituye la inclusión de las tres opciones.

Data Warehouses "Virtual" o "Point to Point"

Una estrategia de data warehouses virtual, significa que los usuarios finales pueden acceder a bases de datos operacionales directamente, usando cualquier herramienta que lo posibilite.

Este enfoque provee flexibilidad así como también la cantidad mínima de datos redundantes que deben cargarse y mantenerse. Además, se pueden colocar las cargas de consulta no planificadas más grandes, sobre sistemas operacionales.

Como se verá, el almacenamiento virtual es, frecuentemente, una estrategia inicial, en organizaciones donde hay una amplia (pero en su mayor parte indefinida) necesidad de conseguir datos operacionales, desde una clase relativamente grande de usuarios finales y donde la frecuencia probable de pedidos es baja. Los depósitos virtuales de datos proveen un punto de partida para que las organizaciones determinen qué usuarios finales están buscando realmente.

Data Warehouses "Centrales"

El concepto de data warehouses centrales es el concepto inicial que se tiene del data warehouse. Es una única base de datos física, que contiene todos los datos para un área funcional específica, departamento, división o empresa.

Los data warehouses centrales se seleccionan por lo general donde hay una necesidad común de los datos informáticos y un número grande de usuarios finales ya conectados a una red o computadora central. Pueden contener datos para cualquier período específico. Comúnmente, contienen datos de múltiples sistemas operacionales.

Los data warehouses centrales son reales. Los datos almacenados en el data warehouse son accesibles desde un lugar y deben cargarse y mantenerse sobre una base regular. Normalmente se construyen alrededor de RDBMS avanzados o, en alguna forma de servidor de base de datos informático multidimensional.

Data Warehouses Distribuidos

Los data warehouses distribuidos son aquellos en los cuales ciertos componentes del depósito se distribuyen a través de un número de bases de datos físicas diferentes.

Cada vez más, las organizaciones grandes están tomando decisiones a niveles más inferiores de la organización y a la vez, llevando los datos que se necesitan para la toma de decisiones a la red de área local (Local Área Network - LAN) o computadora local que sirve a quien toma decisiones. Los data warehouses distribuidos comúnmente involucran la mayoría de los datos redundantes y como consecuencia de ello, se tienen procesos de actualización y carga más complejos.

Tipo de usuario final

De la misma forma que hay una gran cantidad de formas para organizar un data warehouse, es importante notar que también hay una gama cada vez más amplia de usuarios finales.

En general, se pueden considerar tres grandes categorías:

- Ejecutivos y gerentes.
- "Power users" o "Buzo de Información" (analistas financieros y de negocios, ingenieros, etc.).
- Usuarios de soporte (de oficina, administrativos, etc.).

Cada una de estas categorías diferentes de usuario tiene su propio conjunto de requerimientos para los datos, acceso, flexibilidad y facilidad de uso.

SISTEMAS DE GESTIÓN DE BASES DE DATOS

Los data warehouses (conjuntamente con los sistemas de soporte de decisión [Decision Support Systems - DSS] y las aplicaciones cliente / servidor), fueron los primeros éxitos para el DBMS relacional (Relational Data Base Management Systems - RDBMS).

Mientras la gran parte de los sistemas operacionales fueron resultados de aplicaciones basadas en antiguas estructuras de datos, los depósitos y sistemas de soporte de decisiones aprovecharon el RDBMS por su flexibilidad y capacidad para efectuar consultas con un único objetivo concreto.

Los RDBMS son muy flexibles cuando se usan con una estructura de datos normalizada. En una base de datos normalizada, las estructuras de datos son no redundantes y representan las entidades básicas y las relaciones descritas por los datos (por ejemplo productos, comercio y transacción de ventas). Pero un procesamiento analítico en línea (OLAP) involucra varias estructuras y requiere varias operaciones de join para colocar los datos juntos.

El desempeño de los RDBMS tradicionales es mejor para consultas basadas en claves ("Encuentre cuenta de cliente #2014") que para consultas basadas en el contenido ("Encuentre a todos los clientes con un ingreso sobre $ 10,000 que hayan comprado un automóvil en los últimos seis meses").

Para el soporte de depósitos a gran escala y para mejorar el interés hacia las aplicaciones OLAP, los proveedores han añadido nuevas características al RDBMS tradicional. Estas, también llamadas características súper relacionales, incluyen el soporte para hardware de base de datos especializada, tales como la máquina de base de datos Teradata.

Los modelos súper relacionales también soportan extensiones para almacenar formatos y operaciones relacionales (ofrecidas por proveedores como RedBrick) y diagramas de indexación especializados, tales como aquellos usados por Sybase IQ. Estas técnicas pueden mejorar el rendimiento para las recuperaciones basadas en el contenido,

al prejuntar tablas usando índices o mediante el uso de listas de índice totalmente invertidos.

Muchas de las herramientas de acceso a los data warehouses explotan la naturaleza multidimensional del data warehouse. Por ejemplo, los analistas de marketing necesitan buscar en los volúmenes de ventas por producto, por mercado, por período de tiempo, por promociones y niveles anunciados y por combinaciones de estos diferentes aspectos.

La estructura de los datos en una base de datos relacional tradicional, facilita consultas y análisis a lo largo de diferentes dimensiones que han llegado a ser comunes. Estos esquemas podrían usar múltiples tablas e indicadores para simular una estructura multidimensional. Algunos productos DBMS, tales como Essbase y Gentium, implementan técnicas de almacenamiento y operadores que soportan estructuras de datos multidimensionales.

Mientras las bases de datos multidimensionales (Multidimensional Databases - MDDB) ayudan directamente a manipular los objetos de datos multidimensionales (por ejemplo, la rotación fácil de los datos para verlos entre dimensiones diferentes, o las operaciones de drill down que sucesivamente exponen los niveles de datos más detallados), se debe identificar estas dimensiones cuando se construya la estructura de la base de datos. Así, agregar una nueva dimensión o cambiar las vistas deseadas, puede ser engorroso y costoso. Algunos MDDB requieren un recargue completo de la base de datos cuando ocurre una reestructuración.

Nuevas dimensiones

Una limitación de un RDBMS y un MDDB, es la carencia de soporte para tipos de datos no tradicionales como imágenes, documentos y clips de video/audio. Si se necesitan estos tipos de objetos en el data warehouse, se busca un DBMS relacional-objeto (Ejemplo: Illustra de Informix).

Por su enfoque en los valores de datos codificados, la mayor parte de los sistemas de base de datos pueden acomodar estos tipos de datos, sólo con extensiones basadas en ciertas referencias, tales como indicadores de archivos que contienen los objetos. Muchos RDBMS almacenan los datos complejos como objetos grandes binarios (Binary Large Objects - BLOBs). En este formato, los objetos no pueden ser indexados, clasificados, o buscados por el servidor.

Los DBMS relacional-objeto, almacenan los datos complejos como objetos nativos y pueden soportar las grandes estructuras de datos encontradas en un ambiente orientado a objetos. Estos sistemas de base de datos naturalmente acomodan no sólo tipos de datos especiales sino también los métodos de procesamiento que son únicos para cada uno de ellos.

Pero una desventaja del enfoque relacional-objeto, es que el encapsulamiento de los datos dentro de los tipos especiales de datos (una serie de precios de stock a través del tiempo en cada registro de una tabla de stock, por ejemplo), requiere de operadores especializados para que hagan búsquedas simples previamente (por ejemplo, "Encontrar todas las existencias que han mostrado una disminución en el precio de Abril a Mayo 2003").

La selección del DBMS está también sujeta al servidor de hardware que se usa. Algunos RDBMS, como el DB2 Paralelo, Informix XPS y el Oracle Paralelo, ofrecen versiones que soportan operaciones paralelas. El software paralelo divide consultas, uniones a través de múltiples procesadores y corre estas operaciones simultáneamente para mejorar el desempeño.

Se requiere el paralelismo para el mejor desempeño en los servidores MPP grandes y SMP agrupados. No es aún una opción con MDDBS o DBMS relacional-objeto.

La tabla "Matriz de Decisión del Data Warehouse" contiene algunos ejemplos de cómo afectan estos criterios de decisión en la elección de una arquitectura de servidor/data warehouse.

Matriz de Decisión para el Data Warehouse						
Para estos ambientes...			**Elegir...**			
Requerimientos comerciales	**Usuarios**	**Soporte de Sistemas**	**Arquitectura**	**Servidor**	**DBMS**	**Usos**
Alcance: departamental	Pocos, ubicación única	Local mínimo, central promedio	Consolidado, paquete	Procesador único o SMP	MDDB	Análisis de datos
Alcance: departamental	Grande; analistas en una sola ubicación; usuarios informáticos dispersos	Local mínimo, central promedio	Seccionado - detalle en central resumen en local	Grupos de SMP para central; SP o SMP para local	RDBMS para central - MDDB para local	Análisis más informática
Alcance: empresa	Grande; geográficamente disperso	Central fuerte	Centralizado	Grupos de SMP	Objeto-relacional, soporte Web	Análisis más informática
Alcance: departamental	Pocos, pocas ubicaciones	Central fuerte	Centralizado	MPP	RDBMS con soporte paralelo	Investigación

Confiabilidad de los datos

Los datos "sucios" son peligrosos. Las herramientas de limpieza especializadas y las formas de programar de los clientes proporcionan mecanismos de confiabilidad.

No importa cómo esté diseñado un programa o cuán hábilmente se use. Si se alimenta con información mala, se obtendrán resultados incorrectos o falsos. Desafortunadamente, los datos que se usan satisfactoriamente en las aplicaciones operacionales, pueden ser basura en lo que concierne a la aplicación data warehousing.

Los datos "sucios" pueden presentarse al ingresar información en una entrada de datos (por ejemplo, "Sistemas SA" en lugar de "Sistemas S. A.") o de otras causas. Cualquiera que sea, la información sucia daña la credibilidad del depósito completo. A continuación, se muestra un ejemplo de formato de ventas en el que se pueden presentar errores.

Afortunadamente, las herramientas de limpieza de datos pueden ser de gran ayuda. En algunos casos, puede crearse un programa de limpieza efectivo. En el caso de bases de datos grandes, imprecisas e inconsistentes, el uso de las herramientas comerciales puede ser casi obligatorio.

Decidir qué herramienta usar es importante y no solamente para la integridad de los datos. Si se equivoca, se podría malgastar semanas en recursos de programación o cientos de miles de dólares en costos de herramientas.

Limpieza de los datos

La limpieza de datos "sucios" es un proceso multifacético y complejo. Los pasos a seguir son los siguientes:

1° Analizar sus datos corporativos para descubrir inexactitudes, anomalías y otros problemas.

2° Transformar los datos para asegurar que sean precisos y coherentes.

3° Asegurar la integridad referencial, que es la capacidad del data warehouse, para identificar correctamente al instante cada objeto del negocio, tales como un producto, un cliente o un empleado.

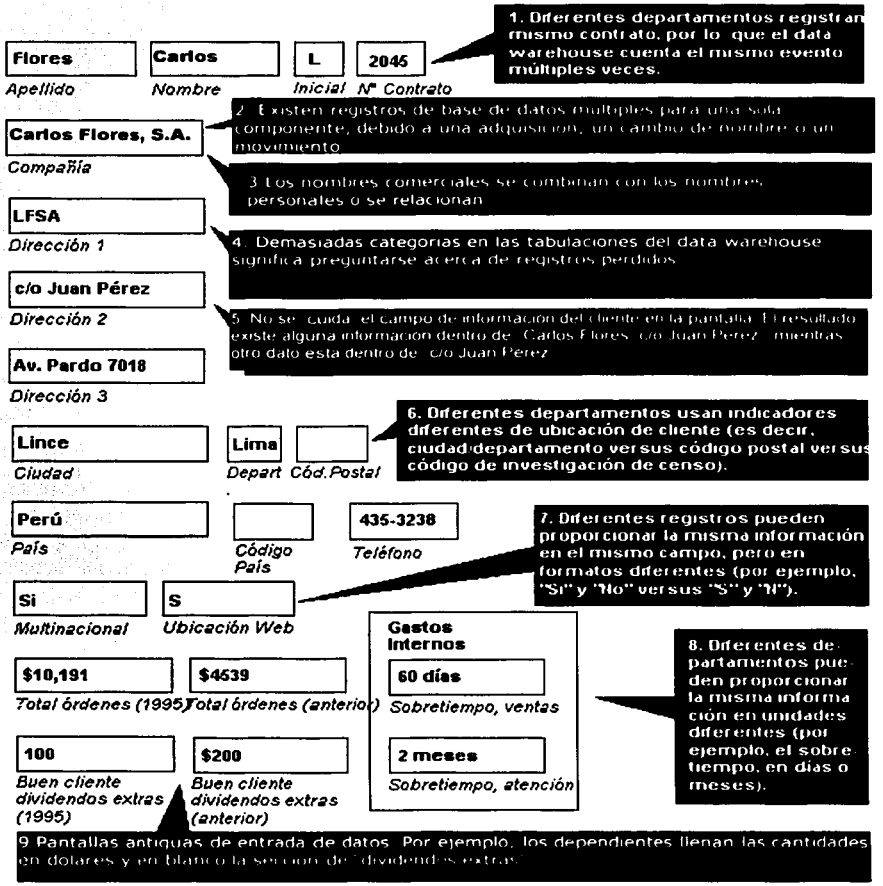
4° Validar los datos que usa la aplicación del data warehouse para realizar las consultas de prueba.

5° Producir la metadata, una descripción del tipo de datos, formato y el significado relacionado al negocio de cada campo.

6° Finalmente, viene el paso crucial de la documentación del proceso completo para que se pueda ampliar, modificar y arreglar los datos en el futuro con más facilidad.

En la práctica, se tendría que realizar múltiples pasos como parte de una operación única o cuando use una sola herramienta. En particular, limpiar los datos y asegurar la integridad referencial son procesos interdependientes.

FORMATO DE VENTAS



Las herramientas comerciales pueden ayudar en cada uno de estos pasos. Sin embargo, es posible escribir programas propios para hacer el mismo trabajo.
Los programas de limpieza de datos no proporcionan mucho razonamiento, por lo que las compañías necesitan tomar sus decisiones en forma manual, basados en información importante y reportes de auditoria de datos.

CONSIDERACIONES PARA LA IMPLEMENTACIÓN DEL DATA WAREHOUSE

1° Identificar el problema en el cual el uso estratégico de la información detallada, permita conseguir una solución para generar una ventaja competitiva o un ahorro de costos.

Ejemplo: Un problema puede ser la ausencia de un modelo para estudios de retención de clientes.

2° Definir el modelo lógico de datos a implementar para resolver el problema planteado.

Ejemplo: Se puede dar un modelo lógico cuando se presenta al usuario la información en términos de dimensiones (clientes, productos, canales de ventas, promociones, adquirientes, etc.) básicas del modelo de datos y hechos que se registrarán para estas dimensiones (medidas de ventas, de costos, de producción, de facturación, de cartera, de calidad, de servicio, etc.).

3° Reunir los datos para poblar ese modelo lógico de datos.

4° Tomar iniciativas de complementación de información para asegurar la calidad de los datos requeridos para poblar el modelo de datos.

Estas definiciones deben estar acompañadas de un servidor apropiado para el data warehouse, así como elementos de comunicaciones, nodos cliente, el manejador de la base de datos del data warehouse y otros hardware y software requeridos para la implementación del proyecto.

5° Evaluar el rendimiento de la inversión.

Cuando se evalúan los costos, el usuario del data warehouse puede no tener el contenido de los costos en mente, pero las preguntas mínimas que puede comenzar a hacerse son las siguientes:

¿Qué clases de costos excedieron el presupuesto en más del 10% en cada uno de los 12 meses pasados?

¿Se aumentaron los presupuestos en más de 5% para cualquier área dentro de los últimos 18 meses?

¿Cómo especificar las clases de gastos entre diferentes departamentos? ¿Entre divisiones? ¿A través de las regiones geográficas?

¿Cómo tener márgenes de operación sobre los dos últimos años en cada área de negocio? Donde han disminuido los márgenes, ¿se han incrementado los costos?

Con frecuencia, los aspectos realmente importantes identificados por una gestión mayor, tienen un valor agregado, en el que ellos saben si tuvieron la información que estaban buscando, lo que significaría una mejora de (por ejemplo) las ventas en 0.5% a 1% - que, si la operación estuvo por los billones de dólares en un año, puede resultar en cientos de millones de dólares. En algunos casos, el costo del depósito inicial se ha recobrado en un período de 6 a 8 meses.

Al hacer preguntas de este tipo, los usuarios comienzan a identificar las áreas en la que los costos han aumentado o disminuido significativamente y pueden evaluar cada una de estas áreas con más detalle.

SOFTWARE EN UN DATA WAREHOUSE

La información estratégica sobre clientes importantes o un exitoso lanzamiento de producto, se almacena en gigabytes de datos de marketing o índice de transacciones de venta. Esa información debe ser extraída de alguna forma para la toma de decisiones.

En este caso se necesita software especializado que permita capturar los datos relevantes en forma rápida y pueda verse a través de diferentes dimensiones de los datos. El software no debería limitarse únicamente al acceso a los datos, sino también, al análisis significativo de los datos. Transformar los datos en información útil para la empresa.

Los softwares o herramientas de negocios inteligentes se colocan sobre la plataforma data warehousing y proveen este servicio. Debido a que son el punto principal de contacto entre la aplicación del depósito y la gente que lo usa, estas herramientas pueden constituir la diferencia entre el éxito o fracaso de un depósito.

Las herramientas de negocio inteligentes se han convertido en los sucesores de los sistemas de soporte de decisión, pero tienen un alcance más amplio. No solamente ayudan en las decisiones de soporte sino, en muchos casos, estas herramientas soportan muchas funciones operacionales y de misión crítica de la compañía. Sin embargo, estos productos no son infalibles ya que sólo se consigue el máximo provecho del data warehouse, si se eligen las herramientas adecuadas a las necesidades de cada usuario final.

El software usado en un data warehouse se clasifica en

1. Herramientas de Consulta y Reporte.
2. Herramientas de Base de Datos Multidimensionales / Olap (On Line Analytical Processing).
3. Sistemas de Información Ejecutivos.
4. Herramientas Data Mining.
5. Los Sistemas de Gestión de Bases de Datos propiamente.

Herramientas de consulta y reporte

Algunos proveedores ofrecen productos que permiten tener más control sobre el procesamiento de consulta.

Las más simples de estas herramientas son productos de reporte y consultas básicas. Ellos proporcionan desde pantallas gráficas a generadores SQL.

Las herramientas visuales de consulta, permiten apuntar y dar un click a los menús y botones para especificar los elementos de datos, condiciones, criterios de agrupación y otros atributos de una solicitud de información.

La herramienta de consulta genera entonces un llamado a una base de datos, extrae los datos pertinentes, efectúa cálculos adicionales, manipula los datos si es necesario y presenta los resultados en un formato claro.

Se pueden almacenar las consultas y los pedidos de reporte para trabajos subsecuentes, como está o con modificaciones. El procesamiento estadístico se limita

comúnmente a promedios, sumas, desviaciones estándar y otras funciones de análisis básicas. Aunque las capacidades varían de un producto a otro, las herramientas de consulta y reporte son más apropiadas cuando se necesita responder a la pregunta ¿"Qué sucedió"? (Ejemplo: ¿"Cómo comparar las ventas de los productos X, Y y Z del mes pasado con las ventas del presente mes y las ventas del mismo mes del año pasado?").

Para hacer consultas más accesibles a usuarios no-técnicos, los productos tales como Crystal Reports de Seagate, Impromptu de Cognos, Reportsmith de Borland, Intelligent Query de IQ Software, Esperant de Software AG y GQL de Andyne, ofrecen interfases gráficas para seleccionar, arrastrar y pegar.

Herramientas de base de datos multidimensionales/OLAP

Los generadores de reporte tienen sus limitaciones cuando los usuarios finales necesitan más que un solo reporte, más de una vista estática de los datos, que no sean sujeto de otras manipulaciones. Para estos usuarios, las herramientas del procesamiento analítico en línea (OLAP - On Line Analytical Processing), proveen capacidades "Slide y Dice" que contestaría "¿qué sucedió?" al analizar por qué los resultados están como están.

Las primeras soluciones OLAP estuvieron basadas en bases de datos multidimensionales (MDDBS). Un cubo estructural (arreglo multidimensional) almacenaba los datos para que se puedan manipular intuitiva y claramente y también ver las asociaciones a través de múltiples dimensiones. Los productos pioneros tal como Essbase de Arbor Software soportan directamente las diferentes vistas y las manipulaciones dimensionales requeridas por OLAP.

Limitaciones del enfoque de bases de datos multidimensionales:

1: Las nuevas estructuras de almacenamiento de datos requieren bases de datos propietarias. No hay realmente estándares disponibles para acceder a los datos multidimensionales.

Los proveedores como Arbor, vieron esto como una oportunidad para crear normas de facto para editar APIs MDDB, propiciando herramientas terceristas y estableciendo asociaciones estratégicas.

Muchas de estas herramientas de consulta y de soluciones data mining soportan directamente Essbase, Oracle Express y otros formatos MDDB comunes. El Commander OLAP, herramienta cliente/servidor de Comshare, se sitúa sobre la parte superior de un data warehouse multidimensional Essbase y soporta el acceso dinámico y la manipulación de los datos.

2: La segunda limitación de un MDDB concierne al desarrollo de una estructura de datos. Las compañías generalmente almacenan los datos de la empresa en bases de datos relacionales, lo que significa que alguien tiene que extraer, transformar y cargar estos datos en el hipercubo.

Este proceso puede ser complejo y consumidor de tiempo. Las herramientas de extracción de datos y otras automatizan el proceso, trazando campos relacionales en la estructura multidimensional y desarrollando el MDDB sobre la marcha.
Algunos proveedores ofrecen ahora la técnica OLAP relacional (Relational On Line Analytical Processing - ROLAP), que explora y opera en el data warehouse directamente usando llamadas SQL estándares. Las herramientas de pantallas permiten retener los pedidos multidimensionales, pero el motor ROLAP transforma las consultas en rutinas SQL. Entonces se recibe los resultados tabulados como una hoja de cálculos multidimensional o en alguna otra forma que soporte rotación, drilling down y reducción.
Así como la extracción de los datos, el desarrollo y evolución de la estructura MDDB puede cambiarse. Los administradores ROLAP deben afrontar algunas veces las tareas de desarrollar las rutinas SQL para agregar e indexar los datos ROLAP, así como, asegurar la traducción correcta de las peticiones multidimensionales en la ventana de comandos SQL.
Los defensores de ROLAP argumentan que se usan estándares abiertos (SQL) y que se esquematizan (a nivel de detalle) los datos para hacerlos más fácilmente accesibles. Por otra parte, argumentan que una estructura multidimensional nativa logra mejor desempeño y flexibilidad, una vez que se desarrolla el almacén de los datos.

Lo bueno es que estas tecnologías evolucionan rápidamente o pueden proveer una pronta solución OLAP. Algunos productos ejemplos son PowerPlay de Cognos, Business Objects con el software del mismo nombre, Brio Query de Brio Technology y una serie de DSS Agent/DSS Server de MicroStrategy.

Los retos administrativos y de desarrollo de OLAP, a diferencia de las encontradas con las herramientas de consulta y reporte, son generalmente más complejos. Definiendo el OLAP y el software de acceso a los datos, se requiere un claro entendimiento de los modelos de datos de la corporación y las funciones analíticas requeridas por ejecutivos, gerentes y otros analistas de datos.

Los usuarios de estos productos deben decidir sobre si los datos del procesamiento analítico en línea, deberían almacenarse en bases de datos multidimensionales especialmente diseñadas o en bases de datos relacionales. Esto depende de las necesidades de la organización.

Sistemas de información ejecutivos

Las herramientas de sistemas de información ejecutivos (Executive Information Systems - EIS), proporcionan medios fáciles de usar para consulta y análisis de la información confiable. Generalmente se diseñan para el usuario que necesita conseguir los

datos rápidamente, pero quiere utilizar el menor tiempo posible para comprender el uso de la herramienta.

Un uso típico de un EIS es facilitar al usuario la recuperación y análisis de las métricas, del desempeño de la organización.

El precio de esta facilidad de uso es que por lo general existen algunas limitaciones sobre las capacidades analíticas disponibles con el sistema de información ejecutivo. Además, muchas de las herramientas de consulta/reporte y OLAP/multidimensional, pueden usarse para desarrollar sistemas de información ejecutivos.

El concepto de sistema de información ejecutivo es simple: los ejecutivos no tienen mucho tiempo, ni la habilidad en muchos casos, para efectuar el análisis de grandes volúmenes de datos. El EIS presenta vistas de los datos simplificados, altamente consolidados y mayormente estáticas.

Categorías de Ambientes EIS:

A diferencia del modelo OLAP, donde el incremento de niveles de información se da a conocer cuando el analista manipula los datos, un ejecutivo espera una descripción global. No deberían escudriñar para obtener respuestas.

Los pioneros en el mercado de EIS incluyen Comshare, creadores del Commander EIS y Pilot Software, desarrolladores del Pilot Command Center.

Herramientas data mining

Data mining es una categoría de herramientas de análisis open-end. En lugar de hacer preguntas, se toma estas herramientas y se pregunta algo "*interesante*", una tendencia o una agrupación peculiar. El proceso de data mining extrae los conocimientos guardados o información predictiva desde el data warehouse sin requerir pedidos o preguntas específicas.

Las herramientas Mining usan algunas de las técnicas de computación más avanzadas como:

- Redes neuronales.
- Algoritmos genéticos.
- Análisis de la canasta de mercado y
- Razonamiento basado en la memoria.

Para generar modelos y asociaciones. Mining es conducida por datos, no conducida por una aplicación.

El Intelligent Miner de IBM para AIX soporta sofisticadas técnicas mining, así como las funciones de preparación de los datos para extraer información desde bases de datos Oracle o Sybase y cargarlos en DB2 para mining. Con su opción Data Mine para el motor Red Brick Warehouse 5.0, Red Brick integra la funcionalidad de un data mining y la arquitectura de almacenamiento.

Otros ejemplos de herramientas data mining comerciales incluyen Darwin de Thinking Machines, herramientas de visualización de datos en MDDB de SAS Institute, SGI MineSet y Focus 6 Serie de Visualización y Análisis de Information Builders.

SISTEMAS DE GESTIÓN DE BASES DE DATOS

Estos software's proporcionan procesamiento en paralelo y/o algo fuera de los aspectos ordinarios, que puedan ser especialmente interesantes para la gente de desarrollo de data warehouse y de sistemas de soporte de decisiones.

ELECCIÓN DE HERRAMIENTAS DE ANÁLISIS DE DATOS

Hay algunas reglas obvias a seguir cuando se eligen herramientas de análisis de datos. Las herramientas se combinan según las necesidades de los usuarios finales, capacidad técnica empresarial y la fuente de datos existente.

1° Si se elige un proveedor de depósito que además ofrece herramientas integradas, probablemente se ahorrará un tiempo de desarrollo significativo al elegir un conjunto de herramientas compatibles.
De otro modo, se selecciona un conjunto de herramientas que soporte la fuente de datos original. Sin ese soporte, se debería optar por una solución OLAP relacional debido a que provee una arquitectura abierta.

2° Después que se ha seleccionado un conjunto de herramientas compatible con la fuente de datos, se determina cuánto análisis necesita realmente.
Simplemente se necesita saber "cuánto" o "cuántos", será suficiente una herramienta básica de consultas y reportes.
Si se requiere un análisis más avanzado que explique la causa y los efectos de las ocurrencias y las tendencias, se busca una solución OLAP.
Las herramientas data mining sofisticadas requieren expertos en técnicas de análisis de datos y se necesitan para pronósticos avanzados, clasificación y creación del modelo.

3° Como con cualquier tecnología, para el mejor desempeño de la compañía, se puede optar por una solución única o un conjunto de soluciones. El personal debe comprender los requerimientos de tecnología, desarrollar soluciones que reúnan esos requerimientos, así como mantener y mejorar efectivamente los sistemas.
Los software's de negocio inteligentes son sólo herramientas. Todavía se necesitan gerentes y ejecutivos que capten los conocimientos derivados y tomen decisiones. Estos software's requieren todavía inteligencia propia.

En la siguiente tabla se definen los parámetros a tener en cuenta para la elección de las herramientas adecuadas:

Elegir la herramienta adecuada			
Tipo de Herramienta	**Pregunta básica**	**Modelo de Salida**	**Usuario típico**
Consulta y Reporte	¿Qué sucedió?	Reportes de ventas mensuales; histórico de inventario	Necesita data histórica puede tener aptitud técnica limitada
Procesamiento analítico en línea (OLAP)	¿Qué sucedió y por qué?	Ventas mensuales vs. cambios de precio de los competidores	Necesita ir de una visión estática de los datos a "slicing and dicing" técnicamente astuto
Sistema de Información Ejecutiva (SIE)	¿Qué necesito conocer ahora?	Libros electrónicos; Centros de comandos	Necesita información resumida o de alto nivel puede no ser técnicamente astuto
Data mining	¿Qué es interesante? ¿Qué podría pasar?	Modelos predictivos	Necesita extraer la relación y tendencias de la información ininteligible técnicamente astuto.

Capítulo III

LA PREPARACIÓN DE LOS DATOS (DATA PREPARATION)

Es una serie de actividades y procesos que se realizan sobre los datos para saber cuales de ellos son los que se requerirán para un determinado proceso de minería de datos, para saber algo sobre la empresa, para verificar o corregir la integridad referencial de los mismos; las aplicaciones normales, se limitan la disponibilidad y acceso a los datos y su conservación adecuada durante su estancia en la aplicación, pero no a que los datos sean realmente coherentes y necesarios para una toma de decisiones.

La preparación de los datos no comienza desde los datos a preparar, es aun más atrás, se comienza desde la identificación de las necesidades que inician el almacenamiento de los mismos, esto es, cuales de las actividades que se realizarán son las que generarán datos a ser almacenados, qué tipo de dato, con que frecuencia se actualizarán, las relaciones que estos tienen con otros datos, con otras actividades que también generan datos para almacenar. Cualquier dato puede ser importante, pero no todos se usan ni tienen la misma relevancia en la toma de decisiones igualmente.

El objetivo del tratamiento de los datos es poder transformar cualquier conocimiento en un conjunto de datos manipulables o comprensibles por los humanos.

La exploración de datos es un proceso de negocios multietapas, con el que la gente trabaja usando una metodología estructurada para descubrir y evaluar los problemas apropiados, definir soluciones e implementar estrategias que produzcan resultados medibles.

No siempre se puede asegurar que los datos contenidos en la base de datos este de tal forma que se pueda lograr un entendimiento a priori, o que una persona experta en la minería de datos pueda extraerlos (de forma manual o automática) para encontrar el conocimiento oculto.

La solución es la preparación de los datos. No todas las variables son de tipo numérico, por lo que su precisión o manipulación se complica. No es tratada igual una variable de tipo numérica, de tipo cadena (nombre o código postal) o de tipo booleana o binaria (sexo).

Para todas las posibles interpretaciones que se le pueden dar a las variables de distintos tipos, es que se realiza la preparación de los datos para realizar una minería de datos que extraiga patrones interesantes y confiables.

Tiempos en un proyecto de exploración de datos

Actividad	Teórico		Práctico	
Exploración del problema	10%		15%	
Exploración de la solución	9%	20%	14%	80%
Especificaciones de la implementación	1%		51%	
Minería de datos				
• Preparación de datos	60%		15%	
• Modelación de datos	15%	80%	3%	20%
• Inspección de datos	5%		2%	

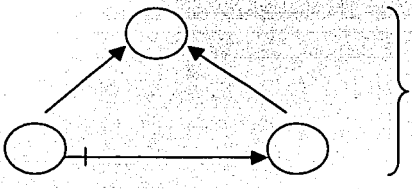
La exploración de los datos, es un proceso de extracción automática de patrones en los datos. Una efectiva exploración provee un aprovechamiento disciplinado para identificar los problemas y ganar entendimiento desde los datos para ayudarnos a solucionarlos. El proceso de la exploración de los datos inicia en la correcta identificación del problema a resolver.

Para poder lograr hacer esto, se requiere definir los problemas de una forma precisa. Así mismo, poder descomponer los grandes problemas en unos pequeños, para poderles dar solución a cada uno ellos paulatinamente. Analizando bien el problema, se puede llegar a la conclusión de muchos más componentes, de más dimensiones y situaciones relevantes para tener una perspectiva real del problema. Una descripción clara del problema es la mitad de la batalla.

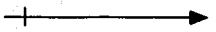
ENTENDIMIENTO DE PROBLEMAS

1. Mapas Cognoscitivos

Para poder tener una mejor perspectiva del problema, se pueden usar mapas cognoscitivos, para ayudar a estructurar el problema de una forma conveniente. Esta herramienta es muy utilizada para la exploración de problemas complejos. Se dibujan los objetos que intervienen en el espacio problema, todos juntos, marcados con las intercomunicaciones e interacciones de las variables de los objetos. Esto es bueno para mostrar en donde existen los conflictos que estructuran el problema.



Este símbolo representa un objeto, el cual es un actor en el problema, por lo que su presencia y descripción son importantes.



Este símbolo representa una conexión negativa o conexión inversa.

Este símbolo representa una conexión positiva o conexión reforzada.

Además, se pueden agregar "pesos numéricos", para indicar la fuerza (peso) de las conexiones. Hay formas de asegurarse de que las soluciones que se estén presentado sean las correctas, de que se tengan previstas o que ya exista una documentación de las mismas. Al resolver un problema, casi siempre se sabe qué esperar de forma aproximada como solución, de lo contrario, esta (y algunas otras) técnica(s), no serán de utilidad.

2. Ambigüedad

La resolución de ambigüedad es una forma de asegurarse que donde existan interpretaciones alternativas de una solución de un problema, cualquiera de las suposiciones será explicada, con esto, se logrará que cualquiera de las soluciones ya esté perfectamente detallada. Una vez que se tienen las posibles soluciones bien detalladas, se hace una categorización, lo cual indicará que aunque se puedan dar varias soluciones a un

mismo problema, cada una tendrá un peso específico, el cual determinará que tiene más elementos esa solución para ser elegida que las otras.

Al hacer una exploración del espacio del problema y dependiendo de su estructura es como se llevará a cabo la solución. Para determinar por qué problema empezar, implica otros esfuerzos, para lo cual se puede implementar la siguiente técnica. Se usa clasificación del par sabio, que es una poderosa técnica para reducir la selección comparativa. Se ilustraran los pasos a seguir al aplicar esta técnica.

La categorización intuitiva es aquel valor que se le asigna a cada uno de los pequeños problemas a resolver según nuestra apreciación, esto es, según lo que nosotros determinamos que es más importante.

Ejemplo de la clasificación del par sabio

Primero, se compararán los dos primeros renglones (problemas a y b), después los renglones 1 y 3 (problemas a y c), por último se compararán los dos renglones 2 y 3 (problemas b y c).

Problema	Importancia	Dificultad	Relevancia
a	9	2	3
b	5	7	7
c	2	9	1

Pesos de los factores del problema

	50%	25%	25%	
Problema	Importancia	Dificultad	Relevancia	
a	9	2	3	5.75
b	5	7	7	6.00
c	2	9	1	3.5

Importancia total del problema

El problema **b** obtuvo la importancia total del problema mayor (6.00), lo que indica que se debe iniciar por este problema. Seguido por el problema **a.**

Los factores de los 3 problemas considerados son:

- El campo "importancia", nos indica cual de esos problemas es el más importante.
- El campo "Dificultad", nos indica cual de esos dos problemas que se están comparando, representa una menor dificultad para poder obtener su solución.
- El campo "Relevancia", nos indica cual de esos dos problemas que se están comparando, regresará la mejor salida (un valor de retorno muy importante).

ALGUNAS FORMAS DE PREPARACIÓN DE DATOS

a) Normalización para Redes Neuronales

Una de las preparaciones de datos que es muy utilizada en las redes neuronales, es obtener un rango de los valores, es decir, una diferencia entre el mínimo y el máximo de los valores.

Esto finalmente será tratado por medio de la función matemática logaritmo base diez, es decir, se aplicara el logaritmo a cada una de las diferencias, que deberán estar en el rango obtenido de la resta:

$$rango = val\_\max - val\_\min$$

$$valor\_manipulado = \log(valor\_actual - valor\_\min)$$

Con lo cual se logra que los valores estén dentro de un rango muy pequeño, ya que log(10) = 1, log(100) = 2, log(1000) = 3, log(10,000) = 4 y así sucesivamente, por lo que es más fácil hacer un tratamiento de estos valores.

Se deben utilizar este tipo de operaciones sobre los datos, es cuando sean funciones o consideraciones no lineales, ya que los valores resultantes no serán tampoco linealmente correspondientes.

b) Manipulación de variables

Una forma de manipular las variables categóricas que existen en un problema dado, es poder hacer un mapeo de su dominio a un dominio numérico, lo cual se realizará haciendo una discretización de los valores.

Esto es, tomar los códigos de los valores y convertirlos en una escala numérica, en la cual, estarán reflejadas las cercanías de valor numérico y la cercanía real, el siguiente paso es ordenar los valores.

El problema de este tipo de manipulación de datos, es que una red neuronal por ejemplo, no sabría que tanta diferencia puede existir entre los estados civiles de una persona: soltero(0.00), divorciado(0.25), casado(0.50), viudo(0.75) y desconocido(1.00), ya que soltero y desconocido por el valor numérico asignado están muy lejanos, pero divorciado y casado son muy cercanos. Esto implica que la conversión numérica lineal no siempre es la mejor solución, aunque es la más sencilla.

SALIDA DE LA PREPARACIÓN DE DATOS

La salida de la exploración debe especificarse para que la solución sea prácticamente implementada de forma automática. Para el problema y la solución, es importante la resolución de ambigüedad.

La técnica de resolución de ambigüedad busca con toda precisión el punto en el que se eliminan los errores y equivocaciones en la comunicación, se revelan las

suposiciones ocultas y se asegura que los puntos clave y salidas sean entendidos por todas las personas que estén involucradas.

Después de que se ha especificado la salida, la ambigüedad se ha eliminado y se tiene una clasificación del problema (importancia que tiene para la empresa esta sección de sus datos), se debe especificar la implementación.

Para implementar la solución se usan las preguntas de las seis W's (del ingles): who(quien), how(como), what(quien), when(cuando), where(donde) y why(porque) (la cual ya fue cubierta en la especificación del problema).

Cuando se realiza la *construcción de modelos*, se deben considerar las *diez reglas de oro*:

1. Seleccionar problemas claramente definidos que puedan arrojar beneficios tangibles.
2. Especificar la solución requerida.
3. Definir como será utilizada la solución encontrada.
4. Entender cuanto sea posible acerca del problema y el conjunto de datos (que es el dominio).
5. Permitir que el problema manipule la solución (por ejemplo, la selección de la herramienta, preparación de los datos, etc., según la naturaleza del mismo).
6. Estipular suposiciones.
7. Refinar el modelo iterativamente.
8. Hacer el modelo tan sencillo como sea posible, pero no simple.
9. Definir la inestabilidad del problema (las áreas críticas donde cambiaría la salida drásticamente por un cambio en las entradas).
10. Definir la incertidumbre del modelo (áreas críticas y rangos en el conjunto de datos donde el modelo produce predicciones de bajo nivel de confianza).

La estadística y la minería de datos están relacionadas; ha llegado a ocurrir que es difícil discernir la línea que divide a estas dos actividades.

Los análisis estadísticos han sido orientados a la verificación y validación de hipótesis. Actualmente existe un área del análisis estadístico llamado "*análisis exploratorio de datos*"

MODELOS ACTIVOS Y MODELOS PASIVOS

Básicamente se pueden clasificar los modelos obtenidos por una preparación de datos en dos grandes rubros: *modelos activos y modelos pasivos.* Finalmente, la elección de un modelo activo o modelo pasivo dependerá de la aplicación y el modelo necesario.

Modelos pasivos. Generalmente responden preguntas y muestran relaciones usando gráficas, palabras, fórmulas matemáticas, etc. Explica de una forma entendible el "porqué" de una relación. Un modelo es pasivo cuando no toma

entradas, proporciona salidas, cambia, reacciona o modifica cualquier cosa que utiliza. Es simplemente una expresión manipulada como una declaración o una pieza de papel.

Modelos activos. Realizan una o más actividades. Un modelo activo construye sus salidas a partir de entradas que combinará para producir salidas no necesariamente lineales.

Principales tareas del preprocesamiento de datos:

1. Hacer una limpieza de los datos. Esto es rellenar los datos nulos (cuando esto sea posible), identificar y/o eliminar los outliers, resolver las inconsistencias y tratar los valores con ruido.
2. Integración de los datos. Lo cual representa hacer una conjunción de todas aquellas fuentes de datos posibles, que se tengan listas para su procesamiento, pueden ser desde archivos planos hasta bases de datos corporativas externas.
3. Transformación de los datos. Se refiere a realizar una normalización y agregación de los mismos, en donde estén siendo depositados (base de datos, archivo plano o un data warehouse, por ejemplo).
4. Reducción de los datos. Se debe obtener una representación reducida de los datos, pero que produce los mismos (o similares) resultados después de su análisis, es decir, una pequeña reducción del conjunto total, pero en calidad es lo mismo.
5. Discretización de los datos. Es un caso especial de la reducción, pero que tiene especial importancia cuando se esta tratando con atributos numéricos.

1. <u>La limpieza de los datos</u>
   a. *Datos nulos*

      Causas:

      La falta de atención por parte de las personas que capturan los datos, una falta de consistencia del sistema que esta capturando los datos de forma automática o semi-automática. También puede ser por que es una variable dependiente y la(s) variable(s) independiente(s) está(n) corrupta(s) de algún modo.

      Soluciones:

      - Se puede ignorar directamente la tupla, aunque no es muy efectivo si el porcentaje de valores nulos por atributo es variable.
      - Rellenar el valor manualmente, tarea tediosa, de dudosa efectividad y muy lento.

- Usar un valor constante para identificar su no existencia (como "desconocido")
- Usar el valor medio de toda la población para rellenar este campo con respecto a los demás registros que están completos.
- Usar el valor medio de la clase a la que pertenece este registro (el más efectivo, probablemente).
- Usar el valor mas probable mediante un árbol de decisión o redes bayesianas.

b. Datos con ruido

Causas:

Estos datos, pueden contener el ruido por un error en la medición, recolección, problemas en la transformación, limitantes de la tecnología y/o inconsistencia en el nombrado de los valores. También requieren limpieza aquellos datos que han sido duplicados o son inconsistentes.

Soluciones:

- Implementar el método de los cubos. Ordenar los datos y dividirlos en cubos de igual longitud. Se suavizan cada cubo por la media, la mediana, la varianza, los límites del cubo, entre otras mediciones estadísticas.
    - Divide el rango en **N** intervalos de igual tamaño.
    - Si **A** y **B** son los valores mínimo y máximo del atributo, la anchura de los intervalos es:

$$W = \frac{(B-A)}{N}$$

    - Es el método más directo.
    - Hace que los outliers dominen.
    - No es aconsejable con distribuciones muy heterogéneas.

- Clustering. Se detectan y eliminan los outliers.
- Combinar el dato ordenador con métodos manuales.
- Regresión. Suaviza el ruido mediante la función obtenida, método parecido al de los mínimos cuadrados.

2. Integración de los datos

Es una combinación de las diversas fuentes, con la finalidad de poder hacer un tratamiento uniforme y medir la compatibilidad de fuentes.

La integración de esquemas consiste en la integración de la metadata de las distintas fuentes.

a. Identificar las entidades: usar la documentación de la base de datos que está por integrarse o del programa que recolecta los datos y así saber cual es la identificación de cada una de sus columnas y variables respectivamente.
b. Detectar y resolver conflictos: para la misma entidad los valores de diferentes fuentes pueden ser tratados diferentes, tener diferentes representaciones, métricas, escalas, entre otras.
c. Manejo de datos redundantes: las redundancias se dan cuando se integran múltiples bases de datos, archivos planos o cualquier otro tipo de almacén de información. Un mismo registro existe más de una vez, una variable tiene más de un nombre o un campo es derivado de otro sin aportar mayor información, entre otros. Se puede detectar por medio de un análisis de correlación. La integración cuidadosa puede ayudar a prevenir / reducir las redundancias e inconsistencias mejorando los resultados.

3. Transformación de los datos

Las acciones que se pueden tomar son:

a. Hacer una *erradicación del ruido*, (hay que recordar que hasta la misma naturaleza de los datos es impura).
b. Hacer una agregación por medio de la construcción de los cubos de datos.
c. Hacer una generalización.
d. Hacer una normalización, (escalar los valores para que caigan en un rango específico).
e. Hacer la construcción de nuevos atributos.

Dentro de las normalizaciones existen de distintos tipos:

- min-max.

$$v' = \frac{v - \min_A}{\max_A - \min_A}(nuevo\_\max_A - nuevo\_\min_A) + nuevo\_\min_A$$

- Z-score.

$$v' = \frac{v - mean_A}{desviacion\_std_A}$$

- Normalización decimal.

$v' = \frac{v}{10^j}$, donde j es el más pequeño entero encontrado por $\max(|v'| < 1$

4. Reducción de datos

Es obtener una representación reducida del conjunto de datos que es mucho más pequeña en volumen pero produce los mismos (muy similares) resultados.

Las estrategias de reducción que existen son:

- Agregación.
- Reducción de dimensiones.
- Discretización.
- Generación de jerarquías de conceptos.
- Compresión de datos.

También se pueden hacer reducciones de atributos, por medio de métodos heurísticos, los cuales son:

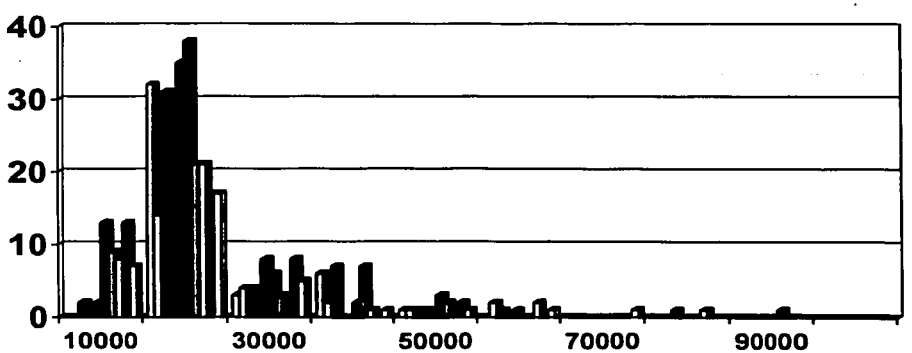
- Selección hacia delante.
- Selección hacia atrás.
- Combinación de estratégias.
- Árboles de decisión.

a. Para la compresión de datos existen muchas opciones, pero normalmente habrá alguna pérdida de información; esto aplica a imágenes, audio o video.

b. Existe la opción de hacer una reducción de dimensiones cuando el Número de estas es muy elevado. Se describe de la siguiente forma:

- Dados N vectores de k-dimensiones, encontrar c <= K vectores ortogonales que se pueden utilizar para representar los datos.
- El conjunto original de datos se reduce a uno de N vectores sobre c componentes principales.
- Cada vector es una combinación lineal de los c vectores de componentes principales (dimensiones reducidas).
- Solo se puede usar con datos numéricos.

Otra forma de reducir dimensiones es por medio de histogramas, que es una técnica muy popular de reducción de datos.



Muestreo de datos

Un muestreo representa la elección de un subconjunto representativo de los datos. Pero un muestreo aleatorio es peligroso porque depende totalmente de la distribución de los datos.

Lo más conveniente es crear un muestreo adaptativo. Existen dos tipos de este muestro adaptativo, que son:

- Muestreo estratificado. Que exista aproximadamente el mismo porcentaje de cada clase de interés que en la población total.
- Muestreo con reemplazo. Esta técnica tiene como característica que se vuelve a colocar aquel dato que fue elegido en la población, para que tenga de nuevo la probabilidad de ser elegido.

5. Discretización de los datos

Reduce el número de valores de un atributo continuo dividiendo el rango del atributo en intervalos. Las etiquetas de los intervalos se pueden usar para reemplazar los valores reales.

Algunos de los algoritmos de clasificación sólo aceptan atributos categóricos. Reduce el tamaño del conjunto de datos.

Algunas de las opciones de discretizaciones para datos numéricos son:

- Intervalos ("cubos")
- Análisis de histogramas.
- Análisis de cluster.
- Discretización basada en la entropía.
    - Dado un conjunto de ejemplos S, si S se divide en dos intervalos S1 y S2 de manera que se minimice la **entropía**.

$$E(S,T) = \frac{|S_1|}{|s|} Ent(S_1) + \frac{|S_2|}{s} Ent(S_2)$$

    - El proceso se aplica de manera recursiva hasta que se encuentre un criterio de finalización.

- Segmentación por particionamiento natural.

LA NATURALEZA DEL MUNDO Y SU IMPACTO EN LOS DATOS

Los datos se exploran para descubrir conocimiento contenido en ellos. La minería de datos se utiliza como una herramienta para descubrir este conocimiento en el conjunto de datos. Una suposición razonable es que el conocimiento descubierto pueda ser aplicado al mundo real.

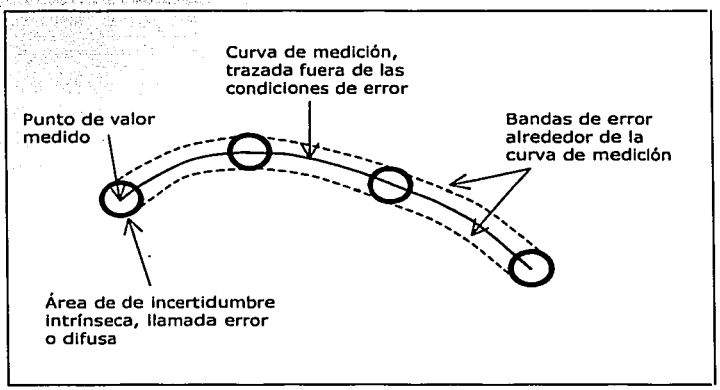
Las suposiciones existentes, afectan a la minería de datos (según el tipo de datos que se están analizando y las salidas que se esperan, las técnicas a utilizar y la herramienta que realizará el proceso de preparación y minería).

Se verán a continuación la naturaleza y las razones de algunos ajustes, alteraciones y reformas que tienen que ser aplicados a los conjuntos de datos para prepararlos para ser minados.

Una tendencia actual, es poder visualizar todas las variables como características de un objeto, es decir, que cada uno de los registros (renglones) de la base de datos se le considera una instancia de objeto, por lo que cada una de las columnas representan las características del mismo. Cada una de sus características son mediciones realizadas.

Es como congelar un momento el mundo, porque ningún valor es constante, considerando un preciso instante como importante, se "toma una fotografía" de esos valores y se conservan.

Al realizar estas mediciones, se pueden presentar problemas de distinta índole, como son los errores de ambiente, que representan una cierta incertidumbre del mundo real; no se puede hacer mucho al respecto ya que así es como esta representado el valor en la naturaleza del problema mismo.



Varios puntos con sus incertidumbres (áreas difusas) y sus curvas de medición así como las de error(bandas)

Es importante en la preparación de los datos que se pueda determinar un error, aunque no compete en esta parte la corrección del mismo, si podría determinarse cual de los componentes es propenso al error o lo trae intrínseco para así poder examinar su forma.

Algunas veces las mediciones están descritas como dos partes que la integran: el valor actual y la distorsión.

La distorsión es a menudo referida como un error. Sin embargo, la distorsión es actualmente una parte integral de las mediciones. Pero utilizar el termino error, tiene connotaciones desafortunadas, como si algo estuviera mal con la medición. Por lo que se asocia que si el medidor fuera aun mejor (o más precavido) se podría eliminar el error. Pero la distorsión es algo mas allá, pueden ser causas que vienen integradas en la misma naturaleza de los datos, y, es algo que se debe considerar como parte de cualquier medición, es algo "normal".

Todas las mediciones se realizan con respecto a una *escala*.

TIPOS DE MEDICIONES

1. Existen las mediciones escalares, habiendo varios tipos y cada una otorga cierta información.

    a. *Mediciones de escala nominal*. Los valores nominales solo son los nombres de las cosas. Es tan solo un poco más que simplemente etiquetar los objetos con fines de identificación. No hay un orden inherente.
    b. *Mediciones de escala categórica*. Las mediciones categóricas agrupan las cosas en formas manejables. Además, no se incluye información para indicar cuan diferentes son cada una de las categorías. Todo lo que se puede decir es que son categorías diferentes. Se denota una diferencia de tipo, pero no esta habilitada una diferencia cuantificada.
    c. *Mediciones de escala ordinal*. Cuando algo más puede ser dicho acerca de una escala de medición, la información adicional se da ordenando las categorías que serán utilizadas para etiquetar las mediciones. La transitividad es una noción razonable de importancia crítica cuando se califican los objetos. Cuando se dice que $A > B$, y $B > C$, entonces $A > C$.
    d. *Medición de escala de intervalos*. Cuando existe información disponible no solo acerca del orden de posición, sino también sus diferencias, se llama intervalo de escala. Sin embargo se debe especificar algo mas, ya que no seria lo mismo considerar un rango de 10°C en el intervalo de -20°C a -10°C, que en el rango de 30°C a 40°C, simplemente para dar el pronóstico del tiempo.
    e. *Mediciones de escala de ratio*. Son variables de tipo cuantitativo. Los números que entrega son adimensionales, por lo que solo se interesa por

entregar resultados manipulables, pero deberán ser interpretados por las personas que posean conocimientos.

2. Las mediciones no escalares, dan origen a variables aun mas complejas llamadas "vectores", con lo que se puede tener información adicional.
   Además de las mediciones, también se deben considerar los tipos de variables que serán empleadas, ya que cada tipo otorga una solución y soporte para un problema específico.
   a. *Variables de un solo valor (constantes).* Este tipo de variables, solo tienen un valor desde el inicio del problema y hasta el final del mismo, esto es muy utilizado, ya que existen datos que nunca cambian de valor durante un problema en específico, como son los días de la semana, pi, centímetros en un metro, etc.
   b. *Variables de dos valores.* Estas son variables que se conocen como dicotómicas, como lo seria el sexo (M o F).
   c. *Variables perdidas o vacías.* Estas variables pueden traer muchos problemas si es que no son tratadas con el debido formato, ya que nunca un valor nulo pude ser promediado, agregado o de cualquier forma tratado. Tienen su origen cuando no se grabo ese valor por el usuario o por la aplicación. El valor vació es un tipo de variable que nunca fue llenada, pero finalmente tendrá que tener una cierta "mascara", con la que se logrará colocar un valor por defecto, el cual puede ser manipulado. Dentro de las bases de datos se trabaja el valor *NULL*, lo cual implica un cierto tratamiento distinto para esta variable.
   d. *Variables binarias.* Son las que solo pueden tomar valores de 0 o 1. Estas normalmente son empleadas como una referencia hacia algo verdadero o falso.
   e. *Variables continuas.* Son aquellas que pueden cambiar de valor dentro de un rango infinito de posibilidades. Requieren cierta precisión. Se debe "discretizar" con respecto a la unidad mas pequeña de expresión, Como lo seria un segundo, un centavo, un milímetro, un gramo, etc. Aunque en el mundo real todo es continuo, para la interpretación humana se debe hacer una discretización.
   f. *Variables categóricas.* Son aquellas que tienen un conjunto de valores perfectamente definidos, tienen un orden natural. Pero cuando se desean mapear al dominio de las variables numéricas se debe introducir una especie de "bandera", que da un calificativo a este valor. Son por naturaleza de tipo carácter.

g. *Variables de rango.* Son variables de tipo categórico y con un orden natural, además de tener una correspondencia al dominio de las variables numéricas, excepto que no tienen un valor exacto, es decir, siempre abarcan un bloque de valores, el cual nunca puede ser precisado.
h. *Variables de intervalo.* Estas son las variables que representan una diferencia entre dos valores. Un tipo de variable de intervalo puede ser desde una fecha y hasta un número, pero un intervalo puede ser fácilmente calculado, más no aritméticamente manipulado.
i. *Variables verdaderamente numéricas.* Estas son aquellas en las que cualquier operación aritmética es posible de realizar. Las variables verdaderamente numéricas son almacenadas en variables numéricas, aunque no todas las variables numéricas son variables verdaderamente numéricas.
j. *Variables de texto, imágenes y audio.* Estas variables no pueden ser minadas directamente, aunque pueden contener mucha información importante para el negocio, lo cual puede ser solucionado al hacer una construcción de variables derivadas, en las cuales se puedan haber separado las características importantes para la minería de datos.
k. *Variables derivadas.* Son aquellas que no fueron medidas o recabadas directamente, sino que tuvieron que recolectarse otras características previas para dar origen a estas, sin embargo, pueden tener gran relevancia para una minería de datos o tener que ser eliminadas para llevar a cabo la minería. Generalmente estas variables surgen por el análisis, combinación o procesamiento de una o varias variables pre-existentes.

El ordenamiento de los datos siempre facilita un minado de los mismos.

También se debe considerar su representación y su accesibilidad para tener una implementación fácil.

La representación de los datos se puede realizar en dos grandes formas, por medio de *variables aisladas* o por medio de *conjuntos de datos*.

Aunque la diferencia pudiera ser obvia, la verdadera diferencia es que de la primer forma, solo se estudian los valores que han podido ser capturados de esas características en particular, con lo cual la relación e interacción entre las variables son de menor o nula importancia. En el segundo caso, lo más importante a estudiar son las relaciones entre los posibles valores encontrados de estas variables en estudio, por lo que los valores mismos están en segundo plano de importancia.

La minería de datos crea modelos que exploran exclusivamente conjuntos de datos.

Borrado de valores. Dependerá totalmente del tipo de problema que se analice, ya que no solo se considera el tipo de variable que es, sino la medición de la que proviene y el estudio que se esta realizando.

Por ejemplo en una compañía de tarjetas de crédito, un caso asilado será el de un fraude cometido por un cuenta habiente, por lo que no deberá ser descartado o eliminado por ser tan diferente de los demás objetos y su comportamiento, ya que se estima que menos del 1% de cuenta habientes son los que realizan fraudes.

La eliminación de un objeto o registro, se da cuando se determine que la información que proporciona no es valida, es decir, no se pueda recuperar o tenga valores tan fuera del rango posible que definitivamente sea preferible eliminarlos.

Los datos vienen en muchas formas, en muchos tipos y en muchos sistemas. Estos siempre vienen sucios, incompletos y algunas veces incomprensibles. Y aun así, esto es la materia prima de la minería de datos.

Las más poderosas técnicas de minería de datos no pueden encontrar patrones interesantes en los datos sin una preparación adecuada y sin un conocimiento sobre ellos y sobre el negocio que los generó.

La tarea incluye la elección de los datos correctos, entender la estructura de los mismos, agregar variables derivadas y trabajar con datos sucios.

¿Qué es un dato sucio?. Cuando se dice que un dato esta sucio, indica que no tiene una calidad en la que se pueda confiar, esto es, que puede estar almacenado de una forma trunca, ambigua o inexistente, por lo que la utilización del mismo se complica, dada la necesidad de hacer un tratamiento para que pueda ser manipulado y así considerado como un dato fidedigno.

PRESENTACIÓN DE LOS DATOS

Los renglones

Un renglón es la unidad de acción y debería ser determinada para entendimiento de cómo serán utilizados los resultados de la minería de datos. La minería de datos sirve para ayudar a los negocios a tomar eventualmente una acción.

Cuando la minería de datos esta centrada en los clientes, cada renglón a menudo corresponde a un cliente.

O en el caso de las aplicaciones web, la unidad podría estar basada en una cookie almacenada en una computadora. Esto difícilmente corresponde a un usuario, porque muchas personas pueden usar una misma computadora o una persona puede ocupar muchas computadoras.

Para una campaña de correo, un renglón podría referirse a un cliente con una dirección de correo válida.

Para una campaña de mercadeo por teléfono, un renglón podría referirse a un cliente con una línea telefónica válida.

Para una campaña por correo electrónico, un renglón puede referirse a un cliente con una dirección de correo electrónica válida.

Los renglones son el nivel de granularidad y la unidad de acción.

Las columnas

Los campos o columnas representan los datos en cada registro. Cada columna contiene valores. El *dominio* de la columna, se refiere a los posibles valores que cada columna puede almacenar.

En una gráfica el eje vertical representa el número de registros para cada valor y el eje horizontal son los valores de cada columna.

La distribución de los valores provee una muy importante información interna. Los métodos estadísticos están muy ligados con las distribuciones, afortunadamente, los algoritmos de la minería de datos son un poco menos sensitivos a la distribución de los datos.

a. Columnas con un solo valor, son llamadas *columnas uni-valuadas*, y no contienen información para distinguir entre distintos renglones, porque contienen el mismo valor para todos los renglones. Deben ser ignoradas para fines de la minería de datos.
   Otra forma de que se obtengan columnas uni-valuadas, es cuando se está estudiando una parte de todos los registros de la base de datos, por lo que todo el segmento coincide con un mismo valor en ese campo.
b. Columnas con casi un solo valor, son aquellas, que en la mayoría de los renglones en estudio, contienen el mismo valor, por lo que es casi considerado como insignificante la porción de registros que no pertenecen a ese segmento. Pero ¿cuándo es que se deben ignorar estas columnas para la minería de datos? Cuando cerca del 97% (±2%) contengan el mismo valor y la población restante sea de irrelevancia para este estudio.
   Cuando se están buscando segmentos de clientes se pueden ignorar estos registros, pero cuando se están buscando posibles fraudes, estos serán los registros en los que se centrará la atención.
c. Columnas con valores únicos, estas contienen algunas veces información muy útil. Ya que estas son el nombre de la persona, sus identificaciones, dirección, número telefónico, entre otros. Normalmente el uso de estas columnas es para extraer columnas con valores derivados.
d. Columnas sinónimas del objetivo, estas son aquellas que se derivan del objetivo que se esta buscando, lo cual puede influir en el resultado y por lo tanto no dejar que sea natural la elección del mismo a partir de las columnas que no son derivadas del objetivo, deben ignorarse.

Rol de las columnas en la minería de datos

Diferentes columnas juegan distintos roles en la minería de datos. Los fundamentales son:

- Columnas de entrada. Usadas como entradas del modelo.
- Columnas objetivo. Usadas solo cuando se construyen modelos predictivos.

- Columnas ignoradas. Columnas que no son de importancia para ese estudio de minería de datos.
- Columnas identificadas. Usadas únicamente para identificar datos, también se ignoran para propósitos de la minería de datos.
- Columnas con peso. Se les aplica un cierto "peso" (importancia), lo cual es una forma de crear registros con mayor o menor importancia en ciertos campos.
- Columnas con costo. Se especifica un costo para cada columna, con lo cual se puede marcar la dificultad para la obtención de este campo en específico.

DATOS PARA LA MINERÍA DE DATOS

Los datos para la minería de datos requieren lo siguiente:

- Todos los datos deben estar en una sola tabla o vista dentro de la misma base de datos.
- Cada renglón debe corresponder a un registro (ejemplo) que sea relevante para el negocio.
- Las columnas con un solo valor deben ser ignoradas.
- Las columnas con diferentes valores para cada uno de los registros también deben ser ignoradas. Como lo son el id o una numeración.
- Para un modelado predictivo, las columnas objetivo deben ser identificadas y las derivadas deben ser eliminadas.

El más grande reto en la preparación de datos es transformar los datos a un formato en el que puedan ser aplicados los algoritmos de la minería de datos.

Los datos provienen directamente de la organización, desde una variedad de sistemas. Algunos de ellos son usados para operaciones del negocio. Otros para simplemente hacer reportes y otros más para obtener una inteligencia.

A menudo los datos son almacenados en bases de datos relacionales como Oracle, Informix, Sybase, DB2 o SQL Server.

Los datos almacenados por estos sistemas son directamente del punto de contacto con el cliente y otorgan el mayor potencial para el entendimiento del negocio.

Los sistemas operacionales definen los datos potencialmente disponibles para la minería de datos.

Cuando estos sistemas no recolectan datos manipulables, el negocio debe gastar más en estos sistemas y tiene fuentes de datos incompletas.

Los datos por si mismos, tienden a estar sucios.

También, los datos de los sistemas operacionales muestran las reglas del negocio.

Otra posible fuente de datos para una minería de datos, es un Data Mart u OLAP; estas son dos fuentes de datos que no son de importancia para este trabajo, sin embargo se proporcionará una breve descripción.

Un Data Mart, esta enfocado a los datos de un departamento en específico, por lo que el volumen de datos que puede manipular no es muy grande, sin embargo, es mayor al de una base de datos relacional y menor al de un Data Warehouse.

OLAP, son las siglas de On-Line Analytic Processing, que es Procesamiento analítico en línea, es una forma distinta de hacer una depuración de datos, a partir de ciertas preguntas interesantes para el negocio, con lo que se logra obtener ciertos datos que pueden dirigir el rumbo de la investigación sobre la información obtenida. Son sistemas especializados para obtener reportes y análisis dirigido.

Algunas consideraciones para la preparación de los datos.

Determinar qué tipo de datos son los que se utilizaran, para lo cual, se deben recordar las siguientes ideas básicas:

- La gente que se ha entrevistado, por algún medio (web, encuestas, e-mail, etc.) no siempre representa la población entera del negocio, por lo que se debe tener cuidado al dejar fuera a aquella población que puede representar un fuerte espacio de clientes de la población.
- La gente que se ha entrevistado no siempre dice la verdad, o puede ser que se equivoque al escribir sus respuestas, por lo que también se debe considerar una posible línea de "ruido" al recabar los datos.
- Los reportes de encuestas anteriores no serán válidos por mucho tiempo, sobre todo, cuando se trata de un ambiente que sea rápidamente cambiante. Por lo que se deben mantener lo más actualizadas posible.
- Se debe considerar la posibilidad de que los datos almacenados en la base de datos sean incompletos, por lo que no pueden ser tomados con toda la libertad.

Existen fuentes de datos externas; las cuales pueden variar enormemente, desde el formato en el que trabajan los datos y hasta el punto en el que es concebida la importancia de cada campo o su manipulación, por lo que estas fuentes externas también implican trabajo para uniformar los datos con datos locales.

Estas fuentes de datos pueden ser de varios tipos, un buró de crédito, los índices de divisas, índices de inflación, cotizaciones, encuestas realizadas por otras empresas, etc.

Una buena opción es seleccionar fuentes de datos externas, cuando no son suficientes los datos con los que se cuenta, por lo que un estudio más a fondo otorgará un aspecto aun más formal a la minería de datos. Pero ¿Cómo poder elegir una fuente de datos externa?. Esto será determinado por las necesidades de investigación que se estén realizando. Si se requiere hacer una investigación de mercado para un deporte en específico, pues se deberá recurrir a las empresas de las revistas especializadas de este deporte y obtener los suscriptores de la misma.

A menudo, la cantidad de datos almacenados y disponibles es tan grande como la necesaria para realizar la minería de datos. Por otra parte, siempre es importante conservar y trabajar con los datos detallados, ya que proveen características importantes del negocio.

Otra consideración para el volumen de los datos, es el tamaño de la población. Una buena idea es tener miles o cientos de miles de registros disponibles para una minería de datos. Trabajar con un 10% de la población es razonable. Si no se puede reducir el número de registros, se debe reducir el número de datos (columnas) por registro a analizar.

Variables derivadas

Un tipo de variables importante en la minería son las llamadas *variables derivadas*. Estas se obtienen por medio de operaciones entre otras columnas de los registros que fueron obtenidas previamente.

Una variable derivada tiene utilidad cuando representa comportamientos ocultos de los datos. Las variables derivadas son creadas de forma natural, en la preparación de los datos.

Los procesos de agregación crean variables derivadas. Hablando de bases de datos, una "vista" es una tabla temporal con una cierta combinación de datos que se crean como variables derivadas.

El principal objetivo de las variables derivadas es encontrar información utilizable; son una buena forma de incorporar nuevas ideas a los datos.

Variables extrapoladas

Otras variables importantes, dependiendo del estudio realizado, son las llamadas "extrapoladas", es decir, aquellas que están fuera del rango normal de valores y que su naturaleza puede provocar la pregunta: ¿Qué fue lo que causo esos valores?.

Existen cinco formas de trabajar con estas variables:

1. Hacer nada. Algunos algoritmos tienen ciertas consideraciones importantes cuando están presentes estos valores, por ejemplo, en los árboles de decisión representan la creación de nuevas clases, pero en las redes neuronales pueden llegar a causar disturbios.
2. Filtrar los renglones que los contienen. Esta puede ser una mala idea, sin embargo puede ser que para las necesidades de nuestro estudio no puedan existir esos valores y simplemente fueron obtenidos a base de errores.
3. Ignorar las columnas. Esta es la acción más extrema. La columna puede ser reemplazada con otros datos referentes a esa misma columna, es decir, obtener su dato a partir de una variable derivada.
4. Reemplazar los valores extrapolados. Esta es una aproximación muy común. El reemplazo puede ser por el valor "null" sí la herramienta de minería datos

puede manipular estos valores. O puede ser "0", o el valor promedio, o un valor máximo / mínimo, o algún otro valor apropiado.

5. Valores dentro de rangos. El ejemplo es colocar los valores bajo, medio y alto.

Estas variables tiene repercusiones en el estudio que se este realizando; si se está investigando posibles fraudes será de vital importancia considerarlas, en cambio, si se están analizando rendimientos de motores, no sería lógico poder considerar aquellos rendimientos superiores al 100%.

Muchas columnas de la base de datos contienen más información de la que se puede apreciar a simple vista, esto es algo muy claro en las columnas con fechas celebres que son: año nuevo, navidad, día de las madres, día del maestro, día del niño, etc. Esto se debe a que el mercado no se comporta de igual forma en un día que es conocido como celebre a un día que no lo es.

Por lo que se deberá de conocer y tomar como referencia si es que esa fecha, día de la semana, o cualquier otra consideración extra puede explicar cierto comportamiento peculiar.

Una formula para poder saber qué día de la semana es, es la siguiente:

$$((fecha\_actual - 01/01/1999) + 4)\%7$$

con lo cual se obtendrán valores entre 0 y 6, siendo el 0 lunes y el 6 domingo.

SERIES DE TIEMPO

Las *Series de Tiempo* representan situaciones que ocurren en ciertos intervalos de tiempo. Una de estas series de tiempo es que cada quincena es depositado un monto en las cuentas bancarias de los trabajadores que reciben su pago de esta forma.

Otro ejemplo es en el caso de las compras de fin de año, como son regalos de intercambio, juguetes para los niños, comida como pavo, el bacalao, etc. Los pagos que se deben hacer mes con mes en el caso de las hipotecas, pagos de impuestos, balances de los periodos del año y sus comparaciones con los años anteriores.

Es muy común que la vida del ser humano este regida por variables de series de tiempo. Los aspectos de interés en series de tiempo son:

- Valores totales o promedios de estas variables.
- Taza de crecimiento de estas variables.
- Número de valores que han excedido un umbral.
- Varianza de las series de tiempo, lo cual da una medición de cuantas veces cambio ese valor.

Las series de tiempo ofrecen muchas oportunidades de agregar variables derivadas.

Capítulo IV

ANÁLISIS DE LAS TÉCNICAS DE LA MINERÍA DE DATOS

Las técnicas de la minería de datos son métodos de aprendizaje utilizados para desarrollar un modelo predictivo.

Cada técnica tiene varios tipos de algoritmos, los cuales pueden tener semejanzas.

La selección de una técnica específica, dependerá del tipo de problema en estudio y de los objetivos de modelado que se estén persiguiendo.

Cada una de estas técnicas tienen una serie de particularidades que las hacen especiales para ser aplicadas en un cierto conjunto de datos o dadas las necesidades que se estén persiguiendo sobre la obtención de resultados por parte de la minería de datos.

Las siguientes técnicas son implementadas en casi cualquier software comercial, por lo que cada productor ha realizado ciertas modificaciones según sus intereses, por lo que estas se detallarán en su forma más común, sin ningún tipo de optimización, ya que así pueden ser implementadas (con ciertas modificaciones específicas según el tipo de problema) al tipo de datos a analizar.

Muchas de estas técnicas se basan en métodos estadísticos, ya que esta ciencia representa una muy buena herramienta cuando se desea hacer una serie de modelos.

Aunque como se verá, no solo se utiliza la estadística.

Algo muy importante que se debe considerar cuando se esta realizando la elección de la técnica para implementar la minería de datos, es que se debe contar con gente que sepa el estado y comportamiento del negocio, ya que si no se cuenta con la gente experta, se pueden hacer elecciones que no sean las óptimas.

El sentido común y el razonamiento de las personas encargadas del análisis de la Tecnología de la Información se deberá complementar con los expertos del negocio. Por lo que se debe concluir que las técnicas no resolverán el problema, solo ayudarán a entender e interpretar de una mejor forma aquellos patrones ocultos en los datos históricos.

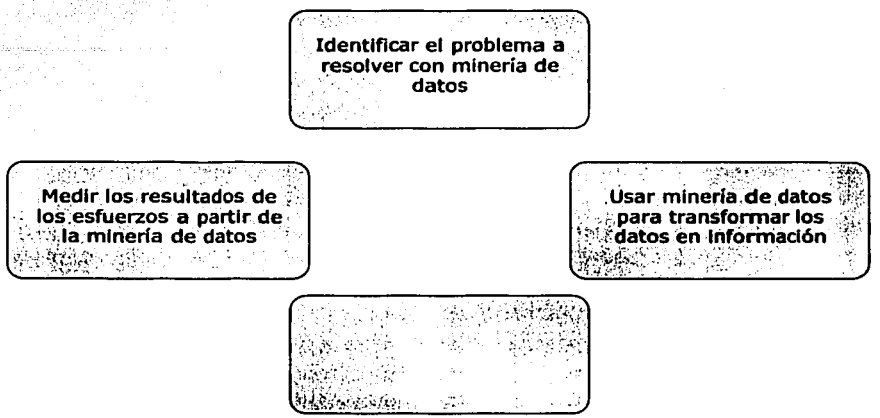
Algunas técnicas son complemento de otras, es decir, para obtener un resultado adecuado por parte de la minería de datos, se requiere la conjuncion de algunas de estas técnicas, siendo algunas utilizadas para la construccion de los modelos y otras para la depuracion de los resultados arrojados por los modelos mismos.

Dadas las características de cada técnica, todas arrojan resultados diferentes (ya sea en su totalidad o de forma parcial), para lo cual es conveniente saber que tipo de resultados se están esperando para así poder elegir cual de las técnicas ofrece aquellos que sean más aproximados a las necesidades del problema en solución.

Finalmente, en la eleccion de la técnica a utilizar, se deben considerar los aspectos de cómputo requerido y tiempo en el que los resultados serán arrojados. Estos dos puntos son muy importantes, ya que existen técnicas que requieren de un conjunto de datos para entrenar los modelos, para depurarlos y finalmente poder tener una respuesta adecuada, lo cual implica que se necesitaran bases de datos de prueba, equipos de cómputo que solo

serviran para las pruebas del modelo, siendo esto un consumo de recursos (cómputo, tiempo, personal, luz, etc.) que algunas ocasiones no se pueden cubrir. Afortunadamente no todas las técnicas requiere de este tipo de prerequisitos.

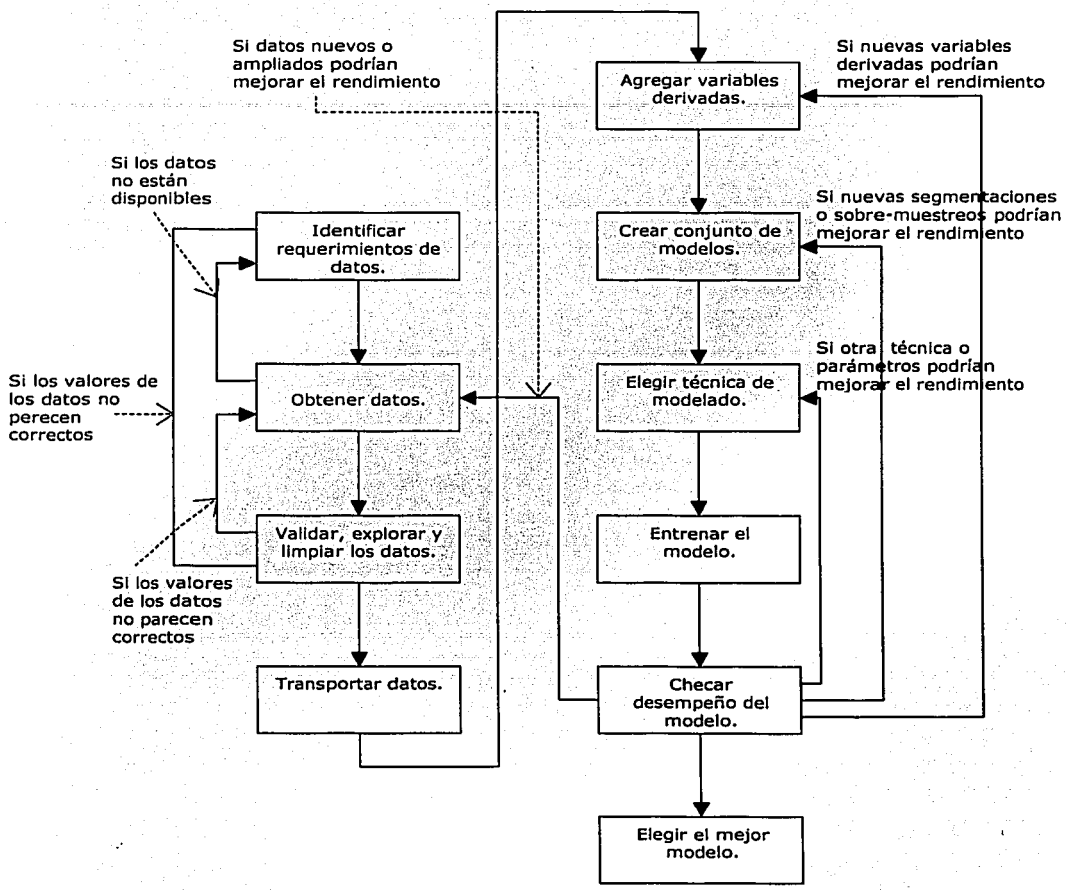
El ciclo de vida de la minería de datos se puede definir de la siguiente forma:



Identificar el problema a resolver con minería de datos. Una *parte indispensable de cualquier proyecto de minería de datos es hablar con la gente que entiende el negocio.*
Usar minería de datos para transformar los datos en información. El corazón de la minería de datos es transformar datos en resultados que lleven a acciones. El primer paso en el proceso de modelado es identificar y obtener los datos correctos.

A menudo se pueden utilizar los datos disponibles, razonablemente limpios y accesibles. Se requieren los datos tan completos como sea posible.

Una de las cosas cruciales de la minería de datos es saber que es un proceso iterativo como lo muestra la siguiente figura.



Construir un modelo de minería de datos es un proceso iterativo.

ANÁLISIS DE LA CANASTA DE MERCADO (MARKET BASKET ANALYSIS)

Esta técnica surge de analizar el comportamiento de un cliente cuando esta realizando sus compras, indica esta técnica que productos tienden a venderse juntos y cuales son mejores para estar en promoción.

No todos los clientes son iguales ni tienen las mismas necesidades ni posibilidades de compra, sin embargo los resultados arrojados por un análisis de la canasta de mercado son muy cercanos a la realidad. Finalmente esta técnica sirve para determinar nuevas reglas de promociones en ciertos productos que conllevan a la venta de otros productos.

Usa la información acerca de "qué" es lo que se compra, "quienes" son quien lo compran, "cuando" es que lo compran. Una vez contestadas estas preguntas puede determinarse el "por qué" hacen esa compra.

Otras aplicaciones en que se emplea esta técnica son:

- Compras con tarjeta de crédito.
- Servicios opcionales de telecomunicaciones.
- Servicios extras otorgados por los bancos.
- Reclamaciones inusuales en agencias de seguros.
- Determinación de indicaciones y complicaciones en pacientes (historia médica), para determinar un tratamiento.
- Fallas en equipo de diversos tipos (computadoras, herramientas, etc.).

Esta es una técnica que puede ser orientada en una minería de datos dirigida o no dirigida. Las técnicas del análisis de la canasta de mercado son extraídas de la probabilidad y la estadística.

Los modelos que se construyen dan la probabilidad de las ventas de diferentes productos y se pueden expresar los resultados como reglas de asociación. Esta técnica se usa como el punto inicial del tratamiento de la información, cuando no se sabe que patrones se deben buscar y hay datos disponibles.

Los tipos de reglas de asociación que se pueden obtener son tres tipos.

1. Reglas útiles de alta calidad. Estas pueden definir acciones y así obtener beneficios. Fácil de entender y visualizar.
2. Resultados triviales, conocidos por todos en el negocio. Los resultados de esta regla, pueden estar midiendo el éxito de las acciones anteriores (por ejemplo, campañas de mercadeo anteriores).
3. Resultados inexplicables que parecen no tener explicación y no sugieren nada. Estos pueden no ser importantes, o ser coincidencias. En ciertos casos pueden representar el inicio de una investigación para la explicación de estos patrones encontrados.

APLICACIÓN DE LA TÉCNICA DE ANÁLISIS DE CANASTA DE MERCADO

Esta técnica inicia con el análisis de las transacciones que contienen productos o servicios ofrecidos y alguna información rudimentaria sobre las transacciones. Para hacer estas asignaciones, es necesario hacer un tipo de matriz, que relacione cada producto que, según los registros existentes, se haya vendido junto con otro.

Por ejemplo, se analizan a cinco clientes que realizaron sus compras, pero solo se considera qué productos compraron, los cuales son representados en una tabla:

Cliente	Productos
1	Jugo, refresco
2	Leche, Jugo, limpiador
3	Jugo, detergente
4	Jugo, detergente, refresco
5	Limpiador, refresco.

Tabla de Clientes-Productos

Después de tener esta tabla, se debe hacer una que nos indique la co-ocurrencia de los productos, esto es, que se muestren las ventas coincidentes entre los productos (las combinaciones de venta entre todos los productos mencionados en la tabla anterior).

Los números que se representan en la intersección de un producto consigo mismo, indican el número total de veces que fue vendido ese producto.

	Jugo	Limpiador	Leche	Refresco	Detergente
Jugo	4	1	1	**2**	**2**
Limpiador	1	2	1	1	0
Leche	1	1	1	0	0
Refresco	**2**	1	0	3	1
Detergente	**2**	0	0	1	2

Tabla de Co-ocurrencia de productos

La intersección de Jugo (fila) y Jugo (columna) nos indica que fue vendido 4 veces el Jugo. También nos indica cosas notorias:

- El Jugo y el refresco son los productos que más se venden juntos (ocurriendo esto 2 veces).
- El detergente nunca se vendió junto con el limpiador ni la leche.
- La leche nunca se vendió junto con refresco o detergente.

También se puede trabajar con cubos (análisis en tres dimensiones), o más dimensiones si se requiere, pero se debe tener en cuenta que el análisis existente deberá

ser fácilmente entendible y con más de tres dimensiones resulta más complejo entenderlo. Este análisis con n productos, tiene un número de posibles combinaciones de n^n.

En esa matriz creada para el análisis, cada columna representa un producto a analizar y cada renglón es un cliente o registro de una transacción de venta; dependiendo si lo que se esta buscando son las ventas conjuntas de productos o el comportamiento de un cliente.

Las reglas generadas en el análisis de la canasta de mercado actúan como puntos de partida para futuras pruebas de hipótesis.

Los pasos que se deben seguir para la realización de un análisis de canasta de mercado son los siguientes:

- Elegir los elementos correctos a analizar.
- Generar reglas para descifrar las co-ocurrencias encontradas.
- Superar los límites prácticos impuestos por miles de registros de transacciones.

Los datos correctos para usar en esta técnica son los que típicamente se detallan en las transacciones registradas en el punto de venta. Estos pueden cambiar en cualquier momento. Elegir el nivel correcto de detalle es una consideración crítica para este análisis.

El detalle se puede basar en una agrupación jerárquica en la que se pueden separar productos, que vistos sin detalle era uno solo.

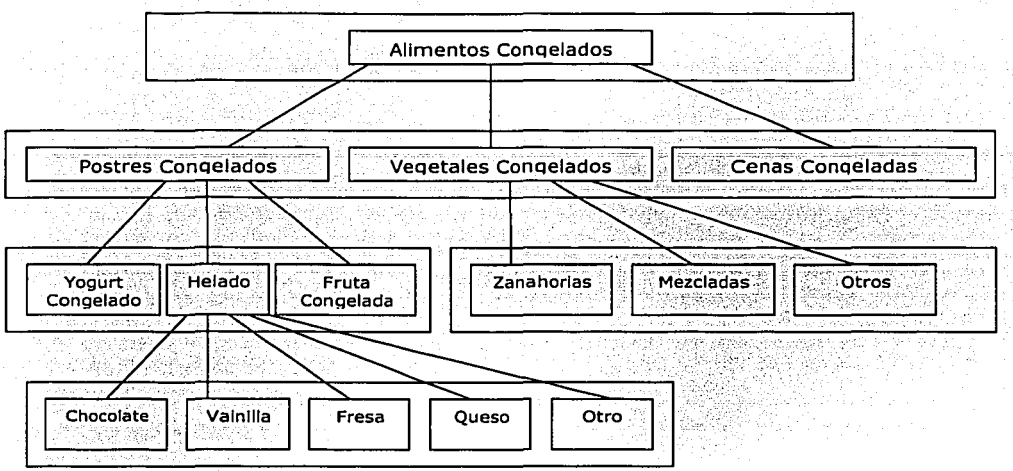
Al usar productos muy detallados, se consigue un análisis muy pequeño y con pocas transacciones.

Un análisis sin detalle, es normalmente usado para ver relaciones entre departamentos.

No es necesario que todos los productos sean llevados al mismo nivel de abstracción, se recomienda iniciar con un nivel de abstracción alto, y al obtener los resultados, se puede eliminar abstracción e ir trabajando con más detalle en cada ciclo que se realice.

Un ejemplo de esto, seria detallar alimentos congelados, teniendo una estructura en la que se pueden englobar varios productos:

Taxonomía



Presentaciones existentes, ediciones especiales por temporada, promociones, entre otras variantes de un mismo producto.

El análisis de la canasta de mercado nos ayuda a describir las campañas de mercadeo anteriores, comportamiento de los clientes según las condiciones a las que esta sometido y posibles líneas de investigación sobre algún patrón encontrado.

La **taxonomía** sirve para definir el nivel de detalle con el que se trabaja con cada producto, si es que se trabajan con grandes volúmenes de información.

Si se generaliza demasiado se puede hablar de relación entre aspectos más grandes relacionados con las ventas (como departamentos de la tienda, temporadas de ventas en el año o comparación de ventas en distintos puntos geográficos); por el contrario, con un nivel muy detallado, solo se podrá hablar de ciertos productos y sus configuraciones (formas en las que es preparado un producto y como se vende más). Como lo muestra la figura anterior.

En un **análisis diferencial** (este es el que se logra con un nivel de abstracción muy alto) se pueden comparar diferentes tiendas, clientes en diferentes grupos demográficos, ventas en diferentes días de la semana, estaciones del año, etc.

Un elemento que se utiliza para esta técnica, es la construcción de **productos virtuales**, para tomar ventaja de la información que va con la taxonomía. Son una mezcla de los límites de ciertos productos.

Son nombres (etiquetas) que engloban productos (por ejemplo lácteos o ferretería), esto se logra porque su grado de detalle es pobre, siendo una especie de generalización.

Pueden incluir información sobre transacciones (por ejemplo forma de pago), sin embargo no es muy recomendable aglutinar tanta información en los productos virtuales. Solo se deben incluir productos virtuales cuando se tiene una idea de cómo se puede obtener información procesable si se obtiene un buen modelo con reglas de asociación de alta confianza. Su uso principal es poder representar comportamientos estacionales, ventas de un departamento, ventas de una marca, etc. Pueden ser causas de reglas redundantes. Hay que tener cuidado de no definir el producto virtual en términos de un solo producto.

Analizando una transacción, se puede dividir en *ventas anónimas* (realizadas en efectivo) lo cual, implica que no se puede dar seguimiento del cliente, ni conocer su perfil de compra como si utilizara una tarjeta de crédito o debito, siendo estas últimas *ventas con "firma"*. Para el caso de las ventas es más enriquecedor que el cliente utilice sus tarjetas de crédito o débito para hacer un posible seguimiento del mismo y así saber que productos son propensos a comprar, por medio de la construcción de reglas.

Una regla es una condición que tiene un resultado cuando ésta se cumple, y en algunas ocasiones también tiene resultado cuando no se cumple. Tiene la estructura de: *Sí condición, entonces resultado.* Esto proporciona una serie de acciones a tomar según sea el caso en el que mejor se ajusten los valores a procesar.

Para lo cual se debe tener el cálculo previo de estas probabilidades. La interpretación de los resultados es que, si es mayor a 1, la regla resultante es buena en su predicción, cuando esta es menor a 1, esta mal.

GENERACIÓN DE REGLAS DE ASOCIACIÓN

Una regla tiene dos partes, una condición y un resultado, y usualmente representada como:

SI *condición* ENTONCES *resultado.*

Este es un proceso de varios pasos:

1. Generar una matriz de co-ocurrencias de elementos simples.
2. Generar una matriz de co-ocurrencias de dos elementos. Esta se usa para encontrar reglas con dos elementos.
3. Generar una matriz de co-ocurrencias de tres elementos. Esta se usa para encontrar reglas con tres elementos.
4. Y repetir.

En la búsqueda de estos elementos, se debe elegir aquellos que cumplan con lo planteado o que enriquezcan las reglas, pero desafortunadamente no todos los productos que aparecen son utilizables, por lo que se debe hacer una serie de reglas que aseguren que los elementos encontrados sean los necesarios, ni más ni menos.

Esto se llama *soporte mínimo de corte*, en donde el corte se refiere a eliminar todos aquellos elementos que no pertenecen a las reglas en estudio. Existen dos formas de lograr esto. La primera es eliminando los elementos que no se requieren por medio de ciertas consideraciones y la segunda es usando la taxonomía para generalizar los elementos que intervienen, reduciendo así el número de productos.

Para mostrar la importancia del número de elementos que se deben considerar para la obtención de reglas de asociación, se toma el ejemplo de un súper mercado que genera aproximadamente en un año cerca de 10 millones de transacciones. Si se considera que un negocio de comida rápida con 100 productos, en una presentación cada uno, vendiéndolos en paquetes de tres, se puede tener una combinación de 3 presentaciones con 100 productos, es igual a 161,700 opciones. Si se aumentan el número de presentaciones (en tres tamaños, chico mediano y grande de cada producto), se tiene una combinación de 9 presentaciones con 100 productos, es igual a 1,902,231,808,405 opciones.

Esto requiere mayor tiempo de procesamiento, mayor inversión en equipo de cómputo y con utilizar la taxonomía de los productos se puede reducir esto drásticamente.

Una característica del análisis de la canasta de mercado es que estudia las ventas simultáneas de varios productos, por lo que se habla de un instante, pero que es lo que sucede con las compras subsecuentes de los clientes. Para esto se requiere poder reconocer al cliente en el transcurso del tiempo, siendo esto basado en las ventas con firma (con tarjetas de crédito o debito). Esto se llama *análisis secuencial de series en el tiempo.* Solamente este análisis es posible cuando se tiene una forma de seguir el comportamiento de compra del cliente y cuando se tiene una forma de referenciar en el tiempo las transacciones realizadas por el cliente. Con lo que se puede lograr predecir después de un cierto número de veces que se repitiera un ciclo de acciones realizadas por el cliente, cual seria su próximo movimiento, compra.

¿Cómo obtener el nivel de confianza de las reglas?. Su nivel de confianza es el porcentaje de exactitud con el que una regla proporciona sus resultados. Obviamente este resultado depende de la predominancia de un cierto elemento en los registros a analizar. Se desarrollará un ejemplo sobre tres elementos y se verificarán todos estos conceptos.

Si a y b, ENTONCES c.
Si a y c, ENTONCES b.
Si b y c, ENTONCES a.

Combinación	Probabilidad
a	45%
b	42.5%
c	40%
a y b	25%
a y c	20%
b y c	15%
a y b y c	5%

Probabilidades de las posibles combinaciones de los elementos

Sin embargo, si se observa bien, ninguna de las combinaciones otorga un porcentaje de confianza mayor que al de los elementos por sí solos. Lo cual sugiere otra forma de medición, llamada Mejora (improvement).

La mejora (improvement) es una de las formas de medir la efectividad de una regla, se hace a partir de las probabilidades de cada uno de los elementos que intervienen en la regla y la probabilidad de que todos los elementos ocurran al mismo tiempo.

El improvement indica de mejor forma cual de las reglas es mejor para predecir el resultado que se obtendrá comúnmente en primer lugar:

$$improvement = \frac{p(condicion\_y\_resultado)}{p(condicion) \cdot p(resultado)}$$

El resultado obtenido por *improvement* deberá ser mayor que 1.00 para que pueda ser considerado como bueno, por debajo del mismo, es un resultado intrascendente.

Con esta nueva medición, los resultados de la tabla anterior se realizan mucho más completos, quedando de la siguiente forma:

Regla	P(condición)	P(condición_y_resultado	Confianza	Soporte	Improvement
SI a y b ENTONCES c	25%	5%	0.20	5%	0.5
SI a y c ENTONCES b	20%	5%	0.25	5%	0.59
SI b y c ENTONCES a	15%	5%	0.33	5%	0.74
SI a ENTONCES b	45%	25%	0.59	25%	1.31

Probabilidades de las reglas generadas y sus combinaciones

En algunas circunstancias especiales, se obtienen mejores resultados al negar la salida de la regla, ya que se obtiene el complemento de la probabilidad de la misma. Por ejemplo:

SI a y b, ENTONCES c; con 33.3% de confianza.

Si se realiza el siguiente cambio:

SI a y b, ENTONCES NO c; con 66.6% de confianza.

Siendo esta una de las posibles soluciones, aunque no es la única y como ya se vio, resultaría mucho mejor usar el resultado arrojado por *improvement.*

Sus ventajas. Esta técnica arroja resultados claros y entendibles, los cuales están listos para ser expresados en términos comunes para que sean entendidos por cualquier persona relacionada con el tema que esta siendo tratado o para poderlos llevar fácilmente a una declaración de SQL.

Es potente en la minería de datos no dirigida, ya que cuando no se tienen premisas para iniciar un análisis de los datos, con esta técnica se tiene un buen inicio para entender los datos existentes.

No requiere que se realicen transformaciones a los datos, los puede procesar naturalmente, logrando con esto no tener perdida alguna de información.

No requiere cálculos muy sofisticados, por lo que desde una hoja de cálculo se puede hacer este análisis, comúnmente utilizado como precursor de las técnicas de algoritmos genéticos o redes neuronales. Se utiliza principalmente, cuando las ventas son la base del análisis y cuando se requiere realizar una minería de datos no dirigida.

Sus desventajas. Requiere que el poder de cómputo crezca exponencialmente si la cantidad de datos crece también, por lo que las reglas generadas son difíciles de manipular; se tienen ciertos elementos para evitar que los productos crezcan ilimitadamente (como los productos virtuales o la taxonomía).

Tiene un límite en el nivel de atributos de los productos que puede manejar. Teniendo uno de los mayores problemas al realizar esta técnica es cuando se debe elegir el número correcto de elementos a trabajar. Con los niveles de taxonomía y elementos virtuales, los productos que son muy caros y que por eso raramente son vendidos, llegan a ser omitidos en los resultados finales, para estos casos se deben crear elemento virtuales específicos para que no sean excluidos de las reglas finales.

RAZONAMIENTO BASADO EN LA MEMORIA (MEMORY-BASED REASONING, MBR)

Esta es una técnica que se asemeja a la forma en la que las personas toman decisiones, ya que *identifica* situaciones similares basadas en experiencias pasadas, *aplicando* información del problema actual y así obteniendo una solución que previamente fue probada.

También utiliza información de registros vecinos para realizar una clasificación y predicción. Una de las ventajas de esta técnica es que no requieren los datos de casi ninguna preparación.

Aquí existen dos funciones que son sumamente importantes:

- La función de distancia, que asigna una distancia entre dos registros.
- La función de combinación, que combina los resultados de los vecinos para llegar a la respuesta.

Algunas áreas de aplicación de esta técnica son:

- Detección de fraudes. Ya que los nuevos casos de fraudes pueden ser parecidos a casos anteriores, siendo esto el resultado del inicio de una investigación.
- Predicción de respuestas de clientes. Se pueden encontrar ciertas características comunes en los clientes que ya han respondido con los clientes potenciales.
- Tratamientos médicos. Los tratamientos a los que son sometidos los pacientes, hacen que para ciertos padecimientos comunes, se pueda encontrar un patrón que indique que tipo de tratamiento médico se puede sugerir a ese paciente.
- Clasificación de respuestas. Con el MBR se puede lograr un análisis de respuestas en texto, asignándoles un código y así obteniendo un control sobre las mismas.

Puede ser aplicada a casi cualquier tipo de dato. Dadas sus características, se adapta rápidamente a cualquier conjunto de datos agregado, sin importar que no tenga mucho tiempo de entrenamiento o un formato especial de tratamiento.

Los precios a pagar por estos beneficios son que puede ser una técnica muy lenta cuando el conjunto de datos históricos es muy grande, en comparación con una red neuronal entrenada o un árbol de decisiones ya construido.

¿CÓMO TRABAJA EL MBR?

Teniendo un conjunto de datos (una base de datos relacional, o un data warehouse) se aplica esta técnica para obtener un modelo de datos existentes.

Fase de reconocimiento, recorrerá cada uno de los registros para hacer una clasificación dadas las características de cada uno. Esto se realiza sin derivar ninguna regla de los datos originales y sin ningún ciclo de entrenamiento.

Fase de predicción, que se aplica a los nuevos registros a partir de las funciones de distancia y combinación construidas en la fase de reconocimiento, y, el conocimiento de los vecinos cercanos.

Pasos para la aplicación del MBR:

1. Elección correcta del conjunto apropiado de registros históricos.
2. Determinar la forma más eficiente de representar los registros históricos.
3. Determinar la función de distancia y la función de combinación y el número de vecinos.

1. Elección correcta del conjunto apropiado de registros históricos. El conjunto de datos de entrenamiento debe proporcionar una cobertura adecuada, ya que es un subconjunto del total de los datos; a partir de un método de elección aleatorio los registros con mayor presencia, son lo que cuentan con una mayor probabilidad de dominar en este subconjunto y es sesgado el estudio de los vecinos cercanos en la construcción de las funciones de distancia y combinación. Se debe tener un conjunto de registros donde cada una de las distintas categorías tiene aproximadamente el mismo número de registros.

2. Determinar la forma más eficiente para representar los registros históricos. El éxito del MBR al realizar predicciones, depende grandemente de la representación del conjunto de datos de entrenamiento. Esto se puede hacer por medio de una base de datos especializada, pero si una base de datos relacional se deberá recorrer todos los registros, además de requerir un procedimiento almacenado (el cual serviría para realizar de forma automática algunas operaciones); la consecuencia es que cuando el número de registros aumenta, también aumenta el tiempo de búsqueda, ordenamiento y calculo de los vecinos cercanos.

Otra forma de hacer esto aún mejor, es reduciendo el número de registros históricos. Esto se logra al tener una serie de clusters donde están bien definidos (sin traslapes entre estos, ya que de ser así, el MBR produce resultados muy pobres en información procesable), tomando solo el centro de estos clusters (los cuales representan una categoría dentro de los datos). Esto reduce considerablemente el número de registros.

3. Determinar la función de distancia y la función de combinación y número de vecinos. Estos tres parámetros son clave para un buen resultado del MBR. Primeramente se detallará la función de distancia, por ser esta la que algunas veces se toma como referencia para determinar la función de combinación.

Función de distancia. Esta función se denota por *d(A,B)*, donde A y B son dos registros de los cuales se desea tener la distancia entre ellos.

Las cuatro propiedades que debe cumplir esta función son:

1. Buena definición. La distancia entre dos puntos debe ser un número positivo y siempre definido, $d(A,B) \geq 0$.
2. Identidad. La distancia entre un punto y él mismo, siempre es cero, $d(A,A) = 0$.
3. Conmutatividad. La distancia de un punto "A" a un punto "B", es la misma que del punto B al punto A, $d(A,B) = d(B,A)$.
4. Triangulo de desigualdad. Tomando un punto intermedio entre A y B llamado C, se puede dividir la función distancia en: $d(A,B) = d(A,C) + d(C,B)$.

Para la *función de distancia*, los puntos A, B o C, son registros de la base de datos. Y se utiliza para buscar las similitudes entre dos registros en estudio, asignándoles una distancia de proximidad, mientras más cercana sea a cero esa distancia, quiere indicar que son casi idénticos (d(A,B) = 0, es decir A es idéntico a B).

Existen tres formas comunes de definir una función de distancia entre campos numéricos:

a) Valor absoluto de la diferencia: |A-B|
b) Cuadrado de la diferencia: $(A-B)^2$
c) Normalizar el valor absoluto: |A-B| / (diferencia máxima)

La que se recomienda es la c), ya que los valores tienen un rango de entre 0 y 1. Cuando se tienen varios campos a comparar, se debe hacer una mezcla de todos los resultados parciales, siendo necesario hacer una función de distancia que agregue las funciones de distancia de cada campo, calculadas por separado. Existen tres formas para la función de distancia entre registros:

a) Suma total:

$$d_{sum}(A,B) = d_{campo1}(A,B) + d_{campo2}(A,B) + \ldots + d_{campon}(A,B)$$

b) Distancia euclidiana:

$$d_{euclid}(A,B) = \sqrt{(d_{campo1}(A,B))^2 + (d_{campo2}(A,B))^2 + \ldots + (d_{campon}(A,B))^2}$$

c) Suma total normalizada:

$$d_{norm}(A,B) = \frac{d_{sum}(A,B)}{\max(d_{sum})}$$

Las diferencias entre estas tres formas de calcular las distancias entre registros:

- Función Suma total entregará un resultado máximo como el número de campos que sean integrados, esto es, valor máximo de 20 si hay 20 campos en el análisis. De forma idéntica serán ordenados los vecinos cercanos con la
- Función suma total normalizada, solo que el valor entregado estará en el rango de 0 y 1. Siendo en la mayoría de las ocasiones un poco más fácil de identificar los resultados con esta última técnica. Además la característica es que todos los campos son considerados y tratados exactamente igual.
- La distancia euclidiana, considera a todos los campos como iguales.

En cada una de estas funciones se pueden agregar factores de ponderación dependiendo de la importancia relativa de cada campo según la aplicación.

El uso de cada una de ellas dependerá del enfoque particular que se requiera dar para un estudio en específico. Para cada tipo de dato se pueden hacer ciertas modificaciones para asegurar que la función de distancia represente lo que se esta buscando.

En una comparación de campos numéricos de edad o salario, la función sólo hará una resta y lo dividirá entre la diferencia mayor que haya encontrado (sí es que se esta utilizando la función suma total normalizada).

Para el caso de comparar datos como son el sexo, códigos postales o números telefónicos, se requiere hacer una modificación, ya que no se puede realizar una resta numérica. Por ejemplo, en la comparación de sexo, si es M(asculino) y F(emenino), se encontrará un valor de 0 si los dos son iguales d(M,M) = 0 ó d(F,F) = 0, pero será 1 si son distintos d(F,M) = 1 ó d(M,F) = 1, para el caso de números postales, se pueden hacer ciertas consideraciones (región, localidad)que ayuden a obtener un resultado ilustrativo.

Función de combinación. Esta función intenta determinar si un conjunto de vecinos es cercano o no. Hay dos formas principales de construirla.
La primera es dar un peso a las características de cada registro y calcular el valor de retorno de la función de distancia. Esto es totalmente diferente para cada problema en particular.

La función básica es *asignar peso a los datos*, que es dar valores complementarios a la función de distancia. Una ayuda que se puede utilizar son las herramientas estadísticas de interpolación y regresión lineal para cuando se está trabajando con datos continuos. La regresión lineal es muy interesante para cuando solo se tiene una variable y se desean hacer predicciones.

La segunda, es una función llamada "democracia"; esta requiere de tener el número de categorías (c) y el número de vecinos ($c + 1$), los cuales determinaran cuantos subconjuntos de vecinos se formaran, teniendo estos muchas características en común.

Elección del número de vecinos. Esto depende de la distribución de los datos, y de la solución que se requiere para el problema. Pero hay que entender que no hay un número correcto, es decir, no existen reglas. Aunque una buena práctica indica que un número suficiente de vecinos es mejor que uno escaso, esto será en comparación con el total de registros que se estén tratando. Esto se puede obtener haciendo pruebas al variar este número, pero se deberá tener una pequeña idea de lo que se quiere obtener como resultado para saber en que momento dejar de probar.

MBR es una técnica poderosa de la minería de datos dirigida que puede resolver una amplia variedad de problemas. Eligiendo un correcto conjunto de datos de

entrenamiento (esencial para que el modelo realice predicciones óptimas), con un número suficiente de registros de cada categoría para hacer búsquedas bien proporcionadas.

Sus ventajas. Produce resultados que son fáciles de entender; las reglas de asociación que entregan las técnicas de análisis de la canasta de mercado y árboles de decisión (*Si-entonces)* no proporcionan tanta información del por qué de la elección de un registro como vecino cercano o no de otros registros; la justificación es la similitud o diferencia entre los campos de los registros.

Se aplica a casi todos los tipos de datos; esta técnica solo es dependiente de la calidad de las funciones de distancia y de combinación. No depende de la representación de los datos en la base de datos; es la única técnica de la minería de datos que puede ser considerada como independiente de la representación de los datos al momento de ser analizados.

Trabaja eficientemente en casi cualquier número de campos por registro; esto es, que no tiene un límite de campos para poderlos manejar correctamente.
Mantener un conjunto de entrenamiento adecuado requiere de un mínimo esfuerzo.

Sus desventajas. Se requiere un alto poder de cómputo al hacer clasificación y predicción; ya que se requiere hacer una búsqueda en todos los registros de la base de datos, en todos los campos de estos, calculando las funciones de distancia y función de combinación, siendo esto una inversión de tiempo considerablemente alta, esto se lleva a cabo durante el entrenamiento del modelo.

Depende del número de registros existentes en el conjunto de entrenamiento, esto es, si el número de registros en este conjunto es el adecuado, y por obvias razones debe ser grande, entonces se podrá trabajar adecuadamente, de lo contrario, no será posible obtener resultados correctos.

Depende de las funciones de distancia y combinación; además del número *k* de vecinos cercanos elegidos, pero estos resultados son fáciles de comprobar que estén correctos.

DETECCIÓN AUTOMÁTICA DE CLUSTERS (AUTOMATIC CLUSTER DETECTION)

La detección automática de cluster es una técnica de la minería de datos que se caracteriza por ser un conocimiento descubierto no dirigido, además de ser un aprendizaje no dirigido. Esta técnica se basa en hacer una subdivisión del total de la población de registros que se requieren analizar.

Estas subdivisiones se llaman "clusters", los cuales son encontrados a partir de algunas variables que son las que determinaran el estudio de los registros. Tiene su importancia en la minería de datos, porque no requiere de ningún tipo de tratamiento de los datos para ser aplicada esta técnica, se puede utilizar como la primer forma de minar los datos antes de utilizar cualquier otra forma de descubrimiento de conocimiento.

Resulta extremadamente útil, ya que al ser la primera técnica aplicada en los datos, es la que indicará por donde deberán seguir enfocándose los esfuerzos de la minería de datos.

Los usos más frecuentes de esta técnica son:

- Construir un árbol de decisión con una etiqueta del cluster encontrado y la variable objetivo para particionar de primer instancia el árbol y así encontrar aquellas reglas para poder asignar un registro dentro de un cluster dado.
- Usar una visualización para ver como los clusters son afectados por los cambios en las variables de entrada.
- Examinar las diferencias entre las distribuciones de las variables de cluster a cluster, una variable a la vez.

Un de los métodos que se utiliza más frecuentemente en esta técnica es uno llamado "K medias". Este es un método enteramente estadístico, el cual tiene un algoritmo perfectamente definido para realizarse y así obtener los clusters resultantes.

Las consideraciones que se deben tomar para la realización de este método, son:

- Tener un número previamente definido del número de clusters que se quieren formar finalmente.
- Tener por lo menos un número de registros igual o mayor al número de clusters que se desea formar.
- Un equipo de cómputo que pueda realizar tantas veces como sea necesario realizar los ciclos de trabajo en el proceso de encontrar los clusters. Hay que considerar que, mientras mayor sea el número de registros a procesar, mayor será el número de veces que el ciclo se repetirá, por lo que mayor será el tiempo de cómputo requerido para obtener este resultado.

El algoritmo se detalla a continuación:

Paso	Actividad
1	Escoger puntos aleatóriamente entre todos los registros, que representarán los centros de clusters. $Z_j(1)$, j=1..k. Donde $Z_j(1)$ es el centro del cluster j en el ciclo (1). Donde j = 1..k, siendo k el número de clusters que se desea obtener.
2.	Calcular distancias desde todos los puntos hacia los centros aleatorios declarados de los clusters. $\{\overline{x_i}\}_{i=1}^{N}$ Se clasificará a cada punto del espacio ($\overline{x_i}$) de acuerdo a una distancia mínima respecto a los distintos centros aleatorios propuestos de clusters.
3.	De acuerdo con los nuevos puntos que integran los distintos clusters, calcular de nuevo el centro de cada cluster. $$Z_j(t) = \frac{1}{N_j}\sum_{i=1}^{N_j}\overline{x_i}$$
4.	Condición de paro, si existe convergencia, de lo contrario se regresará al paso 2. $$\overline{Z}_j(k+1) = \overline{Z}_j(k) \qquad \forall\ j = 1, .. , k$$

$$d(Z_1) \quad \ldots \quad d(Z_k)$$

$\overline{x_1}$

$\overline{x_2}$

...

$\overline{x_N}$

Desafortunadamente, no siempre es tan sencillo poder hacer un ordenamiento de los datos, ya que no siempre se encuentran en formato numérico, por ejemplo, al hacer un ordenamiento de boletos de avión, reservaciones de rentas de automóviles, llamadas telefónicas, y cientos de otras cosas que no tienen una obvia conexión con puntos en un diagrama.

Para los casos en los que los datos a ordenar en clusters no son numéricos, una solución, pero no la única, es dar una calificación a estos atributos, siendo transformados en datos numéricos. Esta calificación se da arbitrariamente pero teniendo un significado.

Sin embargo, hay dos problemas cuando esto sucede:

1. Muchos tipos de variables cuando son "calificadas", esto es, convertidas a valores numéricos, no tienen un comportamiento adecuado, ya que originalmente no eran datos de tipo numérico.
2. En el sentido geométrico, las dimensiones son categóricamente iguales, pero en las bases de datos, un cambio mínimo en un campo "clave" de un registro, puede implicar un cambio drástico, en comparación al modificar un campo de poca relevancia.

Para evitar estos problemas, se da una clasificación a los distintos tipos de variables. Los cuales se listan en orden de mayor conveniencia para el modelo geométrico.

- Categorías.
- Listas.
- Intervalos.
- Mediciones reales.

Categorías. Las variables categóricas solo indican la pertenencia de una variable a un cierto segmento, más no si es mayor que otra, mejor o peor, si está o no cerca de otra categoría. Se diría que:

Se conoce	Se desconoce
$X = Y, X \neq Y$	$X > Y, X < Y$

Listas. Las listas indican un orden, ya que las variables son posicionadas de forma ascendente o descendente, pero no se sabe cuan mayor es una de la otra, que tan cercano esta el último dato del primero. Se diría que:

Se conoce	Se desconoce
$X > Y > Z$	$(X - Y) = (Y - Z)$

Intervalos. Los intervalos nos permiten saber la distancia que existe entre dos datos. Por lo que podemos saber más detalles del mismo dato.

Mediciones reales. Son intervalos de variables que se miden con respecto al punto de referencia (patrón) cero. Cero como escalar, vector, etc.

Medición formal de la asociación

Dadas las variadas opciones que existen de técnicas que realizan estas mediciones, solo tres serán las que se detallarán, por su efectividad y sencillez. Dos de estas son manejadas por un uso con intervalos y la otra es con variables categóricas.

Distancia entre dos puntos. Cada uno de los campos de cada registro, será un elemento del vector que esta siendo descrito. Por lo que un vector cercano a otro, significa

que son registros muy similares. La forma más común de obtener esta distancia es con la formula de la distancia Euclidiana entre dos vectores.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Pero cualquier función debe ser una métrica real de distancia cumpliendo los siguientes criterios:

- Distancia (X,Y) = 0 sí y solo sí X = Y
- Distancia (X,Y) ≥ 0 para todas las X y todas las Y
- Distancia (X,Y) = Distancia (Y,X)
- Distancia (X,Y) ≤ Distancia (X,Z) + Distancia (Z,Y)

Angulo entre dos vectores. Cuando se desea saber que tan cercanos están dos registros, se debe usar una interpretación geométrica diferente para los mismos datos. Esto se logra obteniendo el ángulo de separación entre los dos vectores X y Y que están siendo comparados. En este caso particular, la magnitud del vector no importa, solo la dirección de este. Con esto se logra que el ángulo de separación entre los vectores no sea influenciado por las diferencias en magnitud de los registros. Para mantener la estructura de trabajar con valores normalizados, se obtiene el seno del ángulo y así, los vectores cercanos tienen un valor cercano al 0, los perpendiculares serán representados con un 1. Los signos no son importantes.

Número de características en común. Esto consta en realizar una comparación campo por campo de dos registros. Cuando el campo de uno difiera con respecto al otro, se puede dar un valor de 1, teniendo como valor máximo el número de campos en un registro. Pero si se requiere categorizar algún campo, se puede asignar un mayor "peso" (valor) a algún(os) campo(s) específico(s).

COMO FUNCIONA EL MÉTODO DE LAS K MEDIAS

Si se tuvieran en un cluster variables que todas fueran dependientes entre sí, solo se podría formar un cluster conteniendo todos los puntos. Por el contrario, si todas fueran independientes, se formarían tantos clusters como puntos en el espacio de trabajo.
Encontrando un punto medio de estos dos extremos, no se sabe cuantos clusters son el número adecuado. Por lo que sí aún no se tiene especificado el número exacto de clusters que se desea formar, se debe realizar varias veces el algoritmo de las K medias para probar con distintos valores de k, hasta obtener aquella configuración que más favorezca a la resolución del problema en particular que se está desarrollando.
La forma en la que se comprobará esto es, en cada ciclo (por cada valor de k), se deberá comparar el promedio de las distancias entre los registros de un cluster con el promedio de las distancias entre clusters, entre otros procedimientos. Pero se debe entender que los clusters deben ser evaluados con métodos subjetivos (para la conveniencia de una mejor solución para el problema).

UN TRATAMIENTO MÁS REAL PARA LOS REGISTROS

El escalamiento se ocupa de que variables diferentes no sean medidas en unidades diferentes. Esto es, cuando se están midiendo distancias en la geometría Euclidiana, los ejes coordenados tienen las mismas dimensiones, por lo que un vector con coordenadas (1,3,3) esta igualmente retirado de otro vector con coordenadas (3,3,1) que un tercer vector (3,4,5); esto es porque los ejes coordenados X, Y y Z se encuentran en las mismas unidades. Pero si se realizara un cambio de unidades, en las que X esta en centímetros, Y en pies y Z en millas, pues no seria la misma distancia que separe estos tres vectores. Desafortunadamente no se pueden hacer estos cambios con variables como edad, peso, sueldo o cosas tan distintas, pero la solución es *escalar* las variables, logrando así que los valores de todas las variables estén en un rango de 0 a 1, obteniendo un cambio en sus valores que sea proporcional entre todas las dimensiones del problema. Esto se logra (existen muchas formas, pero esta es la que se recomienda) dividiendo cada campo entre el campo con el mayor valor (esto es, entre el número más grande, para que este sea el que tenga valor de 1, el menor de 0). Entonces todos los campos contribuyen de igual forma para la distancia entre los registros.

El asignar "peso", se encarga de darle más importancia a ciertas variables. Siendo esta la solución cuando se tienen dos registros que resultan estar muy cercanos (o lejanos) pero hay ciertos campos del registro que tienen mayor importancia que otros (para los propósitos de la minería de datos que se está realizando en particular), por lo que se les asigna un *peso* mayor, para que las diferencias entre los registros en estos campos, sean más notorias y determinantes. Encontrar el "peso" exacto que indique lo que se quiere es un problema de optimización, por lo que se puede hacer en base a prueba y error. Aunque la técnica de algoritmos genéticos mejora esto de una forma muy interesante.

Obviamente, el resultado óptimo, es cuando se realiza primeramente un escalamiento de todos los campos de todos los registros. Después, se recomienda ajustar pesos a los campos para se obtengan resultados mucho más interesantes en la solución del problema.

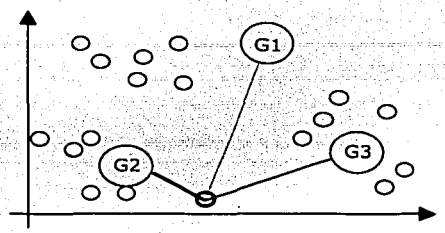
VARIANTES DEL MÉTODO DE LAS K MEDIAS

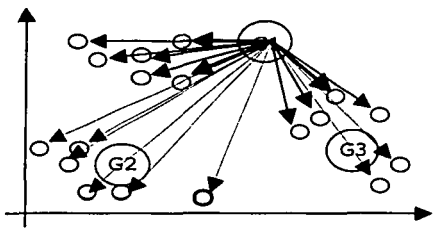
Dada la potencia de este método, existen muchas variantes del mismo que cada productor de software realiza para que su producto sea aún más poderoso. Pero dentro de la gama de variantes que existen en el mercado, la que es más importante es la que a continuación se detallará.

Modelo de Mezcla Gaussiano

Son una variante probabilística del método de las K medias. Su nombre proviene de la distribución gausiana. El número de k, es una medida de distribución Gausianas. Se tienen dos funciones llamadas: Paso de estimación y Paso de maximización.

En el paso de estimación; se calcula la responsabilidad de cada Gaussiano que tiene para cada punto. Como lo muestra la figura a la derecha, cada uno de los Gaussianos (G1, G2 y G3) tiene una proximidad con el punto al cual están unidos por medio de una línea, la cual, por su grosor indica mayor o menor responsabilidad (mayor o menor grosor de la línea respectivamente).





En el paso de maximización; el promedio de cada Gaussiano es movido al centro del conjunto de datos. Otorgándole una "responsabilidad" según la cercanía de los datos con respecto al Gaussiano. Los Gaussianos pueden ser movidos y llegar a crecer, pero esto los puede hacer fuertes o débiles. La responsabilidad de dos Gaussianos para con un punto, puede ser la misma cuando un Gaussiano cercano tiene una varianza baja y cuando un Gaussiano lejano tiene una varianza alta.

La razón por ser un modelo de Mezcla, es porque la probabilidad de cada registro (punto) es la suma de varias distribuciones. Al final del proceso, cada punto es relacionado con algunos clusters, con alta y baja probabilidad (también conocido como clustering suave).

Métodos de aglomeración

En el método tradicional de las K medias, se trabaja con unos puntos iniciales propuestos como centros de los clusters (número propuesto antes de iniciar el algoritmo) y todos los puntos son acomodados dentro de estos clusters. En el método de aglomeración, se inicia haciendo que cada uno de los registros en el espacio forma su propio cluster y gradualmente se fusionan los puntos hasta formar un gran cluster. Antes de formar un solo cluster que englobe todos los puntos del espacio, se debe determinar que nivel de aglomeración es el adecuado, cuantos clusters son lo que convienen para la solución del problema.

ALGORITMO DE AGLOMERACIÓN

El primer paso es crear una matriz de similitudes, es decir una tabla con todos los pares de distancias o grados de asociación entre los puntos. Se puede usar cualquier función de asociación (distancia Euclidiana, ángulo entre vectores, etc.). Una

consideración es que no se requerirán N^2 renglones en la matriz, donde N es el número de vectores, ya que Distancia (X,Y) = Distancia (Y,X), por lo que se requiere $\frac{N^2}{2}$ renglones para la matriz, pero al inicio del proceso, se tienen los N renglones por cada registro. Esta es una matriz triangular inferior.

El segundo paso se busca los dos renglones con el valor de similitud más grande, lo cual englobará dos renglones (clusters) en uno solo, se recalcula la nueva distancia entre el cluster mezclado y los clusters restantes. Existiendo N-1 clusters y N-1 renglones en la matriz.

Esto se puede realizar N veces, donde N es el número de renglones, obteniendo finalmente un solo cluster. Pero se puede detener en cualquier momento para obtener el número de clusters deseado.

Medición entre clusters

La forma de medición entre clusters puede ser cualquiera de las tres más comunes, descritas a continuación:

1. Unión simple. Es la obtención de la distancia entre los dos elementos más próximos de los dos clusters. Se encuentran los elementos que están en los límites próximos de los clusters.
2. Unión completa. Es la obtención de la distancia entre los dos elementos más lejanos de los dos clusters. Se encuentran los elementos que están en los límites extremo de los clusters.
3. Comparación de centróides. Es la obtención de la distancia entre los dos centros de los clusters. Estos se encuentran realizando un promedio de los registros, para obtener el punto de centro de masa.

La forma de medir la similitud entre clusters es la varianza (suma de las diferencias elevadas al cuadrado de cada uno de los elementos). La mayor similitud estará dada por una varianza pequeña.

Otras técnicas para detección automática de clusters son los árboles de decisión y las redes neuronales.

Sus ventajas. Es una detección automática de clusters, lo cual es una técnica de conocimiento descubierto no dirigido, descubriendo estructuras ocultas que pueden ser usadas para mejorar los resultados de técnicas dirigidas.

Se puede aplicar a tipos de datos categóricos, numéricos y textuales. Además de ser fácil de implementar.

Sus desventajas. Puede ser difícil elegir los promedios correctos de distancias y "pesos", ya que esta técnica de la minería de datos depende totalmente de la elección de estos datos, de lo contrario será un estudio en vano. Esto se dice que es una sensibilidad a los parámetros iniciales. Y, no se tiene una garantía de que los clusters detectados tendrán una interpretación o uso práctico para la resolución del problema.

Todo esto se vera aún más complicado si no se tiene una idea muy clara de que es lo que se esta buscando.

ANÁLISIS DE ASOCIACIÓN (LINK ANALYSIS)

Esta técnica estudia aquellas relaciones que asocian a un tipo de dato con otro, dadas ciertas características intrínsecas de los mismos.

Esta técnica es aplicable cuando no se encuentran patrones con otras técnicas. Se utiliza para descubrimiento no para predicción o clasificación. Puede ser precursora de otras técnicas como redes neuronales o árboles de decisión. Esta técnica se basa en la *teoría de Grafos*. Los grafos son muy útiles para visualizar relaciones.

Se ha utilizado principalmente para analizar:

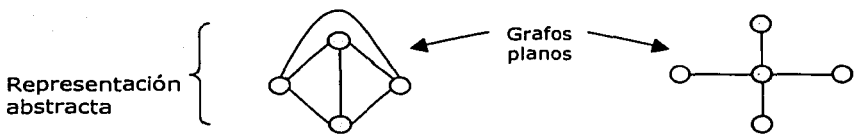
- Patrones en llamadas telefónicas.
- Patrones de prescripción de medicamentos.
- Combinación de datos para resolver crímenes de la policía.
- Preferencia de clientes con tarjetas de crédito hacia restaurantes o tiendas.
- Respuesta de usuarios de ciertos sitios Web ante anuncios.

Una implementación posible de esta técnica es en consultas a bases de datos relacionales, donde por medio de un patrón clave a buscar, se entrega como resultado todos aquellos registros que lo contienen, es decir, se obtuvo su relación entre ellos a partir de un campo común.

La teoría de Grafos es una herramienta de un uso intuitivo; es una grafica abstracta que se usa para representar relaciones.

Un grafo consta de dos partes, que son:

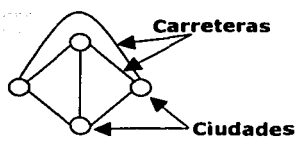
- Nodos. Son registros, datos o cosas que tienen relaciones en el diagrama. Que se representan por puntos y se les pueden colocar algunas características adicionales.
- Aristas. Son líneas que unen dos nodos, mostrando su relación.



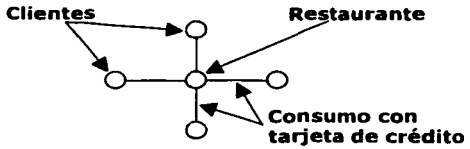
Los dos grafos mostrados (arriba) son *planos*, esto es, que ninguna de las aristas que relacionan a los nodos se intersecan.

El grafo de la izquierda es uno *totalmente conectado*, esto es, que entre todos los nodos existe relación, teniendo **n** nodos y **n!** formas de recorrer las aristas.

Representación de entes reales



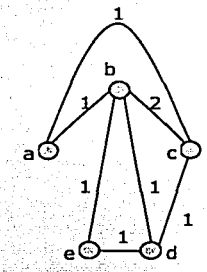
Representación de las carreteras de una ciudad.



Representación de los clientes con el restaurante.

El grafo que se muestra a continuación, es un tipo de *Grafo con pesos*. Este tipo de grafo muestra la importancia de una relación, o número de veces que una cierta relación se ha repetido.

Ejemplo: retomando un ejemplo de la técnica análisis de la canasta de mercado, cada uno de los nodos son los productos que se vendieron en una serie de transacciones vistas en la técnica de análisis de canasta de mercado. Los productos:



Grafo con pesos

Nomenclatura	Producto
a	Detergente
b	Jugo
c	Refresco
d	Limpiador
e	Leche

Esta es una forma de lograr una visualización de las transacciones analizadas con la técnica análisis de la canasta de mercado.

Otro ejemplo del uso de los grafos, sería para determinar cual es la mejor ruta para llegar de un nodo a otro, suponiendo que estos sean destinos unidos por carreteras, teniendo dos grafos del problema, uno que representaría las aristas con pesos equivalentes a la distancia entre los nodos y otro con pesos equivalentes al costo que implica esa carretera.

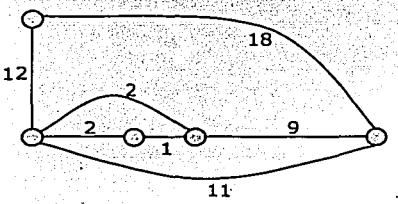
El número de aristas que tiene un nodo se conoce como *grado*. Así también, cada uno de *los nodos tiene un grado*, que son las aristas que lo relacionan con los demás nodos.

Un problema clásico que se resuelve con los grafos es la *Ruta Euleriana*. El cual se resuelve al recorrer cada una de las aristas una vez, esto sin importar cuantas veces se recorra un mismo nodo. Euler demostró que para lograrlo se necesita que todos los nodos deban tener un grado par, excepto máximo dos finales. De lo contrario no seria posible.

Otro problema clásico es la *Ruta* Hamiltoniana conocido como el *problema del agente viajero*. Problema que consiste en encontrar aquella ruta que recorra cada uno de los nodos una sola vez y que la ruta total sea la menor. Las aristas tienen un peso, que es la distancia entre los nodos.

El número de posibles rutas que cubren todos los nodos crece exponencialmente con el número de nodos.

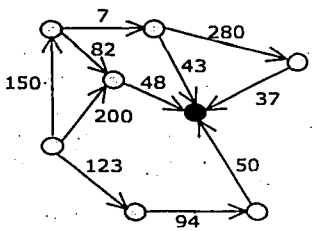
Frecuentemente es mejor utilizar algoritmos heurísticos que producen buenos resultados, pero no perfectos, en lugar de intentar analizar algo muy complejo para llegar a la solución perfecta.



En este grafo se muestra como los nodos son ciudades, conectadas por carreteras que son las aristas y estas tienen un peso, es decir un tiempo que se tardaría en recorrer. Por lo que hay que determinar la mejor ruta, la cual es ABCDE.

Este problema se complica enormemente cuando existe comunicación entre cada uno de los nodos y son un número considerablemente mayor al presentado en este ejemplo. Además de poder tener un segundo peso las aristas, por características propias del problema.

Una aplicación de esta técnica, ha sido para el análisis del comportamiento de líneas telefónicas. Esto consiste en tener un registro por cada una de las llamadas realizadas por todos los clientes (números telefónicos), teniendo datos como, número de donde se originó la llamada, número a donde llega la llamada, duración de la misma, fecha de realización, día de la semana, hora de la llamada, etc. Considerando esto, si es que una línea telefónica se utiliza para conexiones a Internet, se puede ofrecer un plan tarifario diferente para el cliente, una línea telefónica adicional o cualquier otro plan de consumo que con el que se mejoren los ingresos para la empresa.



Comportamiento de líneas telefónicas

El grafo utilizado para este estudio de las líneas telefónicas, seria de la siguiente forma. Los nodos serian los números telefónicos, las aristas serian dirigidas, esto es, con una flecha que indica origen y destino de la llamada telefónica, además de tener un peso, que esta determinado por la duración de la llamada.

Se pueden hacer conclusiones de este grafo, como que el nodo negro, puede ser un número telefónico de ayuda, ya que el promedio de todas las llamadas que recibe es muy parecido, además de que todas las aristas tienen un solo sentido, el de entrada. El nodo sin relleno, es un número telefónico que realiza llamadas

muy extensas, por lo que se le puede ofrecer un plan tarifario al usuario, para que le sean más atractivos los precios.

Pero estas no son las únicas conclusiones a las que se puede llegar y por lo tanto convertir estas asociaciones en entendimiento de las necesidades de los clientes. Se podría detectar cual número telefónico esta siendo utilizado para envío de fax, conexiones a Internet o cuales de esas llamadas fueron equivocadas, entre otras tantas posibilidades.

Un grafo como el anterior, es un *grafo dirigido*. El cual tiene en sus aristas un sentido, el cual no significa ni implica las mismas condiciones en un sentido inverso. Los nodos a los cuales solo llegan aristas, son conocidos como *nodo pozo*, mientras los nodos de los cuales solo salen aristas son llamados *nodo fuente*.

Son utilizados principalmente en representaciones de:

- Segmentos de vuelos de aviones que unen un conjunto de ciudades.
- Patrones referenciales de tratamientos médicos.
- Patrones de llamadas telefónicas.
- Diagramas de transición de estados.
- Árboles de decisión.

Un grafo dirigido, puede mostrar un *grafo cíclico*, el cual tiene por lo menos una secuencia bien definida que se repite al menos una vez. Como puede suceder en vuelos de un aeropuerto.

Esto se logra al observar cuando un nodo pozo, no es un nodo terminal, es decir siempre ese nodo pozo es también un nodo fuente. Siendo esto un ciclo que se puede desarrollar cualquier número de veces.

Cuando no existen ciclos en un grafo, estos son llamados *grafos acíclicos*. Utilizados principalmente para representar relaciones de dependencia o relaciones de un solo sentido.

Por otro lado, los nodos que no cuentan con un sentido en las aristas son los *grafos no dirigidos*. Para los cuales, el sentido de las aristas es indistinto.

Una generalización de los grafos utilizados para el análisis de asociación son las representaciones graficas llamadas *redes*.

Los dos roles que juega el análisis de asociación en el estudio sobre un conjunto de registros son:

1. Poder de visualización. Esta facilidad que presenta un análisis de asociación que se ve muy clara en la representación grafica que se tiene de un grafo; puede ser contraproducente, si se quiere representar una gran cantidad de información por este método. Solo hay que representar aquellos registros que estén en análisis y se requiera tomar una decisión sobre su comportamiento.

2. Aplicación de resultados a grandes volúmenes de información. Esto es, dentro de un gran volumen de información, se pueden hacer generalizaciones, de tal forma, se realizan ajustes a las conclusiones obtenidas y así abarcar a un mayor sector de clientes.

Sus ventajas. Se aprovechan las relaciones; esto es, en ciertas áreas la asociación es una parte intrínseca de la actividad, como en las telecomunicaciones o transportaciones. Siendo esto una facilidad para representarse con esta técnica, más que con cualquier otra.

Se explota la visualización de los datos; hay que entender que una herramienta importante para solucionar un problema es la visualización o representación grafica del mismo, por lo que es de gran ayuda para descubrir conocimiento y no solo es descrito el problema con enunciados que representan reglas de asociación.

Se pueden crear atributos derivados de un análisis, esto es, extraer características (atributos) muy particulares que pueden ser claves en el entendimiento de la solución.

Sus desventajas. No es aplicable a muchos tipos de datos; esto se debe a que no es una herramienta que logre hacer una predicción o clasificación, pero en los datos que si puede ser aplicada, produce resultados muy poderosos.

Pocas herramientas en el mercado soportan esta técnica; ya que la representación de los grafos de cientos o miles de registros, requiere de algoritmos sumamente complejos.

Su implementación en bases de datos relacionales es ineficiente, ya que estas asociaciones en grandes volúmenes de datos, requiere de hacer consultas (joins) que consumen mucho poder de cómputo y tiempo. Por lo que se complica su implementación en estas bases de datos.

ÁRBOLES DE DECISIÓN (DECISION TREES)

Los árboles de decisión son una poderosa y popular herramienta para realizar clasificación y predicción.

Los métodos basados en los árboles de decisión pueden representar reglas, las cuales pueden ser expresadas en nuestro idioma (hacerlas aún más tangibles para toma de decisiones en el negocio) o directamente en consultas de SQL. Además, en muchos problemas a solucionar, se requiere de una justificación sobre ese resultado obtenido, por lo que esta técnica se vuelve una gran opción en comparación con las redes neuronales.

El principio básico de los árboles de decisión es que cada nodo del árbol representa un atributo de los registros del problema, debiendo partir de un nodo raíz, el cual hace la mejor discriminación sobre el punto de partida de clasificación o predicción que se desea obtener, se realiza una prueba al registro en análisis con respecto a este nodo raíz, después se continua con un nodo *hijo*, el cual a su vez también realiza una prueba al registro y así sucesivamente hasta llegar a un nodo final llamado *nodo hoja*. Esta arquitectura arborescente invertida, es la que da el nombre de árboles de decisión.

Se puede describir como un grafo conexo acíclico dirigido.

Si se realizan pruebas a los registros de tal forma en que estas sean planteadas correctamente, en muy pocos niveles del árbol, se pueden encontrar clasificaciones muy interesantes.

Finalmente, la ruta que se sigue desde el nodo raíz hasta el nodo hoja, representan reglas que se usan para clasificar registros.

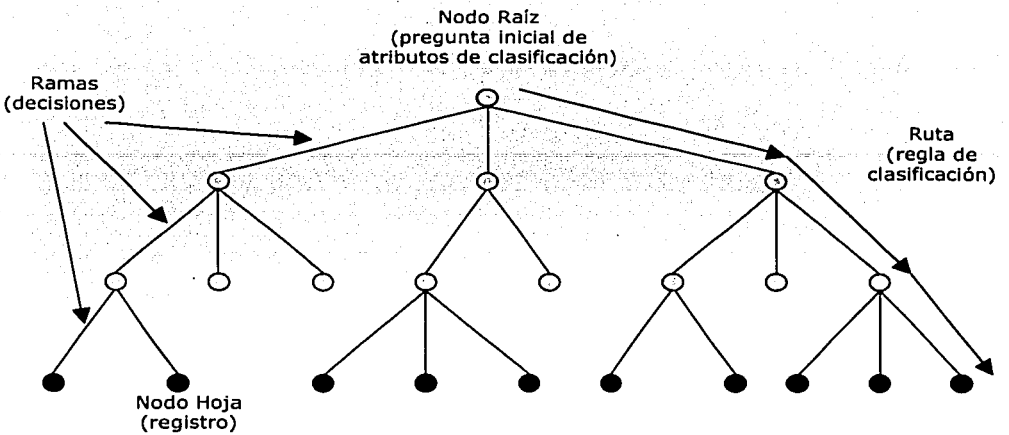
Como se muestra en la siguiente figura, un nodo hoja tiene color obscuro, lo cual indica que es un registro ya clasificado, un nodo de decision es un nodo de color gris y no es un registro, sino una pregunta que se realiza para poder hacer una clasificación sobre los registros. Algunos autores los sustituyen por cuadrados.

Existe una variedad de algoritmos para construir árboles de decisión, con los cuales se obtienen resultados ligeramente diferentes. Estos algoritmos son:

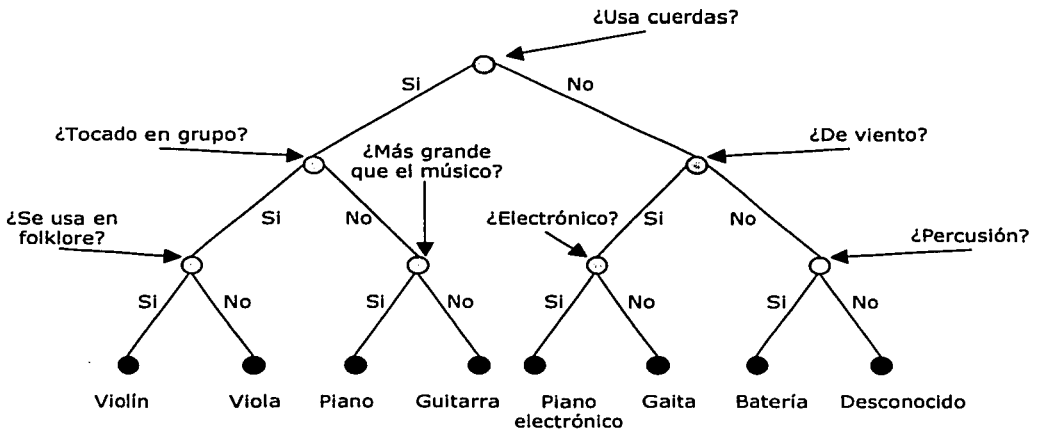
- CART
- CHAID
- C4.5
- ID3

Existen dos tipos principales de árboles de decisión:

- Árboles de clasificación. Proveen un nivel de confianza de clasificación y asignan a los registros en una clase según sus características.
- Árboles de regresión. Estiman el valor de la variable objetivo (valores numéricos).



Un árbol de decisión mezclado, con ramificaciones binarias y ternarias.



Un árbol de clasificación de instrumentos musicales.

CHAID (CHI squared Automatic Interaction Detection). Detección de interacción automática de Chi cuadrado: Una técnica de árbol de decisión usada para la clasificación de un conjunto de datos. Provee un conjunto de reglas que se pueden aplicar a un nuevo conjunto de datos (sin clasificar) para predecir cuáles registros darán un cierto resultado. Segmenta un conjunto de datos utilizando tests de chi cuadrado para crear múltiples divisiones. Antecede, y requiere más preparación de datos, que CART.

$$P(X_0)_D = \int_0^{X_0} p(x)_D dx$$

donde:

- $P(X_0)_D$ es la distribución chi cuadrada con D grados de libertad y
- X_0 es el valor de la estadística para un atributo dado.

Se puede realizar el calculo de la entropía: H_C (entropía de las clases), H_A (entropía de los valores de un atributo dado), H_{CA} (entropía conjunto de clases - valores de atributos), $H_{C|A}$ (entropía de las clases dado el valor del atributo). Donde:

- C = número de clases
- A = número de atributos
- V = número de valores de un atributo específico
- n = número de ejemplos de entrenamiento
- n_i = número de ejemplos de entrenamiento de la clase C_i
- n_j = número de instancias con el j-ésimo valor del atributo
- N_{ij} = número de instancias de la clase i y valor j-ésimo de atributo
- $p_{ij} = n_{ij}/n_{..}$
- $p_i = n_{i.}/n_{..}$
- $p_j = n_j/n_{..}$

$$H_C = -\sum_i p_i \log p_i \qquad H_A = -\sum_j p_j \log p_j$$

$$H_{CA} = -\sum_i \sum_j p_{ij} \log p_{ij} \qquad H_{C|A} = H_{CA} - H_A$$

Todos los logaritmos son en base 2. La ganancia de información se define como la información transmitida por el atributo acerca de la clase del objeto:

$$Ganancia = H_C + H_A - H_{CA} = H_C - H_{C|A}$$

C4.5: Es la más reciente de estas técnicas, a la cual se le han aplicado múltiples mejoras y correcciones desde 1986 (primera versión). No existe una explicacion del nombre. Es similar a CART.

Una de las formas en las que se puede medir la efectividad de un árbol de decisiones es observando el porcentaje de error cuando se clasifica un conjunto de registros sin previo tratamiento (sin clasificación)

Cada uno de los nodos que no son nodos hojas, están haciendo una partición de los registros, por lo que de la correcta elección de estos nodos, dependerá totalmente el buen funcionamiento y una obtención de reglas de clasificación y predicción con un alto porcentaje de efectividad.

La longitud de un árbol de decisión estará determinada por el número de subconjuntos que se requieran formar a través de los nodos anteriores a los nodos hoja, es decir, se tendrán tantos niveles como se requieran hasta obtener una clasificación adecuada según el problema que se este tratando.

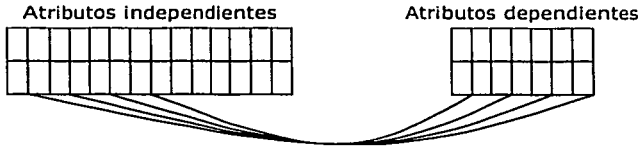
Una consideración importante en la construcción de un árbol de decisión es el número de particiones por nivel, la elección de las características a particionar el árbol y si crecimiento medido para evitar un sobre entrenamiento.

CART (Classification And Regretion Trees)

El algoritmo de CART es uno de los más populares para la construcción de árboles de decisión.

También es conocida como árboles de clasificación y regresión. Es una técnica de árbol de decisión usada para la clasificación de un conjunto da datos. Provee un conjunto de reglas que se pueden aplicar a un nuevo conjunto de datos (sin clasificar) para predecir cuáles registros darán un cierto resultado. Segmenta un conjunto de datos creando 2 divisiones. Genera árboles binarios.

Para iniciar la construcción del árbol de decisión en base al algoritmo CART, se requiere tener un *conjunto de datos de entrenamiento*, el cual consta de registros preclasificados (esto es, que el campo objetivo o las variables dependientes, ya son conocidas). Además, se tienen los datos independientes del registro que determinan el valor del atributo dependiente.



CONSTRUCCIÓN DEL ÁRBOL

Se construye el árbol binario particionando los registros en cada nodo, de acuerdo al atributo que este analizando del registro de entrada. La tarea principal es elegir el atributo independiente del conjunto de registros que hará la mejor partición.

La mejor partición de datos es la que pude separar los registros en grupos donde una clase predomina.

La forma de medir el poder de las particiones realizadas por un atributo se llama *índice de diversidad.* Existen varias formas de medir el índice de la diversidad. Un alto índice de diversidad indica que un conjunto contiene una distribución uniforme de clases, mientras que un índice bajo de diversidad indica que una clase es la que predomina.

La mejor forma de hacer esta elección del nodo principal de partición es, hacer una consideración inicial de todos los campos independientes (posibles nodos raíz), se hace un ordenamiento y se miden los índices de diversidad; aquel que tenga el más alto índice de diversidad, será comparado con otros valores y si no reducen sus índices, será elegido como nodo raíz de la partición del árbol de decisión.

Este proceso será iterativo, ya que después de la partición del nodo raíz, los nodos hijos, tendrán que ser elegidos de la misma forma, hasta el punto en el que no se encuentren más particiones que mejoren la habilidad de clasificación de registros en el árbol. Un nodo no puede ser elegido para partición si solo puede tomar un valor.

Cuando un nodo no puede ser particionado y decrementa considerablemente el índice de diversidad, se considera que es un nodo hoja (nodo final que es la conclusión de una ruta, por lo que es una categoría encontrada).

EVALUACIÓN DEL ÁRBOL CONSTRUIDO

Finalmente, al estar construido el árbol de decisión, se hacen pruebas de clasificación de registros con el conjunto de datos de prueba (son registros ya clasificados), esto con la finalidad de observar los porcentajes de acierto y error que tendrán los nodos de particionamiento. El porcentaje de acierto es medido al realizar una suma de todos los aciertos de los nodos hoja, y dividiendo ese número entre el total de nodos hoja. La diferencia respecto a uno indica el porcentaje de error del árbol de decisión.

Si el porcentaje de error de clasificación del árbol de decisión es alto, se deben reconsiderar los nodos de particionamiento, además de hacer alguna modificación al conjunto de entrenamiento.

Si no se llega a realizar una buena elección de nodos de particionamiento y la tasa de acierto es 100%, entonces, se está efectuando una clasificación específica para los registros de entrenamiento y no así, por características que puedan ser extraídas del conjunto. Al cambiar los registros, se obtendrá un porcentaje de acierto muy bajo. Porque el árbol solo memorizó el conjunto de prueba sin lograr una generalización.

Al evaluar comportamiento del árbol de decisión, pueden encontrarse ramas (conjunto de nodos y vértices) que proporcionan una tasa de acierto baja. Para corregir esta situación existe una técnica llamada *poda* de ramas débiles. Esta consiste en eliminar aquellas con tasa de acierto baja.

Para asegurar el correcto comportamiento en la clasificación de registros, se hacen pruebas con el *conjunto de datos de prueba*, en los cuales se incluyen además de los del conjunto de entrenamiento, algunos registros extra, pero se continúa conociendo aquellas

variables dependientes y campos objetivo; de ser necesario se realizan las mismas consideraciones para aplicar la técnica de poda.

La última prueba se hará en el *conjunto de datos de evaluación*, donde los datos son en su totalidad nuevos para el modelo y sin previa clasificación o conocimiento de alguna variable dependiente o campo objetivo.

C4.5

Este es el algoritmo más reciente en la construcción de árboles de decisión. Desde 1986 se han realizado mejoras sobre este algoritmo. No existe una explicación formal sobre el nombre. Además, es implementado en un programa de minería de datos llamado "Clementine" de Integral Solutions Ltd.

C4.5 es muy parecido al algoritmo CART. Pero existen las siguientes diferencias:

- El tipo de variables que se utilizan en CART son de tipo categórico, por lo que podrán ser divididas las opciones en árboles binarios, mientras que C4.5 trata con variables continuas, obteniendo más de dos opciones por nodo al hacer la partición de los registros. Lo que afectará en el número de nodos particionadores, ya que este será menor que en CART.
- La elección de una partición inicial de los registros de tipo binario, complica demasiado la elección correcta del campo que deberá determinar esta partición. La ventaja de hacer más de dos salidas a partir de los nodos de partición, es que los árboles rápidamente se terminan, es decir, no tienen una profundidad tan grande como en el caso del algoritmo CART.

Existe un criterio llamado *ganancia de información.* Nos indica que el número de bits (partes) requeridas para describir una situación particular (resultado) depende del número de posibles salidas. Por ejemplo, si se tienen 8 clases igualmente probables, se obtiene el $\log_2(8) = 3$ bits, o si fueran 4 clases seria $\log_2(4) = 2$ bits. Para el caso de las ocho clases, si se logra particionar el conjunto en 4 clases, se dice que se tuvo una ganancia de información de 1 bit (parte).

Existe otra medida basada en la información llamada *entropía*, que pede ser usada para particionar recursivamente los valores de un atributo A, resultando en una discretización jerárquica. Una discretización forma un concepto numérico jerárquico por el atributo. Dado un conjunto de tuplas S, el método básico para una discretización basada en la entropía de A es como se muestra:

1. Cada valor de A puede ser considerado un límite de intervalo potencial o nivel de confianza T. Por ejemplo, un valor v de A puede particionar las muestras en S dentro de dos subconjuntos que satisfacen las condiciones $A < v$ y $A \geq v$, respectivamente. Dando como resultado una discretización binaria.

2. Dada S, el nivel de confianza seleccionado es el que maximiza el nivel de ganancia de información resultante de las subsecuentes particiones. La ganancia de información es:

$$I(S,T) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$

Donde S_1 y S_2 corresponden a los ejemplos de S, satisfaciendo las condiciones $A < T$ y $A \geq T$, respectivamente. La función de entropía *Ent* para un conjunto dado es calculada basada en la distribución de las clases de los ejemplos del conjunto dado. Por ejemplo, dando *m* clases, la entropía de S_1 es:

$$Ent(S_1) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

donde p_i es la probabilidad de la clase ***i*** en S_1, determinada por la división del número de ejemplos de la clase ***i*** en S_1 entre el número total de ejemplos en S_1. El valor de $Ent(S_2)$ puede ser calculado similarmente.

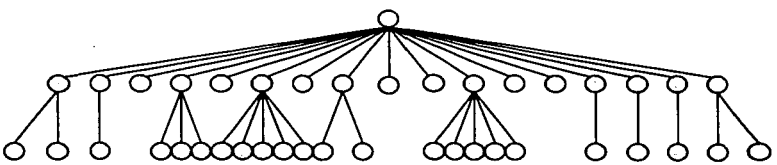
3. El proceso de determinar un valor de nivel de confianza es recursivamente aplicado a cada partición obtenida, hasta que algún criterio que detenga el este proceso sea encontrado, como lo es:

$$Ent(S) - I(S,T) > \delta$$

La discretización basada en la entropía puede reducir el tamaño de los datos.

Una forma de utilizar este criterio es usando como campo de partición el nombre de las personas como nodo raíz, con la consideración de que muy pocas personas tienen el mismo nombre. Obteniendo así una ganancia máxima.

Los posibles problemas que puede acarrear este criterio es un crecimiento de "espesor" por parte del árbol de forma inesperada, esto es, que rápidamente se está trabajando con decenas de nodos hijo en los primeros niveles e partición. Como se muestra en la siguiente figura.



Como se puede observar, rápidamente crecen el número de nodos hijos, siendo esto un problema.

En respuesta a los posibles problemas creados por la ganancia de información, C4.5 usa el *cociente* del total de la ganancia de información, debido al particionamiento propuesto en la ganancia de información atribuible únicamente al número de subconjuntos creados con el criterio de evaluación de la partición propuesta.

C4.5 difiere de CART en la forma de podar (depurar) el árbol. CART usa las mediciones de la complejidad del árbol para etiquetar algunos de los subárboles y probarlos en un conjunto de datos de prueba. C4.5 intenta podar el árbol sin utilizar dato fuera del conjunto de entrenamiento.

La poda del árbol se lleva a cabo examinando la tasa de error de cada nodo hoja, asumiendo que el error total del árbol será sustancialmente mayor. Sí N registros se clasifican en un nodo hoja y E de ellos fueron clasificados incorrectamente, entonces la tasa de error de ese nodo hoja es $\frac{E}{N}$. Ahora, el reto del algoritmo es poder minimizar esta tasa de error, por lo que $\frac{E}{N}$ es la tasa de error más pequeña que se puede lograr.

C4.5 utiliza una analogía del muestreo estadístico para llegar a un estimado del peor error del árbol. Para esto, se utilizan los conocimientos de la estadística para determinar los porcentajes en N intentos con E errores, dando un nivel de confianza.

C4.5 algunas veces sustituye un subárbol por una rama.

Finalmente, en la obtención de las reglas generadas por el algoritmo C4.5, se puede tener un número muy grande de reglas y resultar confusa su utilización, por lo que se deberá hacer una depuración dentro de aquellas que lleguen al mismo resultado con variables diferentes.

Como se observa en el siguiente ejemplo, se trata un ejemplo con tablas de decisión y con reglas generadas que pueden ser mucho más entendibles.

Factor	Regla 1	Regla 2	Regla 3	Regla 4	Regla 5	Regla 5
Gana equipo favorito	**S**	**S**	**S**	**S**	**N**	**N**
Amigos presentes	**S**	**S**	**N**	**N**	**S**	**N**
Salir de casa	**S**	**N**	**N**	**S**	**N**	**N**
Resultado	**Cerveza**	**Cerveza**	**Refresco**	**Refresco**	**Refresco**	**Leche**

Tabla de reglas de decisión en sus posibles combinaciones

Observar el juego y *gana el equipo favorito* y *sale con los amigos* **entonces** CERVEZA.
Observa el juego y *pierde el equipo favorito* y *sale con los amigos* **entonces** CERVEZA.

Observa el juego y *gana el equipo favorito* y *esta solo en casa* **entonces** REFRESCO.
Observa el juego y *pierde el equipo favorito* y *esta solo en casa* **entonces** LECHE.

Si se observa, la primer y segunda regla, pueden ser fusionadas por la siguiente regla:

Observa el juego y *sale con los amigos* **entonces** CERVEZA.

Esto significa hacer una generalización de algunas reglas, en donde algunas de sus cláusulas puedan ser eliminadas y sus clasificaciones sean similares.

Sin embargo, se pueden llegar a tener registros que no pertenezcan a ninguna regla, por lo que se tendrá que asignarles una por default, esta puede ser la que tenga mayor número de ocurrencias en la clasificación de registros.

Otro problema sería tener reglas que no son mutuamente excluyentes; lo cual se resuelve al eliminar una de las dos reglas, la que tenga el menor porcentaje de clasificación de registros.

Así, se obtiene finalmente un número de reglas mucho menor al inicial; sin embargo, se puede hacer una reducción mayor de estas, eliminando aquellas que no sean sustancialmente utilizadas para la clasificación de registros, es decir, aquellas que no tengan mucha incidencia de registros.

CHAID

Este es el algoritmo más antiguo de los tres que se presentan, es el más ampliamente utilizado en paquetes estadísticos (como SPSS y SAS). Es descendiente del Sistema de Detección de Interacción Automática (AID). Por lo que la motivación de CHAID es la detección de relaciones estadísticas entre variables.

Se diferencia en que los otros dos modelos (CART y C4.5) primeramente construyen el árbol sobre ajustado y después realizan una poda. CHAID detiene el crecimiento del árbol antes de que ocurra el sobre ajuste. CHAID se restringe a trabajar con variables categóricas, por lo que las variables continuas deben ser partidas en rangos o sustituidas por clasificaciones (por ejemplo *baja, media, alta*).

Para elegir un campo que particionará como primer nodo, se deben hacer grupos con los campos y así lograr tener los distintos valores que puede tomar esa variable. Estos registros deberán ser significativamente parecidos.

Esto se logra con la prueba de X^2, que es la raíz cuadrada de la suma del cuadrado de las restas de los valores en comparación de entre dos campos potenciales para ser particionadores. Donde un campo potencial para ser particionador, puede constar de más de un elemento a ser considerado.

Se reparticionan los campos, esto es, todos los campos predictores que no produjeron diferencias estadísticas significativas en los valores del campo objetivo, son mezclados. A continuación cada grupo de tres o más predictores es reparticionado por todas las posibles divisiones binarias, si alguno de esos campos ofrece una salida estadísticamente significativa, se conservará.

Para hacer una evaluación de los campos particionadores. Cada uno de los campos predictores ha sido agrupado para producir la máxima posible diversidad de clases del campo objetivo, la prueba de X^2 es aplicada a los grupos resultantes. El campo predictor que genere los grupos que difieran lo más posible de acuerdo a esta prueba, serán elegidos como particionadores para este nodo.

El crecimiento del árbol en este método, esta limitado hasta el punto en el que no se encuentren más particionadores disponibles que conduzcan a diferencias estadísticamente significativas en la clasificación. Este nivel preciso de crecimiento del árbol es el parámetro clave para la construcción en base a este método (CHAID).

¿Cómo es construido un árbol de decisión?

Se construye directamente por una técnica conocida como *particionamiento recursivo*. Es un proceso iterativo de partir los datos en un sentido de arriba hacia abajo, iniciando desde un nodo y terminando con una colección de varias decenas o centenas. Los registros preclasificados son usados para determinar la efectividad.

¿Cómo elegir la partición inicial?

El proceso inicia con un conjunto de entrenamiento, el cual está preclasificado. Esto indica que el campo objetivo o las variables dependientes, ya son conocidas.

La primer tarea es decidir cual de las variables independientes marca la mejor partición. La mejor partición es definida como aquella que hace el mejor trabajo de separar los registros en grupos donde una sola clase predomina. La medición usada para evaluar el potencial de la partición es la reducción de la diversidad.

El índice de diversidad es la *probabilidad de que la segunda cosa sea elegida cuando la primera cosa ya halla sido elegida y sea diferente*. Es conocida como la probabilidad condicional.

$$P(h \mid e) = \frac{P(h \cap e)}{P(e)} \text{; donde } h, e \subseteq \Omega \text{ y } P(e) \geq 0$$

Así para un ejemplo binario, la posibilidad de elegir dos elementos iguales dentro de una población es P(e)*P(e), y la probabilidad de elegir un elemento diferente en dos intentos es 1- ($P(e)^2$ + $P(h)^2$). El valor máximo posible de la diversidad es de $1/2$, dado que hay solo dos clases, o $1/n$ donde *n*, es el número de clases y cada clase tiene el mismo número de miembros y por lo tanto la misma probabilidad de que sean elegidos.

Así, la fórmula de la diversidad es para un índice binario:

$$2p_1(1-p_1)$$

Cuando no se le encuentra sentido a realizar una partición más al árbol, esto es, cuando la creación de un nodo no decrementa el índice de la diversidad, se deberá marcar como un *nodo hoja*, con lo que no se puede profundizar más.

¿Cuál es la característica que debe ser la que primero particione al árbol?

Deberá ser aquella que marque el índice de la diversidad al 50%, con lo que se dividen dos grandes ramas del árbol. Para niveles inferiores, se deberán elegir los elementos que particiones el árbol de tal forma en el que una rama tenga la mayor probabilidad, es decir, que decremente la diversidad lo mayormente posible.

Sus ventajas. Los árboles de decisión están habilitados para generar reglas entendibles, fáciles de "traducir" al idioma en el que las personas nos comunicamos o en SQL.

Los árboles de decisión mejoran el rendimiento de la clasificación sin requerir mucho poder de cómputo.

Los árboles de decisión soportan el manejo de variables categóricas o continuas.

Los árboles de decisión indican claramente cual de los campos es más importante para una clasificación o predicción; esto es porque los campos particionadores más cercanos al nodo raíz, son los campos más característicos para el estudio realizado, es decir, son los más determinantes.

Sus desventajas. Los árboles de decisión son menos apropiados para tareas de estimación donde la meta es predecir el valor de variables continuas.

Los árboles de decisión son problemáticos para datos de series de tiempo, a menos que un gran esfuerzo sea puesto en la preparación de datos, de tal forma que las tendencias y patrones sean visibles.

Existen restricciones en algunos algoritmos, ya que solo pueden manejar valores binarios (si / no, aceptar / rechazar), pero para los casos en los que pueden obtenerse muchos nodos hijos, existe una tendencia a obtener una tasa de error alta por el crecimiento desmesurado del árbol, costando muchos recursos de cómputo.

El usar un solo campo a la vez para clasificar, lleva a definición de regiones rectangulares, lo que puede no corresponder a la definición real de los registros.

REDES NEURONALES ARTIFICIALES (ARTIFICIAL NEURAL NETWORKS)

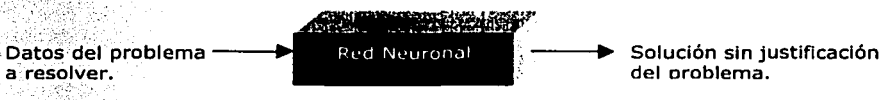
Las redes neuronales artificiales son famosas por su poder de predicción, clasificación y creación de clusters, por lo que es una herramienta muy poderosa. Han sido aplicadas en la industria para predecir series financieras, así como en condiciones medicas, también desde identificación de clusters de clientes y hasta para identificar transacciones fraudulentas en tarjetas de crédito y también para predecir la tasa de fallas en motores de automóviles recién salidos de la fabrica.

Las redes neuronales (siendo un buen modelo y perfectamente bien entrenado) tienen la habilidad de poder generalizar y aprender sobre su experiencia.

Esta habilidad es utilizada en la minería de datos, ya que realizan búsquedas exactas y cada vez son mejores.

Una situación relevante en las redes neuronales, es que no se tiene una explicación del porque la toma de una cierta acción, solo se conocerá que así es obtenido el resultado. Esto es un problema cuando se requiere de realizar una justificación de un cierto resultado. Siendo esto en algunos casos un punto crítico para poder realizar una cierta actividad.

Normalmente se le conocen a las redes neuronales como una caja negra que solo arroja resultados sin justificar el porque de los mismos.



Toda la historia inicia en los años 40, cuando un neuro-psicólogo, Warren McCulloch, y un logístico, Walter Pits, popularizaron un modelo simple de cómo trabaja una neurona biológica.

Aunque la intención de estos dos personajes fue de explicar un funcionamiento fisiológico, se iniciaron una serie de estudios que echarían mano de este conocimiento para resolver problemas utilizando este nuevo esquema.

Al iniciar la era de las computadoras digitales en los años 50, se implementaron modelos llamados PERCEPTRONES basados en el trabajo de McCulloc y Pits. Durante muchos años, los siguientes 30 años, se continuaron buscando mejoras para la solución de las deficiencias existentes del modelo de las redes neuronales. Siendo a finales de los 80 cuando se pudo lograr un avance y un comportamiento de las redes neuronales mucho más aceptable.

Finalmente, parte de la popularidad lograda en estos años, fue porque el poder de cómputo estaba más al alcance de la comunidad de negocios (que es donde se encuentra toda la información), además de que se les pudo ayudar a las redes neuronales con algunos métodos de estadística, conocido como regresión logística, siendo este un método cercano a la regresión lineal, comúnmente considerado como un caso especial de las redes

neuronales. También, cada vez más y más compañías requieren automatizar algunos de sus procesos, por lo que las redes neuronales representan una buena opción y puede llegar a ser algo cercano a la inteligencia artificial; siendo considerada como una de las herramientas más poderosas en un futuro cercano.

El aprendizaje de las redes neuronales es muy parecido a la habilidad de aprender a través de ejemplos como los humanos lo hacen.

VALORACIÓN DEL ESTADO REAL

Una ventaja de las redes neuronales, es que pueden aprender conforme a los ejemplos presentados en una forma mucho muy parecida a la de los seres humanos, esto es, ganando una cierta experiencia.

Si bien es una actividad normal y cotidiana hacer en la vida humana ciertas valoraciones, estimaciones sobre el valor de objetos, esto solo se puede realizar cuando se tiene un conocimiento de muchas cosas ajenas al mismo objeto, pero que son inherentes para poder tener todo el concepto del valor real de ese objeto en curso.

Al intentar hacer una valoración o estimación sobre algún objeto en específico, no solo con tomar en cuenta sus propiedades es suficiente.

Por ejemplo, para considerar el precio de una casa habitacional; no solo se deben saber precios de cada uno de los elementos que la componen, sino también de todos aquellos que la rodean, como la colonia a la que pertenece, los servicios con los que se cuenta, si es una zona comercial o residencial, el estilo de la decoración y hasta el valor sentimental para el sueño, entre otras. Siendo cada una de las anteriores totalmente determinantes para influir en el precio de la vivienda.

Después de haber considerado todas las particularidades anteriormente mencionadas, solo se debe obtener una salida: un precio expresado en dinero.

Para que las redes neuronales puedan hacer un buen trabajo de predicción, el problema debe tener tres características:

- *Las entradas a considerar, deben estar perfectamente bien entendidas.* Se debe tener una idea muy clara de las características que son importantes a ser consideradas, pero no necesariamente hacer una combinación entre ellas, es decir, no realizar las posibles rutas.
- *La salida, debe estar perfectamente entendida.* Se debe saber que es lo que se está intentando predecir, para poder tener una idea de los posibles resultados.
- *La experiencia esta disponible.* Se debe practicar con ejemplos que se tengan las entradas y las salidas conocidas, para saber entrenar a la red neuronal y saber cuando esta bien o mal la predicción.

Para hacer el entrenamiento de la red neuronal se realizaría con ejemplos de ventas de casas habitacionales ya hechas, indicándole a la red neuronal a parte de todas

las características antes mencionadas, el precio total de la venta y la fecha en que se realizó la venta.

Se recomienda además, que los valores estén puestos como variables categóricas o continuas, esto facilitará la ubicación de sí estos valores están cercanos a los reales o no. Siendo una opción de mejora cuando estos valores están entre 0 y 1, lo cual requiere encontrar las máximas diferencias entre los valores, así como la normalización de cada uno de los registros.

Para las variables categóricas se deben de insertar los valores también entre 0 y 1, por ejemplo, si solo hay dos valores A y B, uno de ellos será 0 y el otro 1, sin embargo si hay tres valores (A, B y C), se dividirán en 0, 0.5 y 1, según las características del problema, para que el valor intermedio sea al que se le asigne 0.5.

Después de entrenar a la red neuronal con ejemplos cada una de las ocasiones completamente diferentes, esto para poder garantizar que esta aprendiendo lo más posible y no solo esta realizando combinaciones sobre el mismo ejemplo; se debe tener el conjunto de prueba también completamente diferente al de entrenamiento.

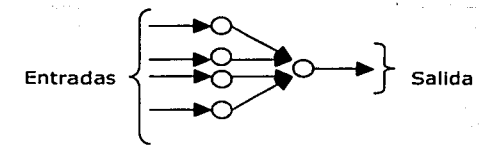
Finalmente, se pueden resumir los pasos para construir un modelo de predicción o clasificación:

1. Identificar las características de las entradas y las salidas.
2. Manipular las entradas y salidas para trabajarlas en un rango de 0 a 1.
3. Construir una topología adecuada de red.
4. Entrenar a la red con un conjunto adecuado de ejemplos representativos.
5. Probar la red con un conjunto estrictamente independiente del conjunto de entrenamiento.
6. Aplicar el modelo generado por la red para predecir resultados a partir de entradas desconocidas.

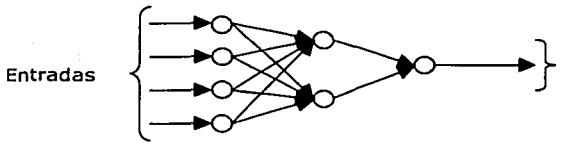
Cuando las condiciones sobre el conjunto de entrenamiento o de prueba han cambiado drásticamente o simplemente hace mucho tiempo que no se da un entrenamiento, se recomienda eliminar toda la configuración anterior, para evitar que puedan llegar a obtenerse resultados incorrectos. Aunque lo mejor es nunca abandonar el modelo tanto tiempo como para que no se pueda corregir cualquier error con un conjunto de registros de entrenamiento.

ESTRUCTURA DE UNA RED NEURONAL

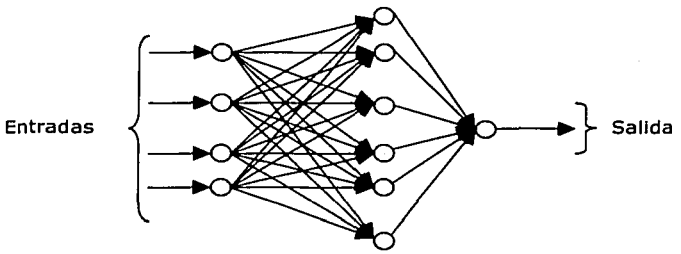
Una red neuronal consiste en unidades básicas modeladas en neuronas biológicas. Cada unidad tiene múltiples entradas que se combinan en una única salida. Estas unidades están interconectadas. En los ejemplos siguientes, se muestran combinaciones de redes donde no hay ciclos de retroalimentación de la red de forma interna. Es decir, la información solo fluye en un solo sentido.



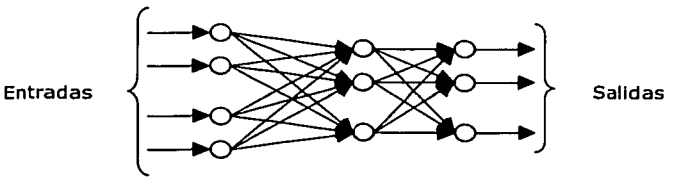
Esta es una red neuronal muy simple, los resultados arrojados son equivalentes a los de la técnica estadística regresión logística.



Esta es una red con una capa intermedia llamada *capa superior*. Esta capa la hace mucho más potente para reconocimiento de patrones.



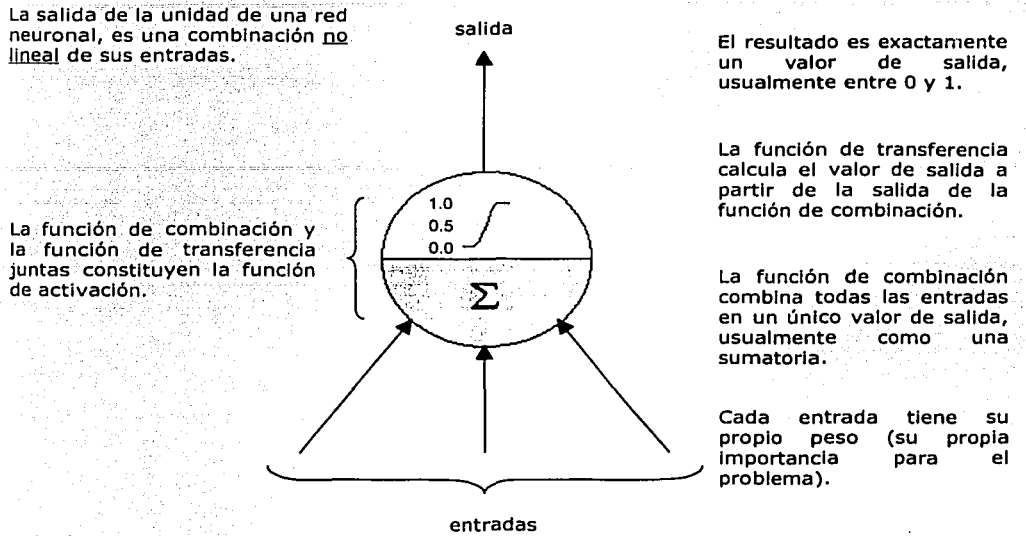
Incrementando el tamaño de la capa superior se hace una red aún más poderosa, pero se corre el riesgo de hacer un sobre ajuste. Con una capa superior normalmente es suficiente.



Una red neronal que produce múltiples valores como salidas.

¿Qué es la unidad de una red neuronal?

Como se ha mencionado anteriormente, la red neuronal artificial se compone por unidades básicas diseñadas para modelar el comportamiento de las neuronas biológicas (como lo muestra la siguiente figura).
Esta unidad combina las entradas en un único valor de salida.



La combinación realizada dentro de la unidad de la red neuronal se llama la función de activación de la unidad, dentro de la cual se puede tener un umbral de activación.

Es obvio que cuando existen variantes en las entradas, la salida se verá afectada en la medida en la que la función de activación dependa de ese valor de entrada; esto es debido a que *tiene un comportamiento no lineal.*

La función de activación se divide en dos partes:

- La función de combinación, la cual mezcla todas las entradas en un solo valor (parte sombreada del círculo de la unidad de la red neuronal artificial).
- La función de transferencia, la cual tiene ese nombre por el hecho que transfiere el valor de la función de combinación a la salida de la unidad.

Los valores que toma la función en un momento específico no son tan importantes como la esencia de la misma función.

Las redes neuronales hacen un buen trabajo de predicción en tres tipos de problemas:

- Problemas lineales.
- Problemas casi lineales.
- Problemas no lineales.

Una red puede contener unidades con diferentes funciones de transferencia, la más común es para problemas no lineales.

La función más común para problemas no lineales es la *función Sigmoid* la cual produce valores en el rango entre 0 y 1 para todas las posibles sumatorias. La formula es:

$$Sigmoid(x) = \frac{1}{1+e^{-x}}$$

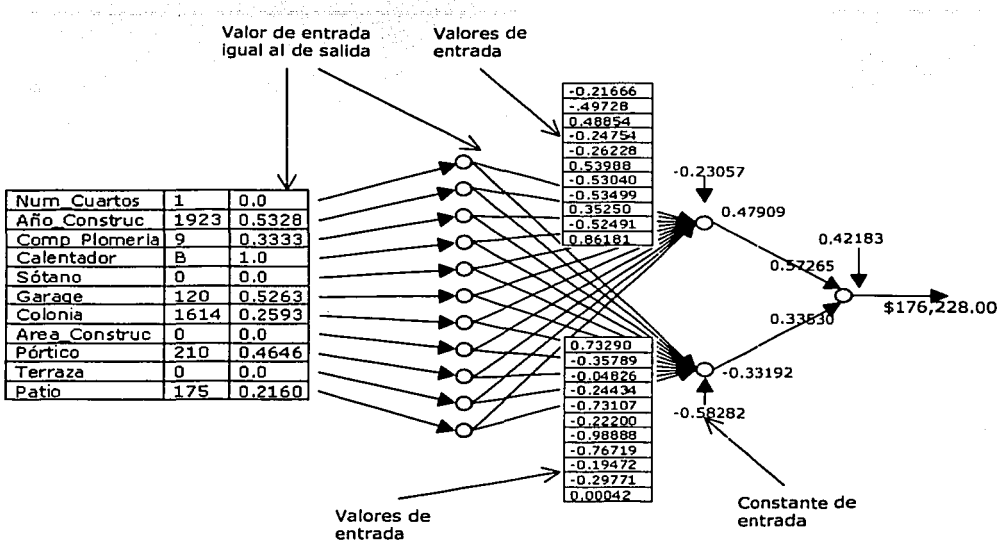
Cuando es usada esta función en una red neuronal, la x es el resultado de la función de combinación, típicamente la suma de las entradas de la unidad.

La siguiente tabla muestra una serie de características que se describen en la valoración de la compra de una casa, por lo que el ejemplo se mostrará con caso real.

Característica	***Descripción***	***Valores***
Número_cuartos	Número de recamaras	1-3
Año_Construc	Año de construcción	1850-1986
Comp_plomería	Número de composturas de plomería	5-17
Calentador	Tipo de calentador	A – B
Sótano	Número de autos que caben	0-2
Garaje	Área para guardar cosas	0-228 (m^2)
Colonia	Número de la colonia a la que pertenece	0-738(m^2)
Área_Construc	Área total de construcción en la casa	714-4185(m^2)
Pórtico	Área del pórtico	0-452(m^2)
Terraza	Área de la terraza	0-672(m^2)
Patio	Área del patio	0-810(m^2)

Característica	***Rango de valores***	***Valor original***	***Valor manipulado***
Precio_de_venta	$103,000-$250,000	$171,000	0.4626
Meses_de_la_venta	0-23	4	0.1739
Número_cuartos	1-3	1	0.0000
Año_Construc	1850-1986	1923	0.5328
Comp_plomería	5-17	9	0.3333
Calentador	A – B	B	1.0000
Sótano	0-2	0	0.0000
Garaje	0-228	120	0.5263
Área_Construc	714-4185	1614	0.2593
Colonia	0-738	0	0.0000
Pórtico	0-452	210	0.4646
Terraza	0-672	0	0.0000
Patio	0-810	175	0.2160

Red neuronal que por medio de una topología típica para predicción o clasificación y con tres capas, calcula el valor para la venta de una casa. La capa de la izquierda es conectada a las entradas, mientras que los valores son manipulados para estar en el rango de 0 y 1. Esta es la *capa de entradas* de la red.



Este es un ejemplo de una red neuronal que hace una estimación del costo de venta de una casa según sus caracteristicas. El resultado fue ligeramente superior al real por un 3%.

Las unidades en las capas ocultas, siempre están interconectadas con las demás unidades de las otras capas (ocultas o no). Dentro de una red neuronal, con frecuencia es suficiente con tener una capa oculta. La salida de cada unidad, es la suma de la multiplicación de cada entrada por el peso que tiene asignado esa entrada y aplicándole la función Sigmoid.

Cuando se hacen las correcciones sobre una red al hacer pruebas con un conjunto de entrenamiento, se desea que el error se minimice, más no que desaparezca, ya que al minimizarlo se logra hacer una generalización, siendo lo ideal y cuando se logra un error nulo, la red neuronal no hizo una generalización sino simplemente memorizó el conjunto de entrenamiento, lo cual repercutirá en que fallará con cualquier otro conjunto de datos.

Es en ese momento cuando se para de entrenar a la red neuronal y se comienza a trabajar con entradas nunca antes vistas para el modelo.

Dentro de una red neuronal, se puede hacer un aprendizaje con retro-propagación. La parte central de la retro-propagación es:

1. La red recibe un ejemplo de entrenamiento y, usando los pesos existentes en la red, se calcula(n) la(s) salida(s) del ejemplo.
2. La retro-propagación calcula el error, tomando la diferencia entre el resultado calculado y el esperado.

3. El error es retro-alimentado por la red y los pesos son ajustados para minimizar el error.

El algoritmo de la retro-propagación se puede definir de la siguiente forma:

1. Propaga las entradas a través de la red y calcula la salida
2. Propaga el error hacia atrás
   a. Para cada unidad de salida k, calcula su error (δ_k)

$$\delta k \longleftarrow o_k(1-o_k)(t_k-o_k)$$

   b. Para cada unidad oculta h, calcula su error(δ_h)

$$\delta_h \longleftarrow o_h(1-o_h)\sum_{k\in sal(h)} w_{hk}\delta_k$$

   c. Actualiza los pesos w_{ij}

$$w_{ij} \longleftarrow w_{ij}+\Delta w_{ij} \qquad \text{donde} \qquad \Delta w_{ij}=\alpha\delta_j x_{ij}$$

Donde cada símbolo representa:

- o_j = la salida del nodo j
- t_j = la salida esperada del nodo j
- α = razón de aprendizaje
- w_{ij} = el peso asociado a la i-ésima entrada al nodo j
- x_{ij} = la i-ésima entrada al nodo j
- sal = el conjunto de nodos de salida

¿Cómo se puede hacer que el error se minimice? Simplemente, se inicia una serie de pruebas que consisten en cambiar los valores de cada una de las entradas del modelo de la red neuronal, siendo cada uno de estos cambios sensiblemente diferentes, según la dependencia del modelo a esa entrada. Esto finalmente será diferente en cada red neuronal.

Esta técnica de ajustar los pesos de las entradas en la red se llama *regla delta de generalización.*

La *regla delta de generalización* tiene dos puntos críticos para considerar:

- El momentum, el cual se refiere a la tendencia que tienen los pesos de forma interna en cada unidad para cambiar la dirección que están tomando dentro del modelo. Esto es, que cada peso recuerda si había sido grande o chico, y, el momentum intenta dar el mismo sentido que había tenido ese peso. Cuando el momentum es pequeño, entonces los valores de los pesos pueden oscilar más libremente.

- La tasa de aprendizaje, controla cuan rápido pueden cambiar de valor los pesos. El mejor aprovechamiento es iniciar con un valor grande y decrementarlo lentamente durante el entrenamiento de la red. Inicialmente los valores son aleatorios.

Al tener el momentum y la tasa de aprendizaje controlados, se obtienen los mejores resultados.

Un punto crítico en la construcción de las redes neuronales es el número de unidades en las capas ocultas. Los riesgos que se pueden tener con un número grande de unidades, es un sobre entrenamiento y por lo tanto memorice el conjunto de entrenamiento (más no lo generalice y comprenda), por lo que "más" unidades ocultas no son mejores en este caso. Esto se puede visualizar rápidamente, cuando en el conjunto de entrenamiento se hace muy bien (o hasta perfecto) el trabajo de la red neuronal, pero en el conjunto de prueba se obtienen resultados muy malos.

Entonces, ¿Cuál debe ser el tamaño de la capa oculta? Una regla empírica dice que nunca debería ser mayor a dos veces el número de unidades de la capa de entrada. Se recomienda iniciar con un número de unidades igual al de la capa de entrada. Si esta sobre entrenada, se reduce el número de unidades, si no hace bien el trabajo, se aumentan el número de unidades sin exceder la regla $\text{longitud\_capa\_oculta} \leq \text{unidades\_capa\_de\_entrada} * 2$. Si se esta buscando hacer una clasificación, se debe iniciar con una unidad oculta por cada clase que se tenga.

Otro punto a considerar para el tamaño, es el tamaño del conjunto de entrenamiento. Esto es: se tiene una red con s unidades de entrada, con h unidades ocultas y 1 salida, por lo que se tiene $h*(s+1)+h+1$ pesos en la red. Por ejemplo, se tienen 15 entradas y 10 unidades ocultas en la red, por lo que se tendrán 171 pesos en la red. Más aún, si se tienen 5 ejemplos por cada peso, entonces se requiere que el conjunto de entrenamiento deba tener 810 ejemplos.

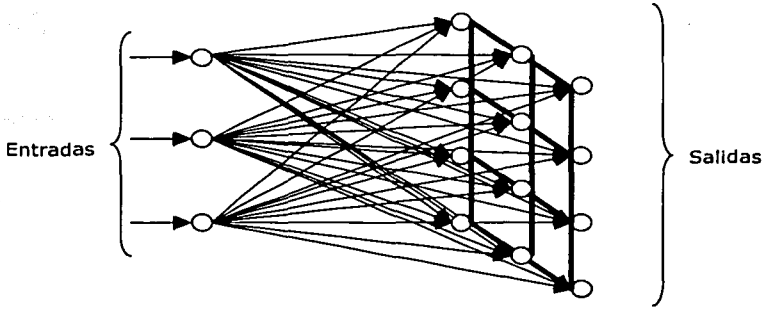
¿Cómo saber que tan sensitiva es una red ante un dato de entrada específico? Se debe tener primeramente el valor promedio de esa característica a analizar, el valor mínimo así como el máximo. Se hace una prueba con la red neuronal variando cada uno de esos valores de cada una de las entradas, pero una sola variación de una sola entrada a la vez. Si el resultado final de la red, se afecta al variar el valor de una entrada, entonces la red es *sensitiva* a esa entrada, de lo contrario es *no sensitiva*.

Finalmente, una prueba de sensibilidad aún mejor es cuando se hacen variar un conjunto de características al mismo tiempo. Esto para ver como se afectan los resultados según ese conjunto de características que han variado.

MAPAS AUTO-ORGANIZADOS

Una especie de redes neuronales ligeramente diferente son las llamadas ***mapas de auto-organización*** (***SOM,*** conocido así por sus siglas en inglés), los cuales son un tipo

en particular de redes neuronales. Estas redes cuentan con una capa de salida con muchas más unidades que las redes neuronales de retro-propagación.



Mapa de auto-organización, utilizado para detección de clusters.

Lo que hace particular este tipo de distribución de unidades, es que se enfoca al reconocimiento de imágenes, por lo que trabaja muy bien en dos dimensiones.

La capa de salida tradicionalmente cuenta con una interconexión de todos sus elementos para dar un aspecto de malla, esto es para dar mayor estabilidad al modelo. Se utilizan para poder encontrar clusters, esto es, cuando se obtienen los resultados de cada una de las salidas, se pueden obtener resultados, pero aún no se sabe nada sobre el problema, por lo que la correcta integración de clusters depende del entendimiento de las circunstancias del problema.

ELECCIÓN DEL CONJUNTO DE DATOS PARA EL ENTRENAMIENTO DE LA RED NEURONAL

Este conjunto de datos consiste en registros con valores de predicción o clasificación ya conocidos. La elección correcta de este conjunto es un punto crítico del éxito de la red neuronal. Las características que debe cumplir este conjunto de datos para el entrenamiento de la red, son las siguientes:

- Cubrir los valores para todas las características. El conjunto de entrenamiento debe cubrir el rango total de valores para todas las características que la red neuronal puede encontrar. Además, los valores de entrada de la red neuronal deben estar entre 0 y 1, por lo que previamente fueron manipulados y encontrado el rango en el que puede oscilar ese valor. Una recomendación que se debe tomar muy en cuenta, es que se debe mantener un cierto rango de holgura para poder trabajar con mayor precisión, esto es, al manipular los datos en un rango de 0 y 1, el valor mínimo tendrá el rango de 0, por lo que un valor inferior queda totalmente

inaccesible para la red neuronal, por lo que se recomienda trabajar en un rango de 0.1 y 0.9.

- Número de características. El tiempo necesario para entrenar una red neuronal, esta directamente relacionado con el número de características de entrada. Se recomienda descartar manualmente las características irrelevantes para la solución del problema y así aumentar la velocidad de convergencia de la red. Una de las formas de poder determinar cuales son las características más importantes es por medio de la construcción de árboles de decisión.
- El número de entradas. Mientras mayor sea el número de características a analizar en la red, también tendrá que ser mayor el número de entradas, esto para darle una buena cobertura a los patrones. Para estas consideraciones no existen fórmulas, sin embargo una buena aproximación es tener de 10 a 20 entradas por cada característica del modelo.
- El número de salidas. Siempre es mayor el número de entradas que el de salidas, esto es porque se necesitan cubrir cualquier posibilidad de valores de salida. Lo más adecuado es surtir al modelo, con un 50% de buenos ejemplos, del total de datos, y, con un 50% de ejemplos malos. Consiguiendo con esto, que sean cubiertos los casos raros (aislados, como fraudes o piezas defectuosas) y los casos comunes (normales, clientes normales).
- Poder de cómputo disponible. El mayor reto del modelo de las redes neuronales, es lograr la convergencia del mismo en el menor tiempo posible, ya que en muchas ocasiones requiere de pasar el conjunto de datos docenas o cientos de veces antes de quedar liberado el modelo.

La obtención de valores en un rango entre 0 y 1, se realiza de la siguiente forma:

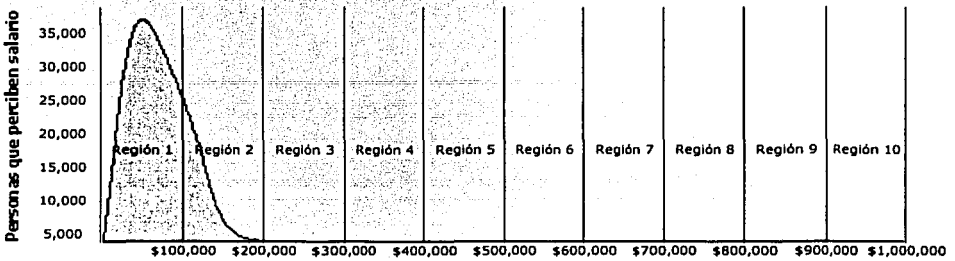
$$valor\_manipulado = \frac{valor\_actual - valor\_\min}{valor\_\max - valor\_\min}$$

donde, *valor_manipulado* es el valor resultante entre 0 y 1; *valor_actual* es el valor que se desea cambiar al rango entre 0 y 1, *valor_min* es el valor más pequeño de todos los encontrados en el conjunto de entrenamiento (no es un valor entre 0 y 1) y *valor_max* es el valor más grande de todos los encontrados en el conjunto de entrenamiento (no es un valor entre 0 y 1).

Para obtener un valor manipulado con una holgura del 10%, es decir entre 0.1 y 0.9, es de la siguiente manera:

$$valor\_manipulado\_holgura = \frac{(0.8)(valor\_actual - valor\_\min)}{valor\_\max - valor\_\min}$$

Un problema que se puede presentar, es cuando los valores están sesgados, esto es, que el rango esta muy amplio y solo se utiliza una pequeña fracción del mismo.



La solución para este problema, es utilizar una escala logarítmica, con lo que se logra reducir la gama tan amplia de valores y se continúa con la consistencia de los mismos datos sin tener que perder precisión.

Los valores estarían localizados de la siguiente manera: log(10) = 1, log(100) = 2, log(1000) = 3, ya que representa la potencia a la que están elevadas estas cifras. La tabla entonces quedaría de la siguiente forma:

Entrada	Entrada manipulada	Log(Entrada)	Logaritmo manipulado
$10,000	0.0101	4.0000	0.0000
$18,000	0.0182	4.2553	0.1276
$32,000	0.0323	4.5051	0.2526
$63,000	0.0636	4.7993	0.3997
$100,000	0.1010	5.0000	0.5000
$250,000	0.2525	5.3979	0.6990
$800,000	0.8081	5.9031	0.9515
$1,000,000	1.0101	6.0000	1.0000

Por supuesto, debe estar en escala del 0 al 1 y se puede tratar como cualquier otro valor, obtenido desde cualquier otra forma.

INTERPRETACIÓN DE LOS RESULTADOS

Como se ha visto, los valores que se utilizan en las redes neuronales deben estar manipulados, es decir, comprendidos en un rango de entre 0 y 1, por lo que los resultados también deben ser convertidos a la inversa, esto para que tengan una interpretación real.

Para un ejemplo en el que la red neuronal clasifica elementos en dos clases.

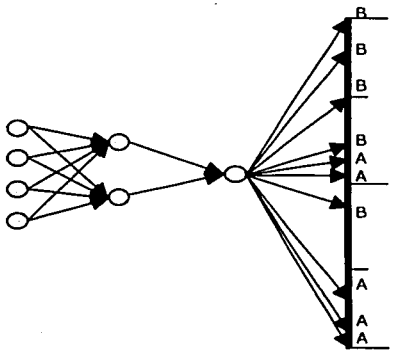
Una de las formas de poder clasificar las salidas de la red neuronal, es diciendo que todos aquellos valores predichos por debajo del 0.5, pertenecen a una categoría, y así los que están por arriba del 0.5 pertenecen a otra categoría.

Un ejemplo sería la siguiente tabla, en la que se muestra las salidas, el valor obtenido y su confianza:

Valor de salida	Categoría	Confianza
0.0	A	100%
0.2	A	80%
0.49	A	51%
0.51	B	51%
0.8	B	80%
1.0	B	100%
0.63	A	37%

En el caso del último elemento, que coincide dentro del rango de la clase B, seria recomendable poder trabajar una apertura del rango entre las dos clases, es decir poder decir que el punto de partida inicia en 0.67 y no en 0.5, con lo que se logra un mejor ajuste, según los valores reales (no ideales).

Dado que las salidas de la red neuronal son continuas, es difícil poder convertirla en variable categórica, para lo cual el umbral que se coloque, deberá ser en el punto que se determine a partir del conjunto de datos de prueba.



La figura de la izquierda, muestra la situación antes descrita, en la que se debe contemplar la posibilidad de mover el centro que separa las clasificaciones de las clases de la red neuronal, si es que estos casos son como el que se muestra.

Sus ventajas. En una red neuronal, se pueden manejar grandes volúmenes de datos y producir buenos resultados en dominios complicados, esto se debe a su carácter no lineal, dando una serie de posibilidades aún mayor.

Puede trabajar con variables continuas y categóricas. Además, es ampliamente soportada esta técnica en muchas herramientas computacionales.

Sus desventajas. Se requiere de una manipulación de los datos de entrada y salida, y deben estar en un rango de entre 0 y 1, para poder dar una facilidad de uso en la red.

No tiene forma de sustentar sus resultados, debido a que una red neuronal es una "caja negra".

Se pueden obtener resultados incorrectos por un prematuro entrenamiento de la red neuronal, siendo un punto imprescindible la elección de un correcto conjunto de entrenamiento, sin sobre entrenar a la red.

ALGORITMOS GENETICOS (GENETIC ALGORITMS)

ANTECEDENTES

La evolución se produce como resultado de dos procesos primarios: la selección natural y la reproducción sexual. La primera determina que miembros de la población sobrevivirán hasta reproducirse, la segunda garantiza la mezcla y combinación de sus genes entre la descendencia. En la fusión del óvulo y el espermatozoide, los cromosomas homologados se estiran y adosan uno al otro, y luego se entrecruzan en zonas intermedias, intercambiando así material genético.

En la naturaleza, los individuos compiten entre sí por recursos tales como comida, agua, refugio. Adicionalmente, los animales de la misma especie normalmente antagonizan para obtener una pareja.

Esta es la teoría de la evolución, especies naturales que van evolucionando para adaptarse al medio que las rodea; aquellos individuos que tenga más éxito en tal adaptación tendrán mejor probabilidad de sobrevivir hasta la edad adulta y probablemente un número mayor de descendientes, por lo tanto, mayores probabilidades de que sus genes sean propagados a lo largo de sucesivas generaciones.

La combinación de características de los padres bien adaptados, en un descendiente, puede producir muchas veces un nuevo individuo mucho mejor adaptado que cualquiera de sus padres a las características de su medio ambiente.

Este proceso no debe verse en ningún momento como un proceso determinista, sino como un proceso con una fuerte componente estocástica. Es decir, si un individuo se adapta al entorno, lo más que se puede afirmar es que ese individuo tendrá mayor probabilidad de conservar sus genes en la siguiente generación que sus congéneres. Pero solo es una probabilidad, no es un hecho absolutamente seguro. Siempre existirá la posibilidad de que a pesar de estar muy dotado por alguna razón no consiga reproducirse. Pero en cuanto a la especie como un conjunto o población, si puede afirmarse que irá adaptándose al medio.

ALGORTIMOS GENÉTICOS

La idea de aplicar estos conceptos a problemas computacionales, surgió en la universidad de Michigan, Estados Unidos donde el profesor J. H. Holland, consciente de la importancia de la selección natural introdujo la idea de los Algoritmos Genéticos en los años sesenta y al final de esta década desarrollo una técnica que permitió incorporarla en un programa de computadora. Su principal objetivo era lograr que las computadoras aprendieran por sí mismas. A la técnica inventada por Holland se le llamo inicialmente Planes Reproductivos pero se hizo popular bajo el nombre de Algoritmos Genéticos, A.G.

Obviamente desde aquellos años sesenta hasta ahora, muchas otras personas han contribuido de modo notable al desarrollo de estas ideas, abriéndose muchos nuevos frentes de trabajo y subdividiéndose la idea original en múltiples disciplinas. Estas técnicas se usan principalmente en países desarrollados como Japón, Estados Unidos y en Europa.

Los organismos vivos poseen destreza consumada en la resolución de problemas. Una definición bastante completa de un Algoritmo Genético es la propuesta por Jhon Kosa:

> "Es un Algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual.
>
> Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se asocian con una cierta función matemática que refleja su aptitud.
>
> Los Algoritmos Genéticos utilizan una analogía dirigida del fenómeno de evolución en la naturaleza. Trabajan con una población de individuos, cada uno representando una posible solución a un problema dado.
>
> A cada individuo se le asigna una puntuación de adaptación, dependiendo de que tan buena fue la respuesta al problema. A los más adaptados se les da la oportunidad de reproducirse mediante cruzamientos con otros individuos de la población, produciendo descendientes con características de ambos padres. Los miembros menos adaptados poseen pocas probabilidades de que sean seleccionados para la reproducción, y desaparecen. El evaluar esta adaptación no es sencillo de hacer, pues el entorno está modificándose constantemente por lo que nunca se llegara al súper individuo perfecto, sino que la naturaleza tenderá a optimizar los individuos de cada especie en las circunstancias actuales".

CLASES DE ALGORITMOS GENÉTICOS

Existen varios tipos de Algoritmos Genéticos, cada uno basado en una metáfora distinta de la naturaleza.

1. Algoritmos Genéticos Generacionales

Se asemejan a la forma de reproducción de los insectos, donde una generación pone huevos, se aleja geográficamente o muere y es substituida por una nueva. En este momento se realizan cruces en una piscina de individuos, los descendientes son puestos en otra, al final de la fase reproductiva se elimina la generación anterior y se pasa a utilizar la nueva. Este modelo también es conocido como Algoritmo Genético Canónico.

2. Algoritmos Genéticos de estado Fijo

Utilizan el esquema generacional de los mamíferos y otros animales de vida larga, donde coexisten padres y sus descendientes, permitiendo que los hijos sean educados por sus progenitores, pero también que a la larga se genere competencia entre ellos.

En este modelo, no solo se deben seleccionar los dos individuos a ser padres, si no también cuales de la población anterior serán eliminados, para dar espacio a los descendientes.

La diferencia esencial entre el reemplazo generacional y el modelo de estado fijo es que las estadísticas de la población son recalculadas luego de cada cruce y los nuevos descendientes están disponibles inmediatamente para la reproducción. Esto permite al modelo utilizar las características de un individuo prometedor tan pronto como es creado.

Algunos autores dicen que este modelo tiende a evolucionar mucho más rápido que el modelo generacional, sin embargo investigaciones de Goldberg, encontraron que las ventajas parecen estar relacionadas con la alta tasa de crecimiento inicial; ellos dicen que los mismos efectos pueden ser obtenidos en rangos de adaptación exponencial o selección por competencia. No encontraron evidencia que este modelo sea mejor que el Generacional.

3. Algoritmos Genéticos Paralelos

Parte de la metáfora biológica que motivó a utilizar la búsqueda genética consiste en que es inherentemente paralela, donde al evolucionar se recorren simultáneamente muchas soluciones, cada una representada por un individuo de la población. Sin embargo, es muy común en la naturaleza que no solo sea una población evolucionando, sino varias poblaciones, normalmente aisladas geográficamente, que originan respuestas diferentes a la presión evolutiva. Esto origina dos modelos que toman en cuenta esta variación, y utilizan no una población como los anteriores sino múltiples concurrentemente.

a. *Modelos de Islas*

Si se tiene una población de individuos, esta se divide en subpoblaciones que evolucionan independientemente como un Algoritmo Genético normal.

Ocasionalmente, se producen migraciones entre ellas, permitiéndoles intercambiar material genético.

Con la utilización de la migración, este modelo puede explotar las diferencias en las subpoblaciones; esta variación representa una fuente de diversidad genética. Sin embargo, si un número de individuos emigran en cada generación, ocurre una mezcla global y se eliminan las diferencias locales, y si la migración es infrecuente, es probable que se produzca convergencia prematura en las subpoblaciones.

b. *Modelo Celular*

Coloca cada individuo en una matriz, donde cada uno sólo podrá buscar reproducirse con los individuos que tenga a su alrededor (más cerca de casa) escogiendo al azar o al mejor adaptado. El descendiente pasará a ocupar una posición cercana.

No hay islas en este modelo, pero hay efectos potenciales similares. Asumiendo que el cruce esta restringido a individuos adyacentes, dos individuos separados por

20 espacios están tan aislados como si estuvieran en dos islas, este tipo de separación es conocido como aislamiento por distancia.
Luego de la primera evaluación, los individuos están todavía distribuidos al azar sobre la matriz. Posteriormente, empiezan a emerger zonas como cromosomas y adaptaciones semejantes. La reproducción y selección local crea tendencias evolutivas aisladas, luego de varias generaciones, la competencia local resultará en grupos más grandes de individuos semejantes.

ELEMENTOS DE UN ALGORITMO GENÉTICO

Como los Algoritmos Genéticos se encuentran basados en los procesos de evolución de los seres vivos, casi todos sus conceptos se basan en conceptos de biología y genética que son fáciles de comprender.

- Individuo
  Un individuo es un ser que caracteriza su propia especie. El individuo se representa mediante un cromosoma y es el código de información sobre el cual opera el algoritmo. Cada solución parcial del problema a optimizar está codificada en forma de cadena o String en un alfabeto determinado, que puede ser binario. Una cadena representa a un cromosoma, por lo tanto también a un individuo y cada posición de la cadena representa a un gen. Esto significa que el algoritmo trabaja con una codificación de los parámetros y no con los parámetros en si mismos.

- Genotipo
  El genotipo, es el conjunto de genes ordenados y representa las características del individuo. Cada individuo tiene una medida de su adecuación como solución al problema.

- Población
  A un conjunto de individuos (Cromosomas) se le denomina población. El método de A.G´s consiste en ir obteniendo de forma sucesiva distintas poblaciones. Un Algoritmo Genético trabaja con un conjunto de puntos representativos de diferentes zonas del espacio de búsqueda.

- Función Adaptación (Fitness)
  Para usar un algoritmo genético es necesario que exista una función llamada fitness (adecuación o adaptación), que para determinar cuan bueno es un individuo dado para la solución de un problema. Esta función fitness o de evaluación es el principal enlace entre el Algoritmo Genético y un problema real; debe procurarse que la función fitness sea similar, si no igual a la función objetivo que se quiere optimizar. Esta medida se

utiliza como parámetro de los operadores genéticos, y guía la obtención de nuevas poblaciones.

OPERADORES GENÉTICOS

Son los diferentes métodos u operaciones que se pueden ejercer sobre una población y que nos permite obtener poblaciones nuevas. Una vez que se ha evaluado cada individuo sobre una función fitness, se aplican los operadores genéticos. En Algoritmos Genéticos se destacan los siguientes operadores.

OPERADOR DE SELECCIÓN

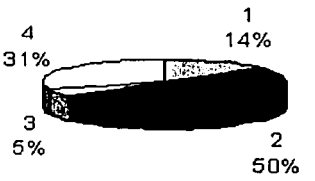
El paso siguiente a la selección del tipo de algoritmo genético, es escoger los miembros de la población que serán utilizados para la reproducción. Su meta es dar más oportunidades de selección a los miembros más aptos de la población. Se calcula el cociente entre el valor fitness de un individuo y la suma total de los valores fitness de todos los individuos de la población. Este resultado mide la probabilidad de selección Ps(i) de cada individuo.

$$Ps(i) = \frac{f(i)}{\sum_{i=1}^{N} f(i)}$$

Probabilidad de selección

Empezando desde la población en el tiempo t, P(t), de N individuos, se obtiene una nueva población P(t+1) aplicando N veces el operador de selección. Los individuos se seleccionan de una especie de rueda de ruleta (como se muestra en la siguiente figura) donde cada uno tiene asignado un trozo en proporción a su probabilidad de selección Ps.

INDIVIDUO	FITNESS	PROBABILIDAD DE SELECCIÓN
1	169	14,4
2	576	49,2
3	64	5,5
4	361	30,9
TOTAL	1170	100



Operador de Selección

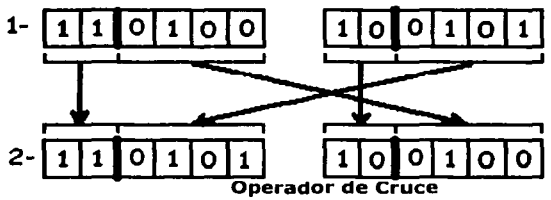
Este mecanismo puede causar problemas de convergencia prematura, causada por la aparición de un individuo que es mucho mejor que los otros de la población aunque esta lejos de óptimo; las copias de este individuo pueden dominar rápidamente a la población, sin poder escapar de este óptimo local.

OPERADOR DE CRUCE

Consiste en unir en alguna forma los cromosomas de los padres que han sido previamente seleccionados de la generación anterior para formar dos descendientes.

Existen diversas variaciones, dependiendo del número de puntos de división a emplear y la forma de ver el cromosoma. El operador cruce se aplica en dos pasos: en el primero los individuos se aparean (se seleccionan de dos en dos) aleatóriamente con una determinada probabilidad, llamada probabilidad de cruce Pc; en el segundo paso a cada par de individuos seleccionados anteriormente se le aplica un intercambio en su contenido desde una posición aleatoria K hasta el final.

K es el punto de cruce y determina la subdivisión de cada padre en dos partes que se intercambian para formar dos nuevos hijos, según podemos ver en la siguiente figura. Esto se conoce como *cruce ordinario o cruce de un punto*. El objetivo del operador de cruce es recombinar subcadenas; esta gestión recibe el nombre de construcción de bloques.



Operador de Cruce

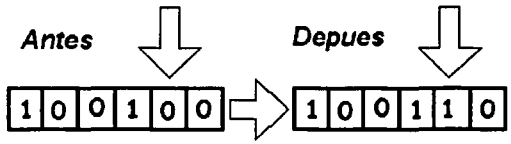
Un cromosoma con alto fitness, puede pasar en forma directa a la nueva generación. Esto se denomina *pasaje directo.*

MUTACIÓN

El operador de mutación consiste en la alteración aleatoria cualquiera de los genes del individuo con una probabilidad de mutación PM, como se puede ver en la siguiente figura.

El objetivo de la mutación es producir diversidad en la población. Si al generar aleatóriamente la población inicial o después de varias generaciones, en la misma posición de todos los cromosomas sólo aparece un único elemento del alfabeto utilizado, esto supondrá que con los operadores de reproducción y cruce, nunca cambiará dicho elemento, por lo que puede ocurrir que jamás se alcance la solución óptima del problema.

La probabilidad de aparición del operador de mutación no debe ser grande para no perjudicar la correcta construcción de bloques. El operador de mutación origina variaciones elementales en la población y garantiza que cualquier punto del espacio de búsqueda pueda ser alcanzado.



Operador de Mutación

CICLO GENERAL DE UN ALGORITMO GENÉTICO ESTANDAR

El AG estándar se puede expresar en pseudo código con el siguiente ciclo:

1. Generar aleatóriamente la población inicial de individuos P(0). Generación = 0
2. Mientras (Número _ generaciones <= máximo _ números _ generaciones)
   Hacer
   {
   Evaluación;
   Selección;
   Reproducción;
   Número_generaciones ++;
   }
3. Mostrar resultados;

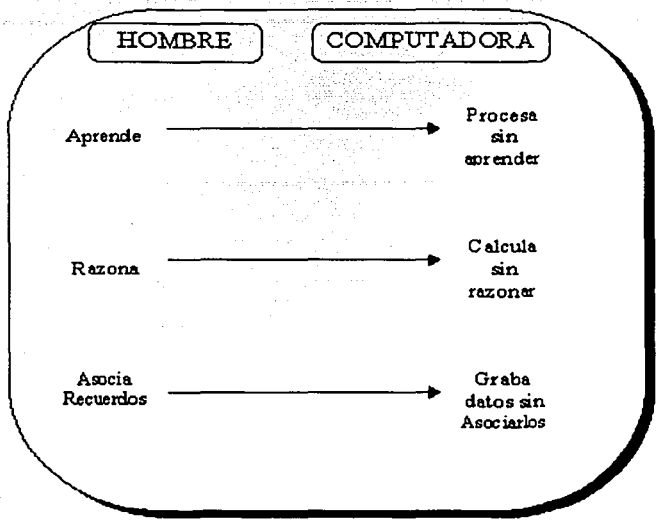
DIFERENCIAS ENTRE AG Y OTROS MÉTODOS DE OPTIMIZACIÓN

1. Un Algoritmo Genético trabaja con codificación de los parámetros que busca optimizar y no con los parámetros en sí mismo.
2. Un Algoritmo Genético trabaja con un conjunto de puntos representativos de diferentes zonas del espacio y no con un solo punto. Por el contrario necesita considerables recursos de computación.
3. La aplicación de AG no depende de ninguna propiedad de la función a optimizar (derivable, continua, ni siquiera conocida), o de que el conjunto de posibles soluciones sea finito o no lo sea.
4. Un AG utiliza reglas de transición probabilísticas, no deterministas, lo cual hace que dos aplicaciones consecutivas de un AG a un mismo problema puedan producir dos soluciones distintas.

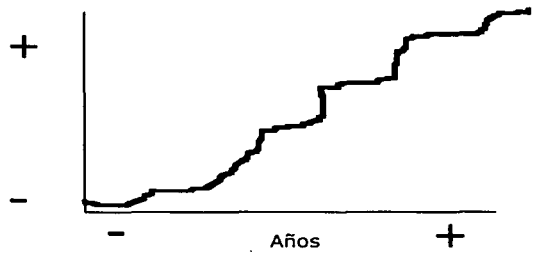
Pseudocódigo ilustrativo de un Algoritmo Genético.

```
BEGIN /* Algoritmo Genético  */
  Generar aleatoriamente una población inicial.
  Calcular la función de evaluación de cada individuo.
  WHILE NOT  Terminado  DO
  BEGIN /* Producir nueva Generación */
       FOR Tamaño Población /2 DO
       BEGIN /* Ciclo Reproductivo */
         1. Seleccionar dos individuos de la anterior generación,
            aplicando el operador de selección.
            (probabilidad de selección proporcional a la
            Función de evaluación de individuo).
         2. Cruzar con cierta probabilidad los individuos seleccionados
            obteniendo descendientes
         3. Mutar si es necesario algunos de los descendientes con cierta probabilidad.
         4. Insertar los dos descendientes en la nueva generación.
       END

       IF solución ad-hoc encontrada THEN
       Terminado =TRUE

  END
END
```

Haciendo una comparación entre las computadoras sin algún tipo de inteligencia (proporcionada por algún método como algoritmos genéticos, redes neuronales o razonamiento basado en memoria) y los humanos, realmente existe una gran diferencia, como lo muestra la siguiente figura.



Se espera que con el paso del tiempo, cada una de las especies (de seres humanos) evolucione de tal forma en la que se puedan superar los errores anteriores. Se supone que la experiencia esta directamente ligada con el tiempo.



Siguiendo esto, se debe esperar, que después de que se realizó un entrenamiento con los algoritmos genéticos por medio de sus funciones e fitness, las siguientes generaciones sean aún más aptas que los predecesores. Siendo esta la idea principal de la cadena evolutiva en la naturaleza, de donde se obtuvo la idea de los algoritmos genéticos.

Sus ventajas. Produce resultados explicables, ya que con la ayuda de la función de adaptación, se pueden ver rápidamente las explicaciones.

Resultados fácil de aplicar, porque toman la forma de los parámetros de la función de adaptación.

Siendo que la función de fitness, sin estar basada en redes neuronales, se aplica a muchos tipos de datos.

Se le puede aplicar una optimización, ya que la función de fitness no sabe si es que se están obteniendo resultados óptimos, por lo que se pueden hacer ciertas correcciones para mejorar los resultados.

Buena integración con las redes neuronales.

Sus desventajas. Es difícil codificar muchos problemas para su aplicación, siendo este el más grande problema de los AG, ya que los problemas deben ser codificados en "cadenas" de genes arregladas; esto determinará gran parte del éxito o fracaso del resultado.

No garantiza resultados óptimos, los AG son una técnica global de optimización, ya que busca la mejora inmediata, más no la mejor solución.

Necesidades de cómputo muy costosas.

Capítulo V

Programas y Código Fuente

Algunos de los programas más reconocidos en el medio para poder realizar una minería de datos (parcial o completa).

Solución		Predicción				
Compañía	Producto	Modelos de predicción de ANN	Funciones de predicción basadas en radios	Predicciones logias difusas	Análisis de series de tiempo	Razonamiento basado en memoria
Adaptative Methods Group at UTS	?	X	X		X	X
Bluecrest Consultancy Ltd.	NeuralParts	X				
Business Objects	BusinessMiner					
ClopiNet	ClopiNet	X				
Cognos	4Thought	X				
Cognos	Scenario					
CSI, Inc.	Advisor Toolkit	X	X	X	X	X
HYPERparallel	Discovery	X				
IBM	Intelligent Miner	X	X		X	X
Integral Solutions Ltd.	Clementine	X			X	
Intellix A/S	KnowMan	X				X
Megaputer Intelligence, Ltd.	PolyAnalyst				X	X
MIT GmbH	DataEngine	X		X		
Neuralware Inc.	NeuralWorks Predict	X				
SAS Institute Inc.	SAS Enterprise Miner	X			X	X
Silicon Graphics Inc.	MineSet		X			
SLP-Infoware	Statlab					
SPSS Inc.	SPSS Products	X	X		X	
Thinking Machines.	Darwin	X				X
Torrent Systems, Inc.	ORCHESTRATE					
Trajecta	dbProphet	X			X	
Unica Technologies, Inc	Unica Pattern Recognition Workbench	X			X	X

A continuación se encuentra el código fuente de una de las técnicas de la detección automática de clusters, el algoritmo de K-medias, escrito en PERL.

```
#! /usr/bin/perl
#algoritmo de las k medias que tomara los datos de un archivo que contendrá los puntos en forma de una
columna
open(coord, "plano.txt") || die "No existe el archivo plano.txt";
$i=0;
while(<coord>){
  chomp;
  ($x[$i],$y[$i])=split(/\s+/,$_);
  $i++;
}
close coord;
#se manejaran las coordenadas como dos por arreglos por separado, con lo que se
#facilitara su manipulación en el calculo de las distancias.

$elementos = $i;
print "\n Se cuenta con ", $i;
print " elementos. \n\n Cuantos clusters deseas formar: ";
chomp($k=<STDIN>);
while ($k <= 0){
   print "\n\nDeben ser uno o mas clusters.";
   print "\n\n Cuantos clusters deseas formar: ";
   chomp($k=<STDIN>);
}

#dado que k es el Número de clusters, se elegirán los primeros k elementos
#del archivo plano.txt, con lo que se toman aleatoriamente.
$kk=0;
while($kk < $k){
#en las variables centro* estarán los centroides que en esa iteración corresponden.
   $centrox[$kk]=$x[$kk];
   $centroy[$kk]=$y[$kk];
   $kk++;
}
#se realizara el calculo de las distancias entre las coordenadas y los centros elegidos

for($j=0; $j < $k; $j++){
    #for ($i=0; $i < $#x; $i++){
    for ($i=0; $i < $elementos; $i++){
                    #se cuenta con una serie de columnas que son las distancias entre todos los puntos y
                    #cada uno de los centros elegidos, por lo que se deben almacenar en un arreglo
                    #bidimensional.
                    #$j representa el centro N, y $i representa el elemento.
      $distx[$j][$i] = $centrox[$j] - $x[$i];
      $disty[$j][$i] = $centroy[$j] - $y[$i];
      $distxy[$j][$i] = sqrt ( ($distx[$j][$i] * $distx[$j][$i]) + ($disty[$j][$i] * $disty[$j][$i]) );
      #printf "\nLa distancia con el centro %d del punto %d es: %f", $j+1, $i+1, $distxy[$j][$i];
    }
}

#se realizaran las comparaciones entre las distancias de los puntos con respecto a los centros
#aleatorios, para determinar la menor y así reacomodar los puntos a un cierto cluster.
    open (clusters, ">>cluster");
for($i=0; $i <= $#x; $i++){
   open (orden, ">>orden.txt");
   for($j=0; $j < $k; $j++){
      print orden $j, " ",$distxy[$j][$i],"\n";
   }
   close orden;
   open (orden, "orden.txt");
   $m=0;
```

```
  while(<orden>){
     chomp;
     ($clus[$m],$dist[$m])=split(/\s+/,$_);
     $m++;
  }
  close orden;
  @dista = sort { $a <=> $b } @dist;

  for ($ord=0; $ord <= $#dista; $ord++){
     if($dista[0] == $dist[$ord]){
        $cluster=$ord;
     }
  }

  `rm -rf orden.txt`;
   print clusters $cluster,"\n";
}
#en el archivo cluster escribo aquel centroide de la tabla que esta mas cercano a ese cluster
close clusters;
`paste plano.txt cluster > plano2.txt`;
$i=0;
$cambios="no";
`sort +2 -n plano2.txt > plano3.txt`;
while( $cambios ne ""){ #este ciclo controla los cambios de estructuración de las coordenadas en clusters
  $kk=0;
  open (newcenter, "plano3.txt");
  while (<newcenter>){ #este ciclo controla el calculo de los nuevos centros de clusters.
    chomp;
    ($xx[$i],$yy[$i],$cluster)=split(/\s+/, $_);
    #si se requiere poner seguridad de $kk <= $k
    if ($kk >= $k){
      $kk=0;
    }
    #$kk es el Número de clusters que el usuario quiere formar.
    if ($kk == $cluster){
     $incrementox+=$xx[$i];
     $incrementoy+=$yy[$i];
    }
    else{
      $centrox[$kk] = $incrementox/($i);
      $centroy[$kk] = $incrementoy/($i);
     $kk++;
     $incrementox=$incrementoy=0;
     $incrementox+=$xx[$i];
     $incrementoy+=$yy[$i];
     $i=0;
    }
    $i++;
  }
  close newcenter;
  `rm -rf plano3.txt`;
  #asignación del ultimo centro del ultimo cluster
  $centrox[$kk] = $incrementox/$i;
  $centroy[$kk] = $incrementoy/$i;
  #inicia el calculo de las distancias de los nuevos centros hacia los puntos.
    for($j=0; $j < $k; $j++){
      for ($i=0; $i < $elementos; $i++){
        $distx[$j][$i] = $centrox[$j] - $x[$i];
        $disty[$j][$i] = $centroy[$j] - $y[$i];
        $distxy[$j][$i] = sqrt ( ($distx[$j][$i] * $distx[$j][$i]) + ($disty[$j][$i] * $disty[$j][$i]) );
      }
    }
    `rm -rf cluster`;
```

```
    `rm -rf orden.txt`;
    open (clusters, ">>cluster");
     for($i=0; $i < $elementos; $i++){
       open (orden, ">>orden.txt");
       for($j=0; $j < $k; $j++){
         print orden $j, " ",$distxy[$j][$i],"\n";
       }
       close orden;
       open (orden, "orden.txt");
       $m=0;
       #sección para la comparación de cual de las distancias es la menor y elegir el siguiente cluster
       while (<orden>){
         chomp;
         ($clus[$m], $dist[$m])=split(/\s+/,$_);
         $m++;
       }

       close orden;
         @dista = sort { $a <=> $b } @dist;
         #$menor=$dista[0];


          for ($ord=0; $ord <= $#dista; $ord++){
           if($dista[0] == $dist[$ord]){
             $cluster=$ord;
           }
          }
       #sección para la comparación de cual de las distancias es la menor y elegir el siguiente cluster

       `rm -rf orden.txt`;
        print clusters $cluster,"\n";
     }
close clusters;
   `paste plano.txt cluster > planox.txt`;
   `sort +2 -n planox.txt > plano3.txt`;
   `rm -rf planox.txt`;
   `rm -rf cluster`;
   #termina el calculo de las distancias de los nuevos centros hacia los puntos.
   $cambios=`cmp plano2.txt plano3.txt`;
   `rm -rf plano2.txt`;
   #`mv plano3.txt plano2.txt`;
   `cp plano3.txt plano2.txt`;
 }
`rm -rf plano3.txt`;
`rm -rf cluster`;
print "\n\nLos resultados están en el archivo plano2.txt";
```

A continuación se muestra un ejemplo de la implementación del algoritmo anteriormente mostrado. El ejemplo 1 con K = 2, muestra el archivo de entrada Plano.txt, y el archivo de salida Plano2.txt. En ambos casos, las columnas corresponden a las coordenadas (X,Y); en el caso del archivo Plano2.txt, la tercer columna representa el cluster al que pertenecen. De igual forma para el ejemplo 2 con K = 4.

Ejemplo 1 con
K = 2

Plano.txt

1	0
2	1
3	2
4	3
5	4
6	4
7	5
8	6
9	7
1	8
1	9
2	3
2	4
3	5
34	24
4	6
5	7
56	16
7	9
78	28
8	7
9	2
9	5
9	6
9	7
11	21
12	12
23	13
12	32

Plano2.txt

1	0	0
11	21	0
12	12	0
12	32	0
1	8	0
1	9	0
2	1	0
2	3	0
23	13	0
2	4	0
3	2	0
3	5	0
4	3	0
4	6	0
5	4	0
5	7	0
6	4	0
7	5	0
7	9	0
8	6	0
8	7	0
9	2	0
9	5	0
9	6	0
9	7	0
9	7	0
34	24	1
56	16	1
78	28	1

Ejemplo 2 con
K = 4

Plano.txt

1	0
2	1
3	2
4	3
5	4
6	4
7	5
8	6
9	7
1	8
1	9
2	3
2	4
3	5
34	24
4	6
5	7
56	16
7	9
78	28
8	7
9	2
9	5
9	6
9	7
11	21
12	12
23	13
12	32

Plano2.txt

12	12	0
4	3	0
5	4	0
6	4	0
7	5	0
7	9	0
8	6	0
8	7	0
9	2	0
9	5	0
9	6	0
9	7	0
9	7	0
1	0	1
1	8	1
1	9	1
2	1	1
2	3	1
2	4	1
3	2	1
3	5	1
4	6	1
5	7	1
11	21	2
12	32	2
23	13	2
34	24	2
56	16	3
78	28	3

A continuación se encuentra el código fuente de una de las técnicas de los árboles decisiones.
Este programa esta basado en una colección de paquetes preconstruidos por la Universidad de Nueva Zelanda.
El proyecto tiene por nombre "WEKA". Página de donde se pueden bajar los paquetes weka.core y weka.classifiers.

```
import weka.core.*;
import java.io.*;
import java.util.*;
import weka.classifiers.*;


/*
*Implémentation de un árbol clasificador ID3
*/

public class Id3 extends DistributionClassifier
{
  //El nodo sucesor
  private Id3[] m_Sucesores;
  //atributo para particionar
  private Attribute m_Atributo;
  //valor de clase si es nodo hoja
  private double m_ValorDeClase;
  //distribucion de la clase si es nodo hoja
  private double[] m_Distribucion;
  //atributo de clase del conjunto de datos
  private Attribute m_AtributoDeClase;

  /*
  *Construcción del árbol de clasificación ID3
  */

  public void buildClassifier(Instances data) throws Exception{
    if (!data.classAttribute().isNominal()){
        throw new Exception("ID3: debe ser clase nominal por favor");
    }
    Enumeration enumAtt = data.enumerateAttributes();
    while (enumAtt.hasMoreElements()){
        Attribute attr = (Attribute) enumAtt.nextElement();
        if (!attr.isNominal()){
                        throw new Exception ("ID3: solo atributos nominales por favor");
        }
        Enumeration enum = data.enumerateInstances();
        while (enum.hasMoreElements()){
          if (((Instance) enum.nextElement()).isMissing(attr)){
                        throw new Exception("ID3: no valores vacios o perdidos por favor");
          }
        }
  }
  data = new Instances(data);
  data.deleteWithMissingClass();
  makeTree(data);
}

/*
*Método para la construcción del árbol ID3
*/

  private void makeTree (Instances data) throws Exception{
  //verificar si los objetos no han sido asignados a este nodo.
  if (data.numInstances() == 0){
      m_Atributo = null;
      m_ValorDeClase = Instance.missingValue();
```

```
    m_Distribucion = new double[data.numClasses()];
    return;
  }

  //calculo del atributo con máxima ganancia de información
  double[] infoGains = new double[data.numAttributes()];
  Enumeration attEnum = data.enumerateAttributes();
  while (attEnum.hasMoreElements()){
    Attribute att = (Attribute) attEnum.nextElement();
    infoGains[att.index()] = computeInfoGain(data, att);
  }
   m_Atributo=data.attribute(Utils.maxIndex(infoGains));

  //Hacer nodo hoja si la ganancia de información es cero.
  //de otra forma, crear un sucesor.
  if (Utils.eq(infoGains[m_Atributo.index()],0)){
    m_Atributo = null;
    m_Distribucion = new double[data.numClasses()];
    Enumeration instEnum = data.enumerateInstances();
    while (instEnum.hasMoreElements()){
      Instance inst = (Instance) instEnum.nextElement();
      m_Distribucion[(int) inst.classValue()]++;
    }
    Utils.normalize(m_Distribucion);
    m_ValorDeClase = Utils.maxIndex(m_Distribucion);
    m_AtributoDeClase = data.classAttribute();
  }
  else{
    Instances[] splitData = splitData(data, m_Atributo);
    m_Sucesores = new Id3[m_Atributo.numValues()];
    for(int j=0; j<m_Atributo.numValues(); j++) {
        m_Sucesores[j] = new Id3();
        m_Sucesores[j].buildClassifier(splitData[j]);
    }
  }
}

/*
* Clasificando un conjunto de prueba dado, usando el árbol de decisión.
*/

public double classifyInstance(Instance instance) {
   if (m_Atributo == null) {
      return m_ValorDeClase;
   }
   else {
              return m_Sucesores[(int) instance.value(m_Atributo)].classifyInstance(instance);
   }
}

/*
*Calculando la distribución de la clase para ejemplos usando el árbol de decisión.
*/

  public double[] distributionForInstance(Instance instance){
   if (m_Atributo == null) {
     return m_Distribucion;
   }
    else{
       return m_Sucesores[(int)instance.value(m_Atributo)].distributionForInstance(instance);
   }
}

/*
*Imprimir el árbol de decisión usando el método privado toString.
*/
public String toString() {
```

```
  return "Clasificador ID3 \n================\n" + toString(0);
}

/*
*Calcula la ganancia de información para un atributo.
*/

private double computeInfoGain(Instances data, Attribute att) throws Exception {
  double infoGain = computeEntropy(data);
  Instances[] splitData = splitData(data, att);
  for(int j=0; j<att.numValues(); j++){
     if (splitData[j].numInstances() > 0){
                infoGain -= ((double) splitData[j].numInstances() / (double) data.numInstances()) *
computeEntropy(splitData[j]);
     }
  }
  return infoGain;
}

/*
*Calcula la entropía de un conjunto de datos.
*/

private double computeEntropy(Instances data) throws Exception{
   double[] classCounts = new double[data.numClasses()];
   Enumeration instEnum = data.enumerateInstances();
   while (instEnum.hasMoreElements()){
     Instance inst = (Instance)instEnum.nextElement();
     classCounts[(int) inst.classValue()]++;
   }
   double entropy = 0;
   for(int j =0; j<data.numClasses(); j++){
      if (classCounts[j] > 0){
        entropy -= classCounts[j] * Utils.log2(classCounts[j]);
      }
   }
   entropy /= (double) data.numInstances();
   return entropy + Utils.log2(data.numInstances());
}

/*
*Partición del conjunto de datos de acuerdo con atributos nominales.
*/
private Instances[] splitData(Instances data, Attribute att){
   Instances[] splitData = new Instances[att.numValues()];
   for(int j = 0; j<att.numValues(); j++){
      splitData[j] = new Instances(data, data.numInstances());
   }
   Enumeration instEnum = data.enumerateInstances();
   while (instEnum.hasMoreElements()){
     Instance inst = (Instance) instEnum.nextElement();
     splitData[(int) inst.value(att)].add(inst);
   }
   return splitData;
}

/*
*salidas del árbol en un cierto nivel.
*/

private String toString(int level){
   StringBuffer text = new StringBuffer();
   if (m_Atributo == null){
      if (Instance.isMissingValue(m_ValorDeClase)){
         text.append(": null");
      }
      else{
```

```
            text.append(": " + m_AtributoDeClase.value((int) m_ValorDeClase));
        }
    }
    else{
        for(int j =0; j < m_Atributo.numValues(); j++){
            text.append("\n");
            for(int i = 0; i < level; i++){
                text.append("| ");
            }
            text.append(m_Atributo.name() + " = " + m_Atributo.value(j));
            text.append(m_Sucesores[j].toString(level + 1));
        }
    }
    return text.toString();
}

/*
*Método principal main
*/

public static void main(String[] arg){
    try{
        System.out.println(Evaluation.evaluateModel(new Id3(), arg));
    }catch(Exception e){
        System.out.println(e.getMessage());
    }
  }
}
```

A continuación se muestra un ejemplo en el que se implementó el código fuente anteriormente implementado. Se muestra una tabla, en la cual, se encuentran los atributos que se consideraron para la construcción del árbol binario y en la última columna de la derecha, el resultado predicho por el algoritmo.

Es un ejemplo que predecirá en base a los siguientes atributos, si es que una persona se esperará a comer en un restaurante, bajo ciertas circunstancias y opciones diversas.

Ejemplo	Atributos									Meta
	Alt	Bar	Vier	Ham	Lleno	Precio	Lluvia	Res	Est	Esperar
X_1	Si	No	No	Si	No	Alto	No	Si	No	
X_2	Si	No	No	Si	Si	Bajo	No	No	Si	
X_3	No	Si	No	No	No	Bajo	No	No	No	
X_4	Si	No	Si	Si	Si	Bajo	No	No	Si	
X_5	Si	No	Si	No	Si	Alto	No	Si	Si	
X_6	No	Si	No	Si	No	Alto	Si	Si	No	
X_7	No	Si	No	No	No	Bajo	Si	No	No	
X_8	No	No	No	Si	No	Alto	Si	No	No	
X_9	No	Si	Si	No	Si	Bajo	Si	No	Si	
X_{10}	Si	Si	Si	Si	Si	Alto	No	Si	Si	

La explicación de cada uno de los campos a considerar es:

- Alt, si el restaurante esta cercano.
- Bar, si el restaurante tiene un área de bar cómoda para esperar mesa.
- Vier, si es día de la semana Viernes.
- Ham, si se tiene mucha hambre.
- Lleno, si la capacidad del restaurante esta llena.
- Precio, si el costo de la comida del restaurante es alto o no.
- Lluvia, si es que esta lloviendo o no.
- Res, si es que se requiere reservación y se hizo con anterioridad.
- Est, Si es que el tiempo de espera es mayor a los 20 minutos.

Siguiendo con estas posibles variantes en la elección de un restaurante, la meta a esperar, tuvo una certeza del 80% en su predicción de la espera del cliente, según la importancia dada a cada uno de los atributos a considerar. Como se muestra en la siguiente tabla.

Ejemplo	Atributos									Meta
	Alt	Bar	Vier	Ham	Lleno	Precio	Lluvia	Res	Est	Esperar
X_1	Si	No	No	Si	No	Alto	No	Si	No	**Si**
X_2	Si	No	No	Si	Si	Bajo	No	No	Si	**No**
X_3	No	Si	No	No	No	Bajo	No	No	No	**Si**
X_4	*Si*	*No*	*Si*	*Si*	*Si*	*Bajo*	*No*	*No*	*Si*	***No***
X_5	Si	No	Si	No	Si	Alto	No	Si	Si	**No**
X_6	*No*	*Si*	*No*	*Si*	*No*	*Alto*	*Si*	*Si*	*No*	***No***
X_7	No	Si	No	No	No	Bajo	Si	No	No	**No**
X_8	No	No	No	Si	No	Alto	Si	No	No	**Si**
X_9	No	Si	Si	No	Si	Bajo	Si	No	Si	**No**
X_{10}	Si	Si	Si	Si	Si	Alto	No	Si	Si	**No**

La razón por la cual se estima el error encontrado en la predicción de la espera de los clientes 4 y 6, fueron por el alto valor de importancia dado a los atributos Lluvia y Lleno.

A continuación se muestra un programa que analiza un archivo plano, en el cual están contenidas las ventas de los clientes. Hace el análisis de la canasta de mercado.

```
#!/usr/bin/perl
`rm -rf tablacooc.txt; rm -rf tabla.txt; rm -rf patrones`;
#Programa que encuentra la matriz de co-ocurrencia dentro de una serie
#de productos marcados en un archivo de texto, simulando cada uno un cliente
#y por lo tanto una transacción que se realizo en una venta.

#cargando el archivo con las transacciones.
open(productos,"prod2.txt") || die "No existe el archivo pod2.txt";
$i=0;
while(<productos>){
  chomp($_=lc($_));
  $tabla{$i}=$_;
  $i++;
}
close productos;
$transacciones=$i;

#deshacer el hash para construir las co-ocurrencias de los
#productos que se vendieron juntos en una transacción dada.

foreach $cliente (keys %tabla){
  @transac=split(/,/,$tabla{$cliente});
  @transac = sort @transac;
  #este foreach es para contabilizar los productos, es decir
  #cuantas veces fue vendido un cierto producto, la diagonal
  #principal de la tabla de co-ocurrencia.
  foreach(@transac){
    $nombre_prod1{$_}+=1;
  }
}

%tabla2=%tabla;
open (tablacc, ">>tablacooc.txt");

#EL SIGUIENTE CICLO ES PARA RECORRER CADA CLIENTE, EXTRAER LA PAREJA DE
#PRODUCTOS A ANALIZAR SU VENTA CONJUNTA Y ASI DETERMINAR CUANTAS VECES
#SE VENDIERON JUNTOS. PRIMERO SE TOMARAN LOS DOS PRODUCTOS A ANALIZAR

foreach $cliente (keys %tabla){
  @venta_prod=split(/,/,$tabla{$cliente});
  @venta_prod=sort @venta_prod;
    for($j=0; $j < $#venta_prod; $j++){
    #este es el producto base, prod1
      for($i=$j+1; $i <= $#venta_prod; $i++){
      #este es el producto flotante, prod2
        $prod1 = $venta_prod[$j];
        $prod2 = $venta_prod[$i];
        #inicia busqueda de los productos juntos en cada transacción de
        #cada cliente.
        foreach $client (keys %tabla2){
           $primero=$segundo='n';
           @venta_prod2=split(/,/,$tabla2{$client});
           @venta_prod2=sort @venta_prod2;
           foreach(@venta_prod2){
             if($_ eq $prod1){
               $primero='s';
             }
             if($_ eq $prod2 && $primero eq 's'){
               print tablacc "Combinación de productos $prod1 y $prod2 en el cliente $client\n";
             }
           }
        }
      }
```

```
    }
}
close tablacc;
`sort tablacooc.txt | uniq > temp`;
`mv temp tablacooc.txt`;


#se hará la construcción de la tabla de forma mas visual.
open (tabla, ">>tabla.txt");
foreach $cliente (keys %tabla){
   @venta_prod=split(/,/,$tabla{$cliente});
   @venta_prod=sort @venta_prod;
   for($j=0; $j < $#venta_prod; $j++){
      for($i=$j+1; $i <= $#venta_prod; $i++){
         open (patron, ">patrones");
         print patron $venta_prod[$j], " y ", $venta_prod[$i];
         close patron;
         chomp($ventas_juntos=`fgrep -w -c -f patrones tablacooc.txt`);
         $cooc=($ventas_juntos/$transacciones)*100;
         printf tabla ("%d ventas de %s y %s juntos, con %.2f%% de frecuencia
\n",$ventas_juntos,$venta_prod[$j],$venta_prod[$i] ,$cooc);
         $ventas_juntos=0;
         `rm -rf patrones`;
      }
   }
}
close tabla;

`sort +0 -n -r tabla.txt | uniq > temp`;
open (tabla, ">tabla.txt");
print tabla "\nEn un total de $transacciones ventas, se reconocieron los siguientes patrones:\n\n";
close tabla;
`cat temp >> tabla.txt`;
`rm -rf tablacooc.txt; rm -rf patrones`;
print "\nLa matriz de coocurrencia esta en el archivo tabla.txt\n";
```

A continuación se muestran dos ejemplos en los cuales se implemento el código fuente anteriormente mostrado, los cuales están divididos en la sección izquierda, está un ejemplo del archivo Prod2.txt, el cual contiene los datos de entrada. En la sección derecha, está un ejemplo del archivo Tabla.txt, el cual contiene los datos de salida, un total de transacciones registradas y el soporte de cada una de las ventas conjuntas encontradas en el archivo Prod2.txt. Cabe mencionar que el formato requerido en el archivo Prod2.txt es el mostrado, productos separados por una ",", sin espacios.

Ejemplo 1, con 5 ventas

Prod2.txt

```
soda,jugo
leche,limpiador,jugo
jugo,detergente
detergente,jugo,soda
soda,limpiador
```

Tabla.txt

En un total de 5 ventas, se reconocieron los siguientes patrones:

2 ventas de oj y soda juntos, con 40% de frecuencia
2 ventas de detergente y oj juntos, con 40% de frecuencia
1 ventas de limpiador y soda juntos, con 20% de frecuencia
1 ventas de limpiador y oj juntos, con 20% de frecuencia
1 ventas de leche y oj juntos, con 20% de frecuencia
1 ventas de leche y limpiador juntos, con 20% de frecuencia
1 ventas de detergente y soda juntos, con 20% de frecuencia

Ejemplo 2, con 15 ventas

Prod2.txt

```
soda,jugo,arroz
leche,limpiador,jugo
jugo,detergente,plumero
detergente,jugo,soda
soda,limpiador
jugo,soda,azucar,jamon
leche,pan,mermelada,crema
jamon,pan,jugo,queso
detergente,escoba,plumero
azucar,frijol,arroz,pan
frijol,soda,jamon
jugo,arroz
limpiador,soda,frijol
mermelada,pan,detergente
arroz,frijol,limpiador
```

Tabla.txt

En un total de 15 ventas, se reconocieron los siguientes patrones:

3 ventas de jugo y soda juntos, con 20.00% de frecuencia
2 ventas de mermelada y pan juntos, con 13.33% de frecuencia
2 ventas de limpiador y soda juntos, con 13.33% de frecuencia
2 ventas de jamón y soda juntos, con 13.33% de frecuencia
2 ventas de jamón y jugo juntos, con 13.33% de frecuencia
2 ventas de frijol y soda juntos, con 13.33% de frecuencia
2 ventas de frijol y limpiador juntos, con 13.33% de frecuencia
2 ventas de detergente y plumero juntos, con 13.33% de frec.
2 ventas de detergente y jugo juntos, con 13.33% de frecuencia
2 ventas de arroz y jugo juntos, con 13.33% de frecuencia
2 ventas de arroz y frijol juntos, con 13.33% de frecuencia
1 ventas de pan y queso juntos, con 6.67% de frecuencia
1 ventas de crema y pan juntos, con 6.67% de frecuencia
1 ventas de crema y mermelada juntos, con 6.67% de frecuencia
1 ventas de leche y pan juntos, con 6.67% de frecuencia
1 ventas de leche y mermelada juntos, con 6.67% de frecuencia
1 ventas de leche y crema juntos, con 6.67% de frecuencia
1 ventas de leche y limpiador juntos, con 6.67% de frecuencia
1 ventas de jugo y queso juntos, con 6.67% de frecuencia
1 ventas de jugo y plumero juntos, con 6.67% de frecuencia
1 ventas de jugo y pan juntos, con 6.67% de frecuencia
1 ventas de jugo y limpiador juntos, con 6.67% de frecuencia
1 ventas de jugo y leche juntos, con 6.67% de frecuencia
1 ventas de jamón y queso juntos, con 6.67% de frecuencia
1 ventas de jamón y pan juntos, con 6.67% de frecuencia
. . .
. .

A continuación se muestra la sintaxis general y algunos ejemplos de las cláusulas de la consulta "select" del SQL como leguaje estándar de consulta en las bases de datos. Esta es la forma de poder hacer reglas de asociación en la técnica Análisis de Asociación.

En SQL con subconsultas y joins (que son intercambiadas frecuentemente), y con vistas se puede ocultar la complejidad de las consultas de mayor interacción.

Subconsultas

Las subconsultas también son conocidas como consultas anidadas. Estas son usadas para resolver preguntas multi-partes y a menudo son intercambiadas por joins. Cuando se ejecuta una consulta que contiene una subconsulta puede ser tratada como un join. Por ejemplo en la búsqueda de nombres de empleados que trabajen en el mismo departamento que el empleado llamado Fernando:

Con subconsulta:

```
SELECT name FROM emp WHERE dept_no =
(SELECT dept_no FROM emp WHERE name = 'Fernando')
```

Con join:

```
SELECT e1.name FROM emp e1, emp e2 WHERE e1.dept_no = e2.dept_no
AND e2.name = 'Fernando'
```

Subconsultas no correlacionadas

El mayor uso de las subconsultas es en la cláusula WHERE de las consultas que define la condición que debe tener el registro como valor. Sin embargo, también pueden ser usadas en otras partes de la consulta, por ejemplo:

1) En las cláusulas WHERE, HAVING y START WITH, que definen el conjunto de renglones para las consultas SELECT, UPDATE y DELETE.
2) En la cláusula FROM de una consulta SELECT, creando una tabla similar a las creadas por las consultas INSERT, UPDATE y DELETE.
3) Para definir un conjunto de renglones para ser creados en la tabla objetivo de las consultas CREATE TABLE AS o CREATE SNAPSHOT.
4) Para definir el conjunto de renglones que serán incluidos por una vista o un snapshot en una consulta CREATE VIEW o CREATE SNAPSHOT.
5) Proveer nuevos valores para columnas especificadas en una consulta UPDATE.

Ejemplo de subconsultas con cláusulas NOT y !, en subconsultas a la misma tabla:

```
Select nombre FROM emp WHERE dept_no NOT IN
(Select dept_no FROM emp WHERE nombre = 'CARLOS')

Select nombre FROM emp WHERE dept_no no !=
(Select dept_no FROM emp WHERE nombre = 'CARLOS')
```

Subconsultas en la cláusula FROM

Las subconsultas pueden ser usadas en esta cláusula como el nombre de una tabla. En ese caso los resultados de la subconsulta serán predefinidos como una vista. El siguiente ejemplo regresa el espacio ocupado, el espacio libre y el total localizado por todas las tablas en una base de datos:

```
Select ts.espaciotabla_nombre,
        ROUND (df.mbytes-fs.mbytes,2) "Usada (Mb)",
        ROUND (fs.mbytes,2) "Libres (Mb)",
        ROUND df.Mbytes,2)  "total  (Mb)"
FROM dba_espaciotabla ts,
        (Select espaciotabla_nombre,
                SUM (bytes) /1024/1024 Mb
        FROM dbadata_archivos_libres
        GROUP BY espaciotabla_nombre) fs,
        (Select espaciotabla_nombre,
                SUM (bytes) /1024/1024 Mb
        FROM dba_archivos_de_datos
        GROUP BY espaciotabla_nombre ) df
WHERE ts.espaciotabla_nombre = fs.espaciotabla_nombre
AND fs.espaciotabla_nombre = df.espaciotabla_nombre
```

Subconsultas que no regresan renglones

Cuando se desea crear tablas, puede ser útil escribir una consulta SQL que tan solo regrese como resultado la estructura de la tabla, pero ningún dato contenido en ella. Por ejemplo:

```
CREATE TABLE nueva AS
(Select * FROM estudiantes WHERE FALSE=TRUE)
```

Consultas correlacionadas

Existen dos tipos de subconsultas: las correlacionadas y las no-correlacionadas.

Las subconsultas no-correlacionadas son ejecutadas una vez por la consulta total; las subconsultas correlacionadas son ejecutadas una vez por cada renglón en la consulta. Los ejemplos anteriores de subconsultas han sido de subconsultas no-correlacionadas.

Las subconsultas correlacionadas responden preguntas multi-partes, pero son usadas frecuentemente para verificar existencia o ausencia de correspondencia de registros en la tabla superior y la actual en la subconsulta.

Una subconsulta correlacionada se refiere a una columna desde una tabla en la consulta superior. Estas consultas pueden ser mejoradas por una consulta con join o una subconsulta no-correlacionada, ya que el SQL es más rápido al usar subconsultas

correlacionadas. Por ejemplo, buscar departamentos donde no haya empleados asignados:

a) Subconsulta no-correlacionada.

```
Select dept.nombre
FROM dept
WHERE dept.id NOT IN
        (Select dept_id
         FROM emp
         WHERE dept_id IS NOT NULL)
```

b) Con outerjoin.

```
Select dept.nombre
FROM dept,emp
WHERE emp.dept_id (+) = dept.id
```

c) Subconsulta correlacionada.

```
Select dept.nombre
FROM dept
WHERE NOT EXISTS (Select dept_id
                  FROM emp
                  WHERE emp.dept_id = dept.id)
```

Análisis de estas consultas:

El outerjoin hace mucho más que solo regresar los nombres de los distintos departamentos que no tengan departamentos asignados a él, podría ser también los departamentos que tuvieron empleados asignados.

La primera y la tercera consultas, producen el mismo resultado, por la primera, será más lenta que la tercera si la columna dept_id en la tabla emp estuviera indexada.

La primer consulta no usa índices, solo checa NOT NULL en la subconsulta, por lo que la búsqueda en la tabla puede ser optimizada.

La subconsulta del tercer ejemplo, puede usar los índices y solo el dept_id es regresado por la subconsulta, no siendo necesaria para cualquier acceso subsecuente a la tabla. Por lo que la tercer consulta tiene un desempeño mejor que la primer consulta.

A continuación se muestra la estructura general de la cláusula SELECT del SQL, además de unos ejemplos que muestran su poder para el análisis de asociación.

```
SELECT [ ALL | DISTINCT | TOP n [ ON ( expression [, ...] ) ] ]
    expression [ AS name ] [, ...]
    [ INTO [ TEMPORARY | TEMP ] [ TABLE ] new_table ]
    [ FROM table [ alias ] [, ...] ]
    [ [LEFT | INNER | RIGTH ] JOIN (condition ) ON ( condition )]
    [ WHERE condition [ [BETWEEN n and m] [NULL] ] = SELECT …]
    [ GROUP BY column [, ...] ]
    [ HAVING condition [, ...] ]
    [ { UNION [ ALL ] | INTERSECT | EXCEPT } select ]
    [ ORDER BY column [ ASC | DESC | USING operator ] [, ...] ]
    [ FOR UPDATE [ OF class_name [, ...] ] ]
   LIMIT { count | ALL } [ { OFFSET | , } start ]
```

La cláusula "select" regresará como resultado cero o varias tuplas de la base de datos que coinciden con los patrones de búsqueda que fueron declarados en este análisis de asociación.

El uso de cada una de las cláusulas afectara al resultado final. Donde:

- DISTINCT, hará una discriminación y solo se reflejaran aquellas tuplas que sean diferentes, es decir, eliminara las repetidas.
- ALL, valor por defecto del sql, no hace ningún tipo de filtrado, muestra todas las tuplas resultantes.
- TOP n, muestra las primeras n líneas del resultado de la consulta.
- *Expresión,* son los campos que se requieren mostrar después de la consulta.
- *INTO*, y las demás opciones, se utilizan para la creación de una tabla a partir de los resultados obtenidos de la consulta, por lo que es la forma de mantener como registros permanentes esos resultados. No afecta el desarrollo de la consulta.
- FROM, indica cual es la tabla(s) de origen para la búsqueda de los patrones.
- *Alias*, se utiliza cuando se requiere trabajar con "sobre-nombres" de las tablas, esto es por velocidad de escritura o cualquier otra razón que haga la consulta mas entendible.
- *INNER JOIN*, es otro tipo de composición de tablas, permite emparejar filas de distintas tablas de forma más eficiente que con el producto cartesiano cuando una de las columnas de emparejamiento está indexada.
- *LEFT JOIN,* es lo mismo que INNER JOIN, solo que el resultado lo agrega a la izquierda de la tabla.
- *RIGTH JOIN,* es lo mismo que INNER JOIN, solo que el resultado lo agrega a la derecha de la tabla.
- *WHERE,* son la serie de restricciones (patrones) de búsqueda que deberán cumplir las tuplas para considerarse como resultado de la consulta. Dentro de esta cláusula, se pueden hacer **subconsultas**, es decir, consultas dentro de la misma consulta, esto es para poder ampliar las restricciones o el campo de acción de esta

consulta con select. Es una parte avanzada del SQL y se recomienda cuando en una sola consulta se desean tener resultados aún más reales o tangibles.

- *BETWEEN*, es la expresión que solo mostrara aquellas tuplas que estén dentro del rango que específica esta cláusula.
- *NULL*, es la comprobación de que esta tupla no tenga valores.
- *GROUP BY*, Utilizada para separar los registros seleccionados en grupos específicos.
- *HAVING*, es una segunda búsqueda de patrones, es decir, para hacer una segunda restricción, con lo que se logra hacer aún más restrictiva la búsqueda.
- *UNION*, es la realización de una mezcla (concatenación) de dos tablas.
- *EXCEPT*, da los registros devueltos por la primera consulta que no se encuentran en la segunda consulta.
- *INTERSECT*, da los registros comunes a ambas consultas.
- *ORDER BY*, es la forma en la que serán ordenados los campos mostrados como resultado de la consulta. Es un ordenamiento general (como por orden ascendente, o por orden alfabético, etc.)
- *FOR UPDATE* permite a SELECT realizar un bloqueo exclusivo de los registros seleccionados.
- *LIMIT*, permite devolver al usuario un subconjunto de los registros producidos por la consulta.

CONSULTA	EXPLICACIÓN
SELECT TITLE, SUM(SALARIO) FROM S_EMP WHERE UPPER(TITLE) NOT LIKE 'VP%' GROUP BY TITLE HAVING SUM(SALARIO)> 5000	Con esta consulta se obtiene el personal con cargo en la empresa que no sea Vicepresidente y con un salario superior a 5000.
SELECT AVG(salario), MIN(salario), MAX(salario), SUM(salario) FROM S_EMP WHERE UPPER(title) LIKE 'SALES%'	Con esta consulta se obtiene el promedio, mínimo, máximo y total del personal que pertenece a ventas.
SELECT DEPT_ID, SUM(salario) FROM s_emp GROUP BY dept_id HAVING SUM(salario) BETWEEN 2000 AND 4000	Con esta consulta se obtiene el identificador de departamento, suma total de salaries del personal de la empresa que tiene un salario entre 2000 y 4000.
SELECT a.nombre AS empleado, b.nombre AS depto, a.salario FROM s_emp a, s_dept b WHERE a.dept_id=b.id AND a.salario > 100	Con esta consulta se obtiene el nombre del empleado y del departamento al que pertenece si es que tiene un salario mayor a 100.

SELECT numemp, nombre, oficina FROM empleados WHERE oficina IN (SELECT oficina FROM oficinas WHERE region = 'este')	Con la subconsulta se obtiene la lista de los números de oficina del *este* y la consulta principal obtiene los empleados cuyo número de oficina sea uno de los números de oficina del *este*. Lista los empleados de las oficinas del *este*.
SELECT numemp, nombre, oficina FROM empleados WHERE EXISTS (SELECT * FROM oficinas WHERE region = 'este' AND empleados.oficina = oficinas.oficina)	Con esta consulta se obtiene el número de empleado, su nombre y oficina si están en la región este y tienen oficina en esa misma región.

A continuación se presenta un ejemplo de uso de la técnica Redes neuronales.

Introducción

Las bases de datos, los data marts, los data warehouse están siendo muy utilizados en la actualidad en negocios, finanzas, ingeniería, medicina y en muchos otros campos; todos estos repositorios contienen grandes cantidades de información que pueden ayudar a predecir el futuro. En muchas áreas de la ciencia y la vida, hay siempre una necesidad de mirar hacia el futuro. Preguntas tan comunes como ser cual será la temperatura máxima mañana, cual será el consumo eléctrico para el próximo trimestre, en que acciones conviene invertir, etc. Son ejemplos de predicción.

Una serie de tiempo es una colección de la información histórica de un sistema, como puede ser un precio de acción, la ganancia de una compañía, etc. Cualquier serie de tiempo puede ser utilizada de dos formas diferentes:

1. Mirando hacia atrás, es decir analizando los comportamientos previos de un sistema sobre la base de la información histórica, como pueden ser las aplicaciones que analizan diagnósticos de enfermedades o fallas en maquinarias de una empresa.
2. Mirar hacia delante, para poder predecir comportamientos futuros de un sistema a partir de la información previa.

Predicción sobre Series de Tiempo con Redes Neuronales

Las redes neuronales son una herramienta muy importante en la predicción de series de tiempos. Estas intentan capturar la dinámica del sistema o serie en cuestión a partir del conjunto de entrenamiento para poder predecir, dada una representación del estado actual del sistema, un valor futuro. Ante un problema dado deben considerarse las diferentes arquitecturas de red que pueden ser utilizadas, y además, antes de empezar a predecir tenemos que tener ciertas pautas en claro, como ser:

- Tener la cantidad de datos suficiente como para poder armar el conjunto de entrenamiento de la red, y poder probar así diferentes arquitecturas.
- Que los datos deberán ser preparados para poder establecer los límites del sistema de predicción.
- Que siempre se deberá predecir un valor representable en el futuro.
- El tiempo de cómputo para predecir el valor buscado. Existen redes que predicen más de un valor al mismo tiempo, pero necesitan más tiempo de cálculo.

Dentro de las redes neuronales existen dos enfoques diferentes para predecir series de tiempo, uno es el llamado **Time Windowing** o de ventana deslizante y el otro son las **Redes Recurrentes**, cada uno de estos enfoques sirven para resolver diferentes

problemas. A continuación veremos una descripción simple de cada una de estos enfoques aplicados a ejemplos particulares.

Time Windowing

La idea aquí consiste en ir moviendo una ventana sobre los valores anteriormente calculados. Dicha ventana abarca n valores anteriores (t, t-1,....,t-n), los cuales son utilizados para obtener el valor t o los valores t,t+1,...,t+k, es decir los k+1 valores más próximos futuros.

Se puede pensar que un valor, ya sea presente o futuro, de una serie de tiempo se puede ver como una función **no lineal** que tiene como parámetro los **n** valores anteriores al valor y_k que se quiere predecir. Es decir:

$$y(k) = f(y(k-1), y(k-2), ..., y(k-n))$$

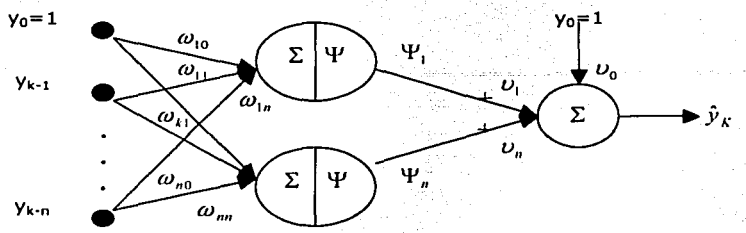
Esta función *y(k)* puede ser obtenida a partir de un modelo regresivo (AR), si es que existe una dependencia lineal de *y(k)* con los últimos **n** valores. Sin embargo para los procesos no lineales un modelo regresivo suele fallar. Por lo tanto se debe encontrar una forma de aproximar esta función bajo cualquier tipo de dependencia; para esto se usan las redes neuronales, que aseguran poder aproximar cualquier función continua. Esto se encuentra demostrado, y el tipo de red utilizada es backpropagation, con una sola capa oculta, y la exactitud de la aproximación depende de la cantidad de neuronas que tienen esta capa oculta, notando que el valor exacto solo se puede obtener con un número infinito de neuronas en dicha capa.

Se verá como poder armar una red para poder predecir valores de una serie de tiempo con una red neuronal bajo este modelo de ventana deslizante. Se debe considerar dos casos, uno donde solo se quiere predecir un valor, y otro donde se quieren predecir más de un valor. Llamaremos a α la distancia de predicción, donde α = 1 indica que solo se quiere predecir un solo valor futuro por paso, y α > 1, si se quiere predecir más de uno. Veamos a continuación ambos casos:

A. Predicción de un sólo valor (α = 1)

Para predecir un solo valor armaremos una red de perceptrones con una capa de entrada, una oculta, y otra de salida, como muestra la figura de más abajo, de **n** neuronas en la capa de entrada ($y_{k-1}, y_{k-2}, ..., y_{k-n}$), para introducir a la red los últimos **n** valores de la serie, **h** neuronas en la capa intermedia y una neurona $\hat{y}_k$ en la capa de salida para predecir el valor futuro. w_{ji} y v_j son los pesos de las conexiones entre las capas

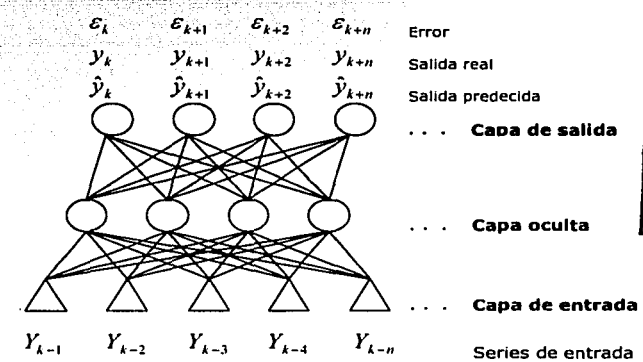
diferentes capas, y ψ_j es la función de transferencia no lineal (normalmente suele ser una función sigmoidea)



Con este tipo de redes se puede predecir el k-ésimo valor, y si queremos el k+1, debemos realimentar la red con el valor k-ésimo antes predicho, es decir desplazando la ventana un paso hacia la derecha.

B. Predicción de varios valores ($\alpha = 2,3,\ldots$)

Para predecir varios valores futuros se puede construir un perceptrón de dos capas (la capa de entrada no se toma como tal), con α_{max} neuronas de salida para predecir en forma simultánea los valores $y_k, y_{k+1}, y_{k+2}, \ldots\ldots, y_{k+\alpha_{max}}$, a partir de las **n** neuronas de entradas ($y_{k-1}, y_{k-2}, \ldots\ldots, y_{k-n}$), como se puede ver en la siguiente figura:



El o los valores predichos por esta red se calculan de la siguiente forma:

$$\hat{y}_k = v_0 + \sum_{j=1}^{h} v_j \psi_j \left(\sum_{i=1}^{n} w_{ji} y_{k-i} + w_{j0} \right)$$

Donde w_{ji} son los pesos de los arcos que conectan todas las **i** neuronas de entrada con las **j**-ésima neurona de la capa intermedia, mientras que v_j son los pesos de los arcos que van de la **j**-ésima neurona de la capa intermedia a la neurona o neuronas de salida.

El error de predicción es la diferencia entre el valor real y el predichos, es decir $e_k = y_k - \hat{y}_k$, dicho error se utiliza para entrenar la red y probar su desempeño.

A continuación se muestra una tabla como condensado de las diferentes variantes que se pueden presentar dentro de una red neuronal.

Ejemplo	Conjunto de entrenamiento	Capa de entrada	Capa oculta	Capa de salida	Número de iteraciones	Valor a predecir	Conjunto de prueba
1	70	20	5	α=1	220	21º	30
2	70	25	5	α=1	220	26º	30
3	70	25	10	α=5	220	26º-30º	30

A continuación se presenta un ejemplo de uso de Algoritmos Genéticos en la Predicción sobre Series de Tiempo.

Se examinan métodos para ejecutar predicción sobre series de tiempo del mundo real. En general series de tiempo sobre temas bursátiles, bioquímicos, financieros, etc. tienen comportamientos no lineales, para lo cual resulta imprescindible contar con herramientas particulares para tratarlas.

Se suele escuchar hablar de predictores no lineales de muy alto desempeño que para funcionar solo requieren de unos cuantos parámetros relacionados con el dominio del problema, por ejemplo 7. En muchos casos encontrar los valores para estos parámetros lleva meses de experiencia, prueba y error, en otros nunca se encuentran y el predictor no se utiliza.

En este ejemplo se intentará mostrar un predictor que no requiere parámetros, o mejor dicho, los encuentra automáticamente.

Derivación automática de parámetros

Hay ciertos algoritmos que requieren varios parámetros para funcionar correctamente. Si se dispone de un método para probar estos parámetros se podría usar por ejemplo fuerza bruta e ir combinando todos los posibles valores de los parámetros, probarlos y decidir cual es el que mejor se adapta al problema. Sin embargo en un caso típico podrían ser 7 los parámetros con 256 valores posibles cada uno. En este caso tendríamos 256^7>72.057.594.037.930.000 combinaciones posibles para lo cual no disponemos de tiempo ni otros recursos para realizar las pruebas correspondientes.

En casos como estos los **Algoritmos Genéticos** son de gran utilidad. Una de sus principales ventajas es que funcionan bien en presencia de ruido o datos irrelevantes.

La idea es utilizar algoritmos genéticos para optimizar un conjunto fijo de parámetros para un proceso interactivo con un ambiente externo (en este caso un predictor sobre series de tiempo).

El primer paso es definir el formato de cada individuo de la población. Un individuo es un juego de valores para cada uno de los parámetros en cuestión. Estos individuos son llamados *Cromosomas*. Supongamos por ejemplo que contamos con 3 parámetros (Pa, Pb, Pc).

Si los parámetros tienen valores reales o un rango excesivamente grande deberemos discretizarlos.

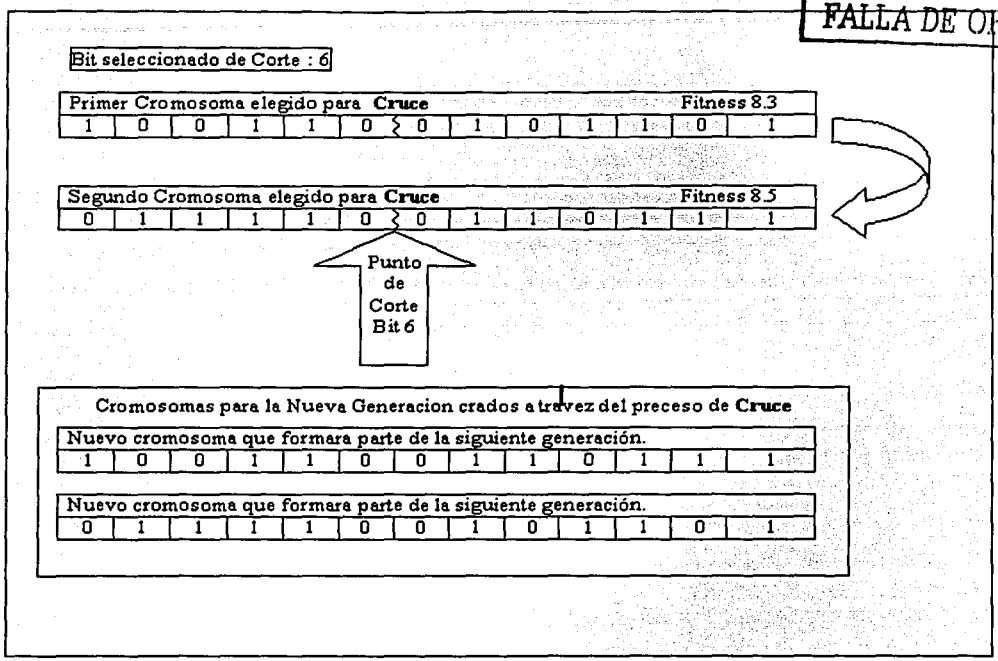
Los parámetros dentro del cromosoma serán expresados en forma binaria y se utilizaran los bits necesarios dependiendo de los valores posibles que estos parámetros puedan tomar. Por ejemplo si Pa tiene valores entre 0 y 10 se podrán utilizar 4 bits para

representarlo dentro del cromosoma, en este caso se deberá tener en cuenta de inhibir la posibilidad que el algoritmo genético produzca cromosomas que para este parámetro contengan valores superiores a 10. También podría pasar que Pb sea un valor real y por algún motivo se decida discretizarlo en 128 posibles valores (7 bits), en este caso antes de probar un cromosoma se deberá pasar el valor discretizado de Pb al valor correspondiente que realmente el algoritmo externo de predicción espera. Podría suceder que el parámetro Pc indique una forma de procesamiento para el algoritmo externo, esta forma de procesamiento podría tener cuatro valores posibles y para estos se utilizarían 2 bits. Ver siguiente figura.

Un Cromosoma														
Pa (4 bits)				**Pb** (7 bits)								**Pc** (2 bits)	
1	0	0	1	1	0	0	1	0	1	1		0	1
Pa=9				Pb=75, que representa por ejemplo un valor entre 129.64 y 131.96.								Pc=1 que indica una forma de Procesamiento.	

Viendo la figura anterior se puede entender fácilmente como se representa un cromosoma, simplemente concatenando la cadena de bits que representan cada uno de los parámetros.

Cada cromosoma deberá ser evaluado por el programa externo (en este caso el predictor) y este le dará un valor indicando que tan "bueno" es el cromosoma, este valor se llama **Aptitud** o **Fitness**.



Creación de una nueva generación de cromosomas

Algunas Consideraciones

Los datos de la serie de tiempo deben discretizarse y en muchos casos normalizarse, pues muchos métodos como las redes neuronales necesitan entradas con valores entre –1 y 1 y otros algoritmos no trabajan bien con valores en escalas muy distintas (precio 100$ y volumen 10.000.000 cm^3). En estos casos el predictor devolverá datos normalizados, escalados y discretizados con lo cual deberán desnormalizarse los valores devueltos y mostrar el rango de error debido a la discretización.

Las series de tiempo siempre contienen discontinuidades (días no laborables, horarios no laborables, feriados, días donde no se opera, etc.). Los algoritmos suelen confundirse en estos casos por lo tanto conviene quitarle a la serie los espacios sin datos. Otro problema muy común es encontrar datos evaluados en intervalos de tiempo no regulares. Esto tampoco es bueno para el algoritmo por lo cual se deberá regularizar estos intervalos de tiempo.

Muchas veces sucede que la variación de los datos en una serie de tiempo es más importante y significativa que los datos en si. Si este es el caso, puede preprocesarse la serie antes o después de la normalización de sus datos e incluir en ella la variación de sus valores.

Descripción de un predictor realizado sobre la base de las herramientas y algoritmos presentados

La idea básica de este predictor es encontrar una distancia entre todos los puntos de la serie. Esta distancia no es fácil de encontrar y no depende del tiempo, sino mas bien de los atributos de la serie. Una vez encontrada la distancia y dado un punto de prueba se buscan los vecinos mas cercanos a este y se evalúa como estos evolucionaron en el tiempo para así predecir el valor futuro a partir de este punto de prueba.

A continuación se muestra un ejemplo de una posible implementación de un sistema que se basa en la técnica de Razonamiento Basado en Memoria. Un caso particular de esta técnica es el empleado aquí, el Razonamiento Basado en Casos.

Durante el proceso enseñanza-aprendizaje frecuentemente ocurren situaciones como la siguiente:

> Después de haber explicado algunos conceptos básicos, el profesor confronta a los alumnos ante un problema a resolver, provocando que los alumnos comiencen a proponer diferentes soluciones (incorrectas, medio correctas y correctas). Al escuchar las propuestas de los alumnos, el profesor empieza a plantear nuevos casos, ejemplos con los que pretende reafirmar el grado de certeza de los alumnos; incluso, el profesor también puede preparar y plantear nuevos casos problemas que se van encadenando entre sí (según el tipo de respuesta de los alumnos) para ir dirigiendo el proceso de aprendizaje del alumno.

El proceso de enseñanza-aprendizaje que se lleva a cabo es, por lo general, un proceso lleno de dinamismo mental. Tanto el profesor como los alumnos hacen intenso uso de sus capacidades cognoscitivas, siendo la memoria uno de los puntos más importantes.

El Razonamiento Basado en Casos (RBC) es un nuevo paradigma en el área de la Inteligencia Artificial (IA) que provee, por un lado, un modelo cognitivo de la organización de la memoria, el razonamiento y el aprendizaje humano; y por otro lado, una nueva técnica computacional para la solución de problemas.

La idea sobre la cual descansa este enfoque es realmente sencilla. Se le da al sistema una especificación de entrada y el sistema busca en su memoria de casos uno ya existente que encaje o corresponda con la especificación de entrada dada. En la mejor de las situaciones se encontrará un caso que encaje perfectamente con el problema dado, obteniendo la solución directamente. Pero si no es así, se encontrará uno o varios casos similares.

Las posibilidades de que se encuentre un caso que encaje perfectamente con la especificación de entrada aumentan cuando la *memoria* del sistema (base de casos) es incrementada. Por otro lado, el encontrar casos que sólo sean similares no significa que ya se haya encontrado una solución completa. Para encontrar la solución completa el usuario y el sistema entran en un proceso de *adaptación de casos*. En este proceso se encuentran y modifican pequeñas porciones del o los casos similares encontrados. Con esto se logran dos cosas: una solución completa al problema y un nuevo caso que el sistema puede *aprender*.

Así, los aspectos clave que se deben afrontar en el desarrollo de sistemas computacionales bajo el enfoque del RBC son: la organización e indexación de la memoria, el criterio de selección, las medidas de similitud entre casos, el criterio de adaptación y el aprendizaje de nuevos casos.

Aunque actualmente la investigación en el área del RBC continúa muy activa en los centros de investigación, ya se han desarrollado varios sistemas prácticos y útiles en la vida real; existen sistemas para el diseño, la planificación, el asesoramiento, la programación de tareas, etc.

El objetivo de la aplicación de esta técnica es, principalmente el diseño de un conjunto de técnicas y algoritmos bajo el enfoque del Razonamiento Basado en Casos para el desarrollo de un sistema computacional de apoyo en el proceso de enseñanza-aprendizaje; y con base en este diseño mostrar la funcionalidad del enfoque propuesto, y así lograr el desarrollo de un prototipo computacional de apoyo en la enseñanza del lenguaje de programación Pascal.

En un proceso de enseñanza-aprendizaje, se consideran los siguientes pasos:

a) El profesor actúa precisamente como un razonador basado en casos: muestra un caso, recibe retroalimentación, recuerda otros casos, quizá los adapta, y los presenta para su consideración; recibe más retro-alimentación; ésta a su vez, le recuerda nuevos casos, los adapta, modifica y confronta a los alumnos con ellos; y así, encadenando caso tras caso va guiando al alumno hasta alcanzar los objetivos.
b) El alumno por su parte, se encuentra realizando un proceso de aprendizaje basado en casos: recibe un caso, lo analiza, recuerda otros similares, ve sus ventajas y desventajas y lo relaciona con casos similares; emite su opinión; recibe nuevos casos; los analiza, los compara con los que vienen a su mente; y genera nuevos casos como generalizaciones y especializaciones.

Algunos trabajos presentan algunas arquitecturas computacionales para la enseñanza basada en casos. Estas arquitecturas se basan en dos suposiciones básicas: a) los expertos son básicamente repositorios de casos; y b) el aprendizaje se lleva a cabo sólo cuando el alumno se encuentra listo para recibir los casos adecuados del repositorio del experto. Las arquitecturas presentadas por Schank explotan, por un lado, la capacidad básica de los alumnos de aprender de las historias o casos que el profesor les cuenta; y por el otro, el deseo natural de los profesores por contar sus experiencias personales. Papert ha distinguido, por su parte, dos principios necesarios para el buen aprendizaje: el *principio de continuidad* y el *principio del poder*. El primero se refiere precisamente al hecho de que el receptor debe estar listo o preparado para escuchar la historia. El segundo se refiere al hecho de que el receptor además debe estar interesado; es decir, la información contenida en la historia debe ser interesante y útil para su inmediato aprovechamiento en algún problema actual del receptor.

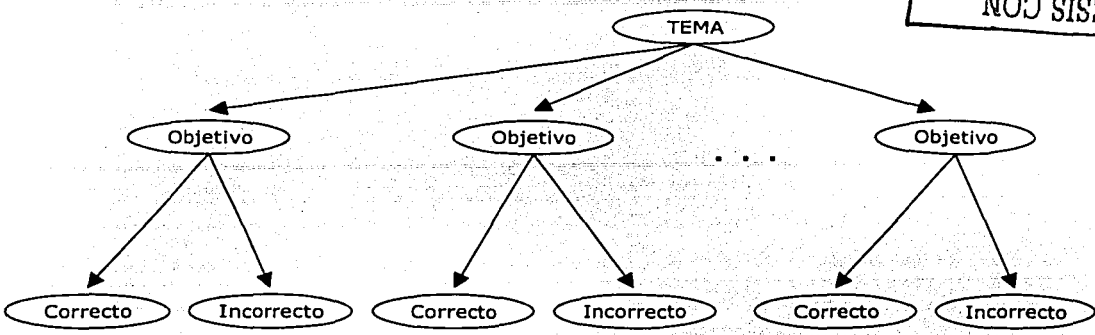
Así pues, siguiendo el enfoque del Razonamiento Basado en Casos, a continuación se presenta el diseño del prototipo del sistema de apoyo en el proceso de enseñanza-aprendizaje del lenguaje de programación Pascal que se propone.

El funcionamiento del algoritmo que podría seguir el sistema se muestra a continuación:

1. Seleccionar un tema en particular.
2. Presentar los conceptos básicos del tema.
3. El sistema entra en un ciclo, repitiendo los siguientes cinco pasos básicos hasta que el alumno encuentre una solución al problema planteado:
   a) Confrontar al alumno con un problema a resolver (completar o remediar algún caso).
   b) El alumno propone una solución (modifica el caso problema dado, obteniendo así un caso nuevo).
   c) El sistema evalúa el caso nuevo y conforme a las características de la solución propuesta (correcta o incorrecta) busca en su base de casos por aquellos que guarden similitudes con el caso actual.
   d) El sistema *aprende* el caso planteado por el alumno, almacenándolo correctamente en su memoria para su futura re-utilización.
   e) El sistema proporciona retro-alimentación al estudiante con base en los casos encontrados, mostrando ejemplos y contra-ejemplos de casos similares con errores similares o relacionados. Con base en esto el sistema puede adaptar algún caso, produciendo un nuevo problema para el alumno.
4. Una vez terminado el ciclo del punto anterior, el estudiante puede volver a empezar desde el punto número uno.

La memoria o base de casos del sistema es el corazón del mismo, a partir de esto se desprende la indexación y selección de casos similares. En memoria se almacenan los casos como objetos formados por pares atributo-valor, tal como puede apreciarse en la siguiente tabla.

Atributo	Valor(es)
Tema	
Objetivo	
Correctitud	
Componentes	
Punto clave	
Explicación	
Lista de chequeo	



Posibles selecciones del MBR según los casos conocidos

Los atributos son descriptores del caso:

1. El tema puede ser por ejemplo "estatutos de repetición", "modularidad", etc.
2. El objetivo indica lo que se persigue con dicho caso; por ejemplo: "ciclos infinitos", "función para el factorial", "búsqueda binaria", etc.
3. El atributo de correctitud es una bandera para marcar los casos como ejemplos correctos o contraejemplos.
4. El atributo de componentes es un atributo multi-valuado en el que se almacenan todos los distintos componentes del caso (instrucciones o partes de ellas). Estos componentes varían en cantidad de caso a caso.
5. El punto clave sirve para marcar el componente crítico que sea de especial interés de cada caso.
6. En la explicación se tiene una breve y concisa descripción del funcionamiento del caso que se presenta.
7. La lista de chequeo se tiene una lista de puntos clave a revisar para la evaluación del caso.

Indexación. Los índices que se usan para acceder eficientemente, los casos en memoria se basan en los tres primeros atributos de cada caso: tema, objetivo y correctitud. Con base en estos índices los casos se almacenan en memoria bajo una estructura jerárquica como la que se puede apreciar en la figura anterior. Los casos se encuentran agrupados en memoria en conjuntos de casos similares según la estructura de índices mostrada.

Selección. Una vez contando con los índices y la organización jerárquica de casos que se mostró anteriormente, el algoritmo de selección resulta realmente sencillo. De esta forma se puede buscar y encontrar de una manera rápida y sencilla por casos similares (tomando como criterio de similitud al indicado por los índices). Obteniendo fácilmente dos categorías de casos similares: correctos e incorrectos.

Adaptación. Una vez que el sistema recibe una propuesta de solución por parte del usuario. El sistema busca por los casos similares en memoria para presentarlos al alumno. Sin embargo ahí no acaba todo, el alumno y el sistema entran en un proceso de evaluación del caso presentado como solución con ayuda de la información almacenada en las explicaciones y las listas de chequeo principalmente. Después el sistema se adapta, cambiando ligeramente algún componente de algún caso (puede ser el caso presentado originalmente, el caso presentado como solución por el alumno o, incluso, alguno de los casos similares que se encontraron) para crear un nuevo caso problema.

Aprendizaje. Por lo general, en cada ocasión que el alumno propone una nueva solución realmente está creándose un nuevo caso (que puede ser o no ser correcto). Cada caso nuevo que el alumno propone el sistema lo va almacenando en su memoria para su futura re-utilización. Para almacenar estos casos el sistema conserva la organización y la estructura jerárquica de índices de los casos mostradas anteriormente. Cabe comentar que ésta no es la única forma en que el sistema puede ir aprendiendo nuevos casos, también puede recibir casos nuevos que sean capturados en forma directa.

CONCLUSIONES Y COMENTARIOS

Después de haber realizado esta investigación sobre la minería de datos, data warehouse y la preparación de datos, son muchos los atributos que puedo atribuirle a estas técnicas de búsqueda de patrones ocultos en la información.

Dados los cambios tan vertiginosos que existen actualmente en cualquier rama del quehacer humano, es necesario poder estar actualizados, pero sobre todo preparados.

La mayoría de las empresas trasnacionales generan tanta información en un día como la que una persona puede leer en toda su vida, esto implica que mucha de esa información tan solo es almacenada en los grandes DBMS y difícilmente es analizada con detalle para poder saber alguna característica de la misma.

Sin duda, una de las aplicaciones de la minería de datos que más llamo mi atención es la de la mercadotecnia, ya que la misma información que genera la empresa es la que se debe y puede utilizar para entender y explotar mejor el mercado de acción.

Para poder realizar la minería de datos es muy importante poder contar con muchas herramientas, personal y plan de trabajo, pero lo que realmente hará que la minería de datos sea exitosa o no, es que la información recabada provenga de una fuente fidedigna, además de que se debe contar con el apoyo de todo el personal que intervenga en la recolección de la misma.

Por lo que la preparación de los datos es un proceso de vital importancia y así poder eliminar casi todas aquellas variantes, duplicidad, inexistencia o cualquier falla que tengan intrínsecamente los datos.

Para su almacenamiento, la óptima organización es dentro de un data warehouse, pero dadas las limitantes económicas, de personal y de tiempo, no siempre es posible realizar la creación del mismo. Sin embargo también se pueden aplicar ciertas técnicas de la minería de datos en una base de datos normal.

Una vez más, hago mención de la importancia de la existencia de una cultura del correcto tratamiento de la información antes de ser almacenada en el data warehouse. Y estas herramientas sólo tendrán éxito cuando todas las personas involucradas proporcionen la información adecuada y obtengan también, información de valía para sus procesos operacionales y tomas de decisiones.

Las diversas técnicas de la minería de datos existen por la gran diversidad de problemas, tipos de datos y enfoques que se pueden encontrar en cada uno de los distintos problemas de la vida real.

Es importante entender que la minería de datos no resolverá los problemas que existan en la pérdida de información, es decir, tan sólo arrojara resultados que deberán

ser interpretados por expertos del tema y poder proponer soluciones con base a los patrones que logró encontrar la minería de datos.

Me interesó este tema por todo el potencial que existe en casi cualquier base de datos y que normalmente se busca en otros lugares la respuesta de muchas de las preguntas, cuando dentro de los mismos datos con los que ya se cuenta están muchas de las respuestas.

Es indudable que en algunas ocasiones se llegara a requerir de fuentes de datos externas, por las que también se debe de poner atención para asegurar su integridad y confiabilidad.

Sin duda alguna, creo que la minería de datos es una aplicación de ingeniería en toda la extensión de la palabra, ya que, no solo toma ciertos aspectos de las ciencias exactas, sino que su enfoque principal es la resolución de problemáticas reales, en cualquier ámbito del quehacer humano.

La minería de datos es una de las mejores formas de poder rescatar información clave, información precisa y útil, si es que se realiza a nuestros datos, ya que de no ser nuestros datos, es probable que las soluciones propuestas, no sean las adecuadas.

Sin embargo, la parte más importante de la minería de datos, no es la técnica implementada, la arquitectura de almacenamiento que se eligió o el pre-procesamiento de la información, sino es la solución propuesta a nuestros problemas con base a nuestra información pre-existente; una vez aplicados el pre-procesamiento, la arquitectura de almacenamiento y la(s) técnica(s) elegida(s).

Ya que de no ser así, no tendría sentido alguno, la monumental puesta en marcha de las consideraciones que se deben realizar al realizar una minería de datos.

Considero que la minería de datos en México aun esta rezagada, lo cual puede afectar a todas aquellas actividades o empresas, que dependen de la toma de decisiones correcta y oportunamente, en comparación con otros países, en donde la minería de datos es una actividad relativamente cotidiana.

Las mayores problemáticas que encontré de la minería de datos, son su excesivo costo comercial y el tiempo que se requiere invertir para su realización. Sin embargo, las propuestas de programas y algunos principios que propongo en este trabajo, puede abatir parcialmente estas dos grandes limitantes.

Una solución para realizar una minería de datos, es poder contratar a una empresa externa que realice este análisis a los datos de la organización. Sin embargo, el riesgo que se tiene, es que los estudios realizados no sean tan exactos y la realización de los mismos sea de un costo excesivo. Otra posibilidad es que una empresa externa trate y manipule los datos de la organización, sin embargo se debe asegurar que la consistencia sea la

adecuada y sea una fuente fidedigna e información. La última posibilidad es que la misma organización sea la que realice el tratamiento y análisis de la información, sin embargo hay que capacitar al personal para que sean estudios de calidad.

Lo que supone, una buena comparación de cualquiera de las opciones para la realización de una minería de datos, para que la inversión realizada en tiempo, recursos y dinero, pueda ofrecer los resultados esperados, siendo suficientes, oportunos y adecuados según la organización y sus necesidades.

Pude determinar que cada una de las técnicas comparadas tiene sus rasgos muy bien definidos para un tipo de problema específico, por lo que su implementación, implica tener un buen conocimiento del dominio del problema.

La construcción del data warehouse, es una serie de esfuerzos y gastos considerablemente grandes, por lo que la mayoría de las organizaciones optan por un almacén de datos sencillo, como un data mart o un DBMS, sin embargo, lo que presenté en este trabajo es un caso ideal, porque proporciona todas las herramientas necesarias.

Una preparación de datos, se debe realizar siempre, con o sin la presencia de una posible minería de datos, ya que esto eliminará mucha de la incongruencia de la información almacenada.

Los programas que realice, están encaminados para dar una posible idea de cómo se realizaría una implementación de la minería de datos, de forma semiautomática, ya que es necesario un preprocesamiento de los datos.

Finalmente, la presente investigación cumple con el objetivo de mostrar las potencialidades de la minería de datos y, que sus resultados al ser interpretados de la forma adecuada, pueden ayudar a la toma de decisiones estratégicas en una organización.

La intención de mostrar los aspectos de la minería de datos mencionados en esta tesis, es para que aquellas personas que requieren de apoyo en la toma de decisiones, consideren estas técnicas como una posible solución.

BIBLIOGRAFIA

- Adriaans, Peter, Dolf Santingue. "*Data mining*", Ed. Softcover, EUA 1999.
- Adriaans, Peter, Dolf Santingue. "*Data Mining*", Ed Addison-Wesley, EUA 1996.
- Berry, Michael J. A., Gordon Linoff. "*Data mining techniques, for marketing, sales and customer support*", Ed. Wiley Computer Publushing, EUA 1997.
- Berry, Michael J. A., Gordon Linoff. "*Mastering Data Mining. The art and science of customer relationship management*", Ed. Wiley Computer Publushing, EUA 2000.
- Fayyad, Usama M, Gregory Piatetsky-Shapiro, Padhraic Smyth & Ramasamy Uthurusamy. "*Advances in Knowledge Discovery and Data Mining*", Ed M.I.T. Press, EUA 1996.
- Gill, Harjinder S., Prakash C. Rao. "*Data warehousing. La integración de información para la mejor toma de decisiones*", Prentice Hall Hispanoamericana, México 1996.
- Inmon, William. "*Building the Data Warehouse, Second Edition*", Ed John Wiley & Sons, EUA 1996.
- Kamber, Jiawei Han and Micheline. "*Data mining, Concepts and techniques*", Ed Morgan Kaufmann, EUA 2000.
- Kimball, Ralph. "*Data Warehouse Toolkit*", Ed John Wiley & Sons, EUA 1996.
- Pyle, Dorian. "*Data preparation for data mining*" Ed Morgan Kaufmann Publishers, EUA 1999.
- Rousseeuw Kaufman, Leonard, Peter J.. "*Finding Groups in Data: An Introduction to Cluster Analysis*", Ed John Wiley & Sons, EUA 1990.
- Witten, Ian H., Eibe Frank. "*Data mining. Practical machine learning tools and techniques with java implementations*", Ed Morgan Kaufmann Publishers, EUA 2000.
- Zaki, Mohammed J.; Ho, Ching-Tien. "*Large-Scale Parallel Data Mining*", Ed Softcover, Alemania 2000.

GLOSARIO

Agregación: Actividad de combinar datos desde múltiples tablas para formar una unidad de información más compleja, necesitada frecuentemente para responder consultas del Data Warehouse en forma más rápida y fácil.

Algoritmos genéticos: Técnicas de optimización que usan procesos tales como combinación genética, mutación y selección natural en un diseño basado en los conceptos de evolución natural.

Análisis de series de tiempo (time-series): Análisis de una secuencia de medidas hechas a intervalos específicos. El tiempo es usualmente la dimensión dominante de los datos.

Análisis prospectivo de datos: Análisis de datos que predice futuras tendencias, comportamientos o eventos basado en datos históricos.

Análisis exploratorio de datos: Uso de técnicas estadísticas tanto gráficas como descriptivas para aprender acerca de la estructura de un conjunto de datos.

Análisis retrospectivo de datos: Análisis de datos que provee una visión de las tendencias, comportamientos o eventos basado en datos históricos.

Árbol de decisión: Estructura en forma de árbol que representa un conjunto de decisiones. Estas decisiones generan reglas para la clasificación de un conjunto de datos. Ver CART y CHAID.

Base de datos multidimensional: Base de datos diseñada para procesamiento analítico on-line (OLAP). Estructurada como un hipercubo con un eje por dimensión.

CART Árboles de clasificación y regresión: Una técnica de árbol de decisión usada para la clasificación de un conjunto da datos. Provee un conjunto de reglas que se pueden aplicar a un nuevo (sin clasificar) conjunto de datos para predecir cuáles registros darán un cierto resultado. Segmenta un conjunto de datos creando 2 divisiones. Requiere menos preparación de datos que CHAID.

CHAID Detección de interacción automática de Chi cuadrado: Una técnica de árbol de decisión usada para la clasificación de un conjunto da datos. Provee un conjunto de reglas que se pueden aplicar a un nuevo (sin clasificar) conjunto de datos para predecir cuáles registros darán un cierto resultado. Segmenta un conjunto de datos utilizando tests de chi cuadrado para crear múltiples divisiones. Antecede, y requiere más preparación de datos, que CART.

Clasificación: Proceso de dividir un conjunto de datos en grupos mutuamente excluyentes de tal manera que cada miembro de un grupo esté lo "más cercano" posible a otro, y grupos diferentes estén lo "más lejos" posible uno del otro, donde la distancia está medida

con respecto a variable(s) especifica(s) las cuales se están tratando de predecir. Por ejemplo, un problema típico de clasificación es el de dividir una base de datos de compañías en grupos que son lo más homogéneos posibles con respecto a variables como "posibilidades de crédito" con valores tales como "Bueno" y "Malo".

Clustering (agrupamiento): Proceso de dividir un conjunto de datos en grupos mutuamente excluyentes de tal manera que cada miembro de un grupo esté lo "más cercano" posible a otro, y grupos diferentes estén lo "más lejos" posible uno del otro, donde la distancia está medida con respecto a todas las variables disponibles.

Data cleansing: Proceso de asegurar que todos los valores en un conjunto de datos sean consistentes y correctamente registrados.

Data warehouse: Base de datos que almacena una gran cantidad de datos transaccionales integrados para ser usados para análisis gestionales por usuarios especializados (tomadores de decisión de la empresa). Sistema para el almacenamiento y distribución de cantidades masivas de datos.

Data Mart: Conjunto de hechos y datos organizados para soporte decisional basados en la necesidad de un área o departamento específico. Los datos son orientados a satisfacer las necesidades particulares de un departamento dado teniendo sólo sentido para el personal de ese departamento y sus datos no tienen porque tener las mismas fuentes que los de otro Data Mart.

Data mining: Análisis de los datos para descubrir relaciones, patrones, o asociaciones desconocidas. La extracción de información predecible localizada de forma difusa en bases de datos.

Datos anormales: Datos que resultan de errores (por ej.: errores en la captura durante la carga) o que representan eventos inusuales.

DBMS: Data Base Management System, Sistema Manejador de Base de Datos. Es el sistema encargado de administrar, soportar y mantener la integridad de la base de datos, así como darle interfaz al usuario y acceso a la misma.

Diccionario de Datos: Un compendio de definiciones y especificaciones para las categorías de datos y sus relaciones.

Dimensión: Entidad independiente dentro del modelo multidimensional de una organización, que sirve como llave de búsqueda (actuando como índice), o como mecanismo de selección de datos. En una base de datos relacional o plana, cada campo en un registro representa una dimensión. En una base de datos multidimensional, una dimensión es un conjunto de entidades similares; por ej.: una base de datos multidimensional de ventas podría incluir las dimensiones Producto, Tiempo y Ciudad.

Drill Down: Exponer progresivamente más detalle (dentro de un reporte o consulta), mediante selecciones de tuplas sucesivamente.

Drill-Up: Es el efecto contrario a drill-down. Significa ver menos nivel de detalle, sobre la jerarquía significa generalizar o sumarizar, es decir, subir en el árbol jerárquico.

DSS: Sistema de Soporte de Decisiones. Sistema de aplicaciones automatizadas que asiste a la organización en la toma de decisiones mediante un análisis estratégico de la información histórica.

ETT (Extracción, Transformación y Transporte de datos): Pasos por los que atraviesan los datos para ir desde el sistema OLTP (o la fuente de datos utilizada) a la bodega dimensional. Extracción, se refiere al mecanismo por medio del cual los datos son leídos desde su fuente original. Transformación (también conocida como limpieza) es la etapa por la que puede atravesar una base de datos para estandarizar los datos de las distintas fuentes, normalizando y fijando una estructura para los datos. Finalmente está el Transporte, que consiste básicamente en llevar los datos leídos y estandarizados a la bodega dimensional (puede ser remota o localmente). Generalmente, para un Data Mart no es necesario atravesar por todos estos pasos, pues al ser información localizada, sus datos suelen estar naturalmente estandarizados (hay una sola fuente).

Jerarquía: Es un conjunto de atributos descriptivos que permite que a medida que se tenga una relación de muchos a uno se ascienda en la jerarquía. Por ejemplo: los Centros de Responsabilidad están asociados a un Tipo de Unidad, el cual pueden corresponder a una gerencia, subgerencia, superintendencia, etc.; por otro parte, cada CR está asociado a otro CR a nivel administrativo y, también existe una clasificación a nivel funcional.

Modelo analítico: Una estructura y proceso para analizar un conjunto de datos. Por ejemplo, un árbol de decisión es un modelo para la clasificación de un conjunto de datos.

Modelo lineal: Un modelo analítico que asume relaciones lineales entre una variable seleccionada (dependiente) y sus predictores (variables independientes).

Modelo no lineal: Un modelo analítico que no asume una relación lineal en los coeficientes de las variables que son estudiadas.

Modelo predictivo: Estructura y proceso para predecir valores de variables especificadas en un conjunto de datos.

Navegación de datos: Proceso de visualizar diferentes dimensiones, "fetas" y niveles de una base de datos multidimensional. Ver OLAP.

OLAP Procesamiento analítico on-line (On Line Analitic prossesing): Se refiere a aplicaciones de bases de datos orientadas a arreglos que permite a los usuarios ver,

navegar, manipular y analizar bases de datos multidimensionales. Conjunto de principios que proveen una ambiente de trabajo dimensional para soporte decisional.

Oltp (On-line Transaction Processing): Sistema transaccional diario (o en detalle) que mantiene los datos operacionales del negocio.

Outlier: Un conjunto de datos cuyo valor cae fuera de los límites que encierran a la mayoría del resto de los valores correspondientes de la muestra. Puede indicar datos anormales. Deberían ser examinados detenidamente; pueden dar importante información.

RAID: Formación redundante de discos baratos (Redundant Array of inexpensive disks). Tecnología para el almacenamiento paralelo eficiente de datos en sistemas de computadoras de alto rendimiento.

Regresión lineal: Técnica estadística utilizada para encontrar la mejor relación lineal que encaja entre una variable seleccionada (dependiente) y sus predicados (variables independientes).

Redes Neuronales Artificiales: Proceso de clasificación, predicción y razonamiento por medio de la simulación del cerebro humano.

Snapshot: Imagen instantánea de los datos en un tiempo dado.

SQL: Structured Query Language, Lenguaje estructurado de Consultas. Es un estándar en las bases de datos. Utilizado para obtener, por medio de enunciados, resultados extraídos de los registros contenidos en el DBMS.

Sumarización: Actividad de incremento de la granularidad de la información en una base de datos. La sumarización reduce el nivel de detalle, y es muy útil para presentar los datos para apoyar al proceso de Toma de Decisiones.

Tabla Dimensional: Dentro del esquema estrella, corresponde a las tablas que están unidas a la tabla central a través de sus respectivas llaves. La cantidad de estas tablas le otorgan la característica de multidimensionalidad a esta estrategia.

Vecino más cercano: Técnica que clasifica cada registro en un conjunto de datos basado en una combinación de las clases del/de los k registro (s) más similar/es a él en un conjunto de datos históricos (donde $k \neq 1$). Algunas veces se llama la técnica del vecino k-más cercano.