matemática: Una publicación del ICM – ESPOL
2003, Vol. 1, No. 2 Publicado:2003-10-30

ALGORITMOS DE LA MINERÍA DE DATOS

Alvarado Juan

Resumen. *Se introduce la tecnología de la Minería de datos (Data Mining), en la cual se indica sus orígenes, su propósito, se da una clasificación de sus métodos y se describe brevemente sus algoritmos mas populares. Al final se realiza un análisis de las diferencias entre la Minería de datos y la Estadística y la oportunidad que tiene la carrera de Estadística e Informática de aplicar esta tecnología en nuestro medio.*

Palabras Claves: Data Mining, Algoritmo Supervisado no Supervisado, Error de Entrenamiento de Prueba.

1. INTRODUCCION

Bajo el nombre de Minería de Datos (Data Mining) se engloban una serie de procedimientos y algoritmos matemáticos con el propósito de encontrar patrones de datos a partir de grandes volúmenes. Los patrones a ser encontrados por estos algoritmos deben tener las siguientes características:

Válidos: Basados en una cantidad aceptable de datos.

Nuevos: Patrones no esperados de encontrar

Útiles: Tengan valor para el dueño de la información.

Interpretables: Sean fáciles de interpretar y asimilar por el dueño de la información.

2. ORIGENES

La Minería de Datos no nació como una disciplina independiente, sino más bien es un supra término que representa a un agregado de diferentes tecnologías matemáticas que en los 70 y 80 se utilizaron para la extracción de patrones de datos a partir de grandes volúmenes de datos. Entre estas disciplinas se destaca la Estadística, la Inteligencia Artificial, Visualización de Datos, Redes Neuronales, Lógica difusa, etc. También en esos años se conocía a la actividad de extracción de patrones de datos como "Reconocimiento de patrones".

Pero fue en los años 90 donde estas tecnologías de reconocimiento de patrones comienzan a integrarse con las tecnologías de bases de datos, debido a que existía la necesidad de extraer información de grandes volúmenes de datos.

Estos grandes volúmenes de datos comienzan aparecer debido dos causas: al uso de tecnologías que permitían ingreso y el almacenamiento masivo de los datos: tales como códigos de barras, páginas web, RAID, etc., y al continuo avance de la automatización de los procesos de negocios por medio de los sistemas de información.

Los reportes de tipo sumarización que tenían los sistemas de información tradicionales no podían mostrar toda la información que estaba almacenada en estas grandes bases de datos, por tal motivo se hizo evidente la necesidad de usar herramientas más sofisticadas para la extracción de esta información.

3. PROPOSITO DE LOS ALGORITMOS DE LA MINERIA DE DATOS

Con los algoritmos de la Minería de Datos podemos realizar por ejemplo las siguientes tareas:

Predecir el nivel de riesgo que va a tener un cliente que aplica a una póliza de seguro.

Encontrar una forma de **clasificar** las solicitudes de crédito sobre la base del historial de créditos, de su información financiera histórica y actual, del tipo de actividad que se dedica, etc.

Dada una base de datos de 10000 e-mails **encontrar** cuales son las personas con más probabilidad que responden a una promoción por correo electrónico.

Dado un conjunto de transacciones, **encontrar** las transacciones que tienen mas probabilidad de ser fraudulentas.

Encontrar nichos de mercados y analizar su comportamiento, en base al historial de compra y a los productos **asociados** con estos clientes.

Agrupar a los clientes según su preferencia de compra, localización, nivel de instrucción, ingreso, etc.

Juan Alvarado Ortega, Ingeniero en Computación. Magíster en Administración Publica ESPOL, Profesor de la Escuela Superior Politécnica del Litoral; (e-mail: jao_ec@yahoo.com)

Áreas de Aplicación

Ejemplos de las aplicaciones de las técnicas de Data Mining son las siguientes:

Instituciones Bancarias: aprobación o rechazo de Créditos.

Marketing: Identificar a los clientes que van a ir a la competencia.

Marketing: Identificar a los clientes que van a responder a las promociones.

Marketing: Identificar grupos de clientes y ofrecer paquetes promociónales.

Detección del fraude: Telecomunicaciones, Compañías de Seguros, Instituciones Financieras.

Recaudaciones de Impuestos: Encontrar las declaraciones de Impuestos con más probabilidad de ser fraudulentas.

Bienes Raíces: Predecir el valor que va tener un inmueble dadas los tipos de inversiones que están próximas y las expectativas de nuevas inversiones.

4. EL PROCESO DE LA MINERIA DE DATOS

El proceso de aplicar los métodos de la Minería de Datos se inicia generando un Data Warehouse (Bodega de Datos) de los datos operacionales de la empresa (datos del día a día). Este proceso genera una base de datos consolidada, consistente y optimizada para ejecutar sobre ella procesos intensos de lectura.

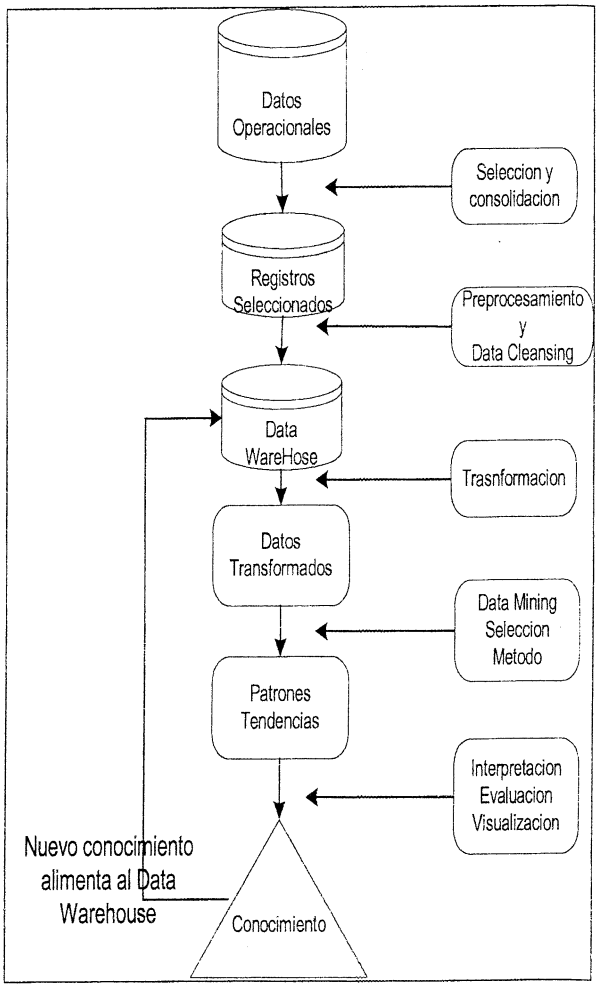
Para conseguir esta Bodega de Datos, se selecciona y se consolida los datos operacionales y se le aplica después un proceso de data cleasing (limpieza de datos inconsistente) y a continuación se transforma estos datos para que soporten procesos intensos de lectura.

Una vez obtenido el Data Warehouse, se realiza un proceso de selección de la información a ser analizada. Y opcionalmente se puede realizar algún preprocesamiento de esta información, tales como extraer la transformada de Fourier o la transformada de Wavelet (Ondeletas) de los datos. Estos tipos de procesamiento cambian la forma de la data, más no el contenido de información de la misma, y este cambio de formato facilita la ejecución del algoritmo de Data Mining.

La data seleccionada y preprocesada podría ser graficada con el objetivo de descubrir visualmente ciertas características más relevantes que tiene la data, y ayudarnos a seleccionar el algoritmo de Data Mining más promisorio para el descubrimiento de los patrones de datos. Esta selección se basa en la estructura del problema a resolver, en la forma visual de los datos y en la experiencia del usuario.

El algoritmo de Data Mining seleccionado arroja patrones de datos, que tiene que ser validado por el usuario con el propósito de juzgar si el patrón descubierto es útil para los objetivos de la organización. (Ver Figura 1)

Figura 1
Algoritmos de la minería de datos
Algoritmo de Data Mining



5. ALGORITMOS DE LA MINERIA DE DATOS

Los algoritmos de Data Mining, se dedican a encontrar relaciones ocultas que están presentes en los datos.

Se puede considerar a los algoritmos de Data Mining que cumplen la función inversa de los algoritmos de simulación matemática, ya que estos últimos parten de definiciones de alto nivel (como las distribuciones y relaciones conocidas de las variables a simular) y generan un conjunto de datos que satisfacen los requerimientos de estas variables.

Desde el punto de vista de la Estadística los algoritmos de Data Mining en su mayoría son métodos no parámetricos ya que no hacen suposiciones expresas acerca de como están distribuidos los datos.

Los algoritmos de Data Mining se dividen en dos grupos:

Algoritmos Supervisados,
Algoritmos No Supervisados

5.1 ALGORITMOS SUPERVISADOS

Los algoritmos supervisados estiman una función f que mejor asocia un conjunto de datos X (variables independientes) con un conjunto de datos Y (variables dependientes), dado un conjunto anterior de observaciones (datos a priori) ejemplos (x1, y1) , (x2, y2) ....(xN, yN).

Estos algoritmos se llaman supervisados por que tienen dos fases:

Fase de entrenamiento o supervisión
Fase Prueba

En cada fase se trabaja con un conjunto de datos diferentes: datos de entrenamiento o diseño y datos de pruebas, ambos conjuntos de datos se sacan del conjunto de datos iniciales.

En la fase de supervisión, al algoritmo se le presentan los datos de entrenamiento y éste ajusta sus parámetros internos de su modelo de tal manera que minimice el error de predicción de la variable dependiente Y.

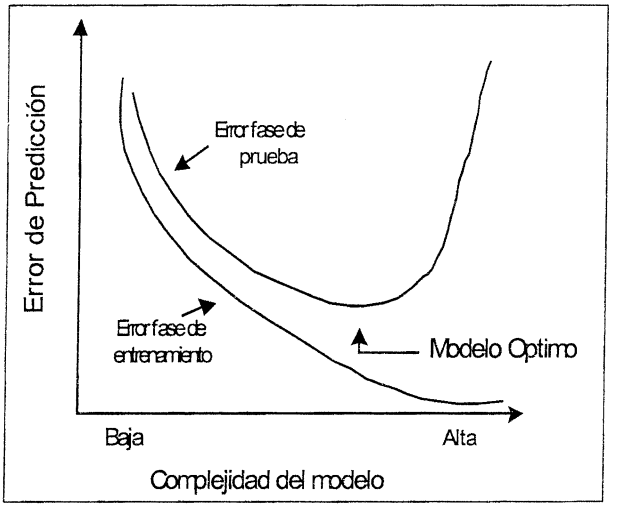
Pasada la fase de entrenamiento se aplica la fase de prueba la cual consiste en la estimación del error cometido por el modelo pero basado en los datos de prueba no en los datos usados en la etapa de supervisión. El error encontrado en la fase prueba es una aproximación más cercana al error de predicción del modelo. Y lo que se quiere hacer es encontrar modelos que minimicen el error de predicción.

La diferencia entre el error de entrenamiento y el error de prueba se debe a que en la fase de entrenamiento el modelo comienza a ajustar sus parámetros para que el error de entrenamiento sea el mínimo posible y esto hace el error de entrenamiento sea menor que el error de predicción que en el caso extremo de tener un modelo que tenga una gran cantidad de parámetros (complejidad) el error de entrenamiento puede llegar a ser cero, pero el error de la fase prueba puede llegar a tener un valor muy alto.

El problema de tener un modelo más complejo que la complejidad real de los datos, es que este modelo se va a limitar a recordar los datos de entrenamiento (de ahí que va tener un error de entrenamiento muy bajo), pero este modelo no va a realizar correctamente su trabajo el cual es predecir el valor de Y para nuevos valores X. A este tipo de modelo se dice que sobre ajustado (overfitting).

En la Figura 2 tenemos gráficos de estimaciones del comportamiento de los errores de entrenamiento y prueba bajo modelos de diferentes complejidad.

Figura 2
Algoritmos de la mineria de datos
Comportamiento de los errores



En este grafico indica que cuando el modelo tiene una complejidad baja, los errores de entrenamiento y prueba casi coinciden pero tiene un error de predicción alta, a medida que aumenta la complejidad del modelo comienza a disminuir el error de prueba (error de predicción) conjunto con el error de entrenamiento. La disminución del error de

prueba llega hasta a un determinado punto, en la cual, si aumenta la complejidad del modelo el error aumenta, y en este punto el modelo alcanza un error mínimo de prueba (predicción) el cual será la complejidad óptima del modelo.

En cambio el error de entrenamiento sigue disminuyendo a medida que se aumenta la complejidad del modelo, generando de esta forma un modelo sobre ajustado.

En términos generales los algoritmos supervisados pueden ser interpretados como algoritmos de optimización, que buscan una función f( ) que minimice cierto funcional P( ) que representa el error de predicción del modelo f( )

$$\underset{f \in V}{\text{Min}} \; P(f)$$

en la cual la región V puede ser eventualmente cualquier conjunto de funciones que puedan definirse sobre los datos de prueba.

La forma general de este funcional P( ) es la siguiente:

$$P(f) = \Sigma E(y_i, f(x_i)) + C(f)$$

Donde

y_i : son los valores de variable dependiente
x_i : son los valores de la variable independiente

f( ): la función a buscar

E(X,Y) : es una función que mide la diferencia entre X y Y.
Ejemplo : $E(X;Y) = (X - Y)^2$ ó
$E(X;Y) = |X - Y|$

C( ): es una función que penaliza la "complejidad" de la función f

La expresión $\Sigma E(y_i, f(x_i))$ se considera el error del modelo f( ) en la fase de entrenamiento.

Ejemplos de Algoritmos Supervisados

Regresión Lineal: Clasifica regiones con límites lineales.

Los K-ésimos vecinos más cercanos (K-Nearest Neighbors).

Árboles de Decisión: Clasifica regiones que pueden dividirse mediante rectángulos.

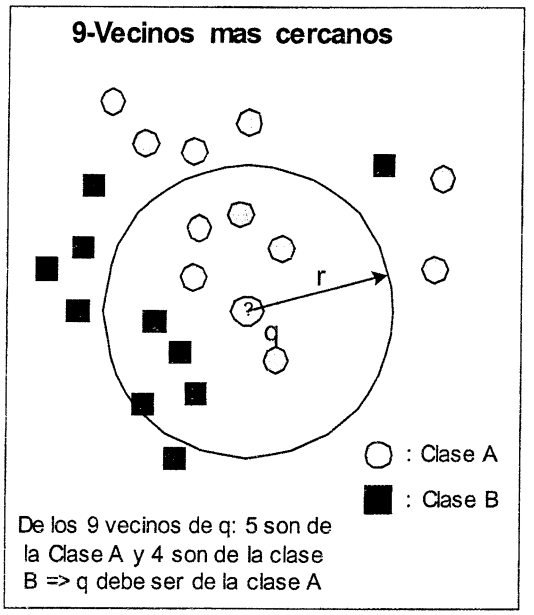
Redes Neuronales: Clasifica regiones arbitrariamente complejas.

Clasificadores Bayesianos: Define regiones de clasificación en base a la regla de Bayes.

LOS k-VECINOS MAS CERCANOS

Dado un conjunto de puntos X = {p1, p2,...pN} a los cuales se le asignado respectivamente un conjunto Y = {e1, e2, ...eN} de etiquetas e_i , las cuales indica a que clase o categoría corresponde cada punto p_i.. Dado un nuevo punto q, este algoritmo estima la clase que correspondería tener a este nuevo punto q, en base a los k puntos más cercanos a este punto q. Este decir a partir del punto q se encuentran los k vecinos más cercanos, y se cuenta cuantos pertenece a cada clase. La clase que posee más vecinos, esa es la que gana y es la clase que va a tener el punto q, como se ve en la Figura 3

Figura 3
Algoritmos de la minería de datos
k-vecinos mas cercanos



La ventaja de este algoritmo es su sencillez de implementación, ya que solo se necesita definir una función d(x,y) de distancias entre los puntos y aplicar el procedimiento de conteo de los puntos mas próximos.

Este algoritmo es un método que se basa en la información local (cercana) al punto q. Cuando se aumenta la dimensionalidad de los puntos X los vecinos cercanos al punto q comienza a distanciarse entonces comienza a aumentar el error de predicción de este algoritmo.

ÁRBOLES DE DECISION

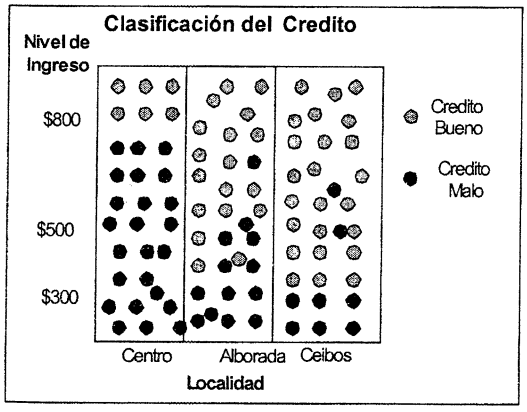
Este es un algoritmo clasificatorio, en la cual los datos de entrada X (variables independiente) se clasifican de acuerdo a regiones rectangulares, y cada región rectangular le corresponde una clase. Esta partición de forma de rectángulos genera una jerarquía de reglas simples de decisión (sobre una variable), un árbol, de tal forma de estas reglas desde la raíz del árbol hasta las hojas definen una región rectangular de decisión.

Por lo tanto los datos que son mas apropiados para aplicarse este algoritmo son los datos que se aproxima a regiones rectangulares

Por ejemplo, se tienen los datos históricos de clientes de un banco en la cual expresa la variable que indica si el crédito fue bueno o malo. Esta variable esta expresada en termino del nivel de ingreso y el domicilio del cliente.

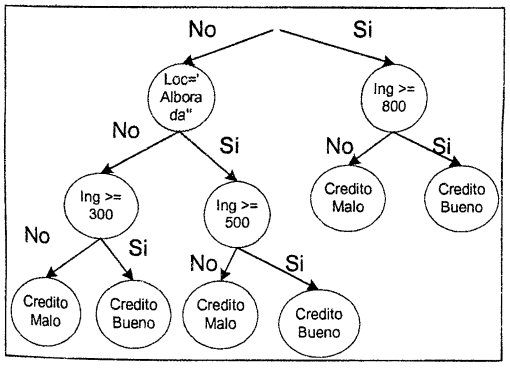
Estos datos se aproximan a regiones rectangulares, por lo tanto se puede usar los árboles de decisión, como se describe en la Figura 4.

Figura 4
Algoritmos de la minería de datos
Clasificación del crédito



El árbol que clasifica estas regiones está dado en la Figura 5:

Figura 5
Algoritmos de la minería de datos
Árboles de decisión



Este árbol puede ser representado usando reglas IF-THEN-ELSE.

IF Loc='Centro' AND Ingreso>=800 THEN Crédito Bueno ELSE Crédito Malo.

IF NOT Loc='Centro' AND Loc='Alborada' AND Ingreso>=500 THEN Crédito Bueno ELSE Crédito Malo

IF NOT Loc='Centro' AND NOT Loc='Alborada' AND Ingreso>=300 THEN Crédito Bueno ELSE Crédito Malo

Existen algunos algoritmos que construyen estos tipos de árboles, el más usado de todos es el CART (Classification And Regression Tree)

Las ventajas de los árboles de decisión es que son rápidos en su evaluación y fáciles de construir, se pueden representar usando reglas IF-THEN-ELSE lo que mejora su interpretación.

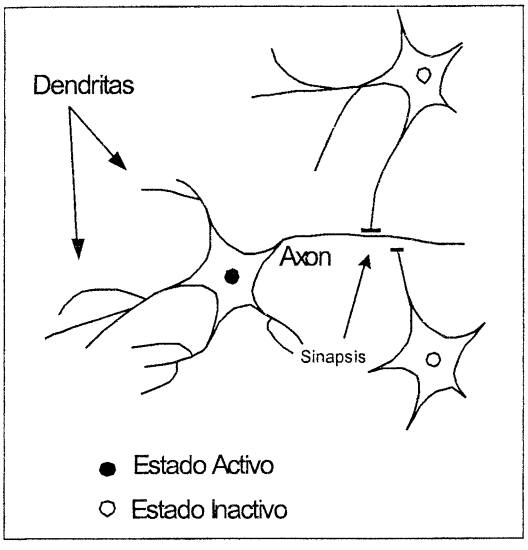
La principal desventaja de estos árboles es que sólo se aplican para regiones simples de decisión (rectángulos) y no se manejan bien cuando la data contiene valores nulos.

REDES NEURONALES

Unos de los métodos de clasificación y predicción más populares que se conocen en las herramientas de Data Mining son las redes neuronales.

Este método consiste en la imitación vía programación del comportamiento de una neurona real, la cual se porta como un dispositivo que se activa solo cuando recibe una suficiente cantidad de estímulos desde sus terminales nerviosas (dendritas), estos estímulos pueden provenir de las neuronas vecinas o de los nervios sensomotrices. Una vez que la suma de los estímulos alcanza un determinado valor, la neurona se activa y permanece en el estado binario de 1 (activo), y desde este estado comienza a estimular a otras neuronas por medio de una prolongación de la neurona llamado Axon el cual conduce el estímulo de la neurona activada, este estímulo pasa a las dendritas de las neuronas vecinas a través las sinapsis tal como se ve en la Figura 6.

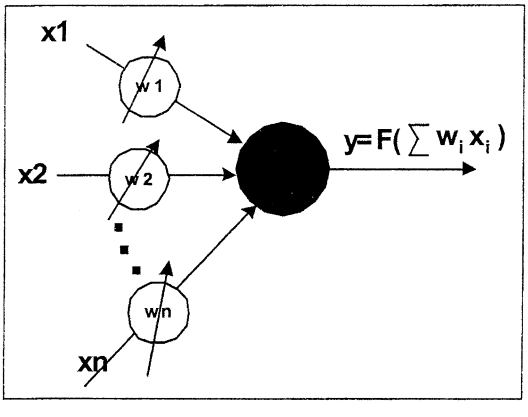
Figura 6
Algoritmos de la minería de datos
Neurona real



Bajo este modelo, una neurona tiene la capacidad de activar a otras neuronas dependiendo de cuanto líquido neurotransmisor exista en la sinapsis, el cual va tener la capacidad de aumentar o disminuir el estímulo inicial.

Una neurona real se representa en forma artificial en el siguiente diagrama (Figura 7)

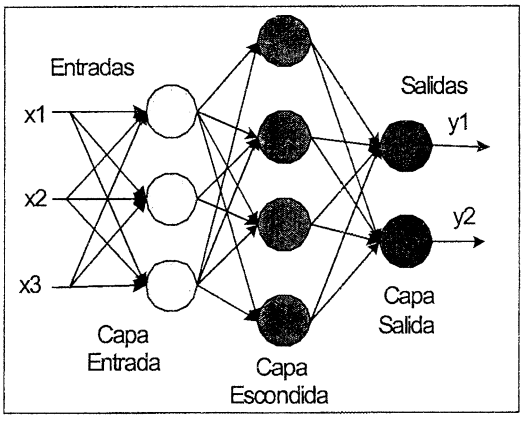
Figura 7
Algoritmos de la minería de datos
Neurona artificial



Los valores x1..xn, representan los estímulos que llegan a la neurona y los pesos w1..wn representa el poder de amplificación que existen de los estímulos en las sinapsis. El producto interior del vector estímulo (X) y del vector pesos (W) representa la suma total de los estímulos que le llegan a la neurona. Esta cantidad de estimulo debe transformarse mediante la función de activación g( ) en la repuesta final de la neurona Activa (1) o Inactiva (0).

La salida de una neurona puede servir de entrada a otras neuronas y las salidas de estas neuronas a su vez pueden ser entradas de otras neuronas, y ese conjunto de neuronas interconectadas es lo que forma una red neuronal. Como se describe en la Figura 8.

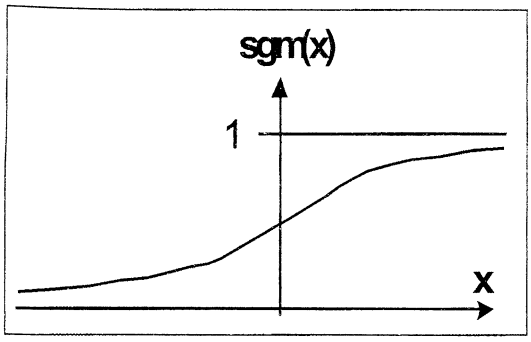
Figura 8
Algoritmos de la minería de datos
Red neuronal



Existen algunos tipos de redes neuronales: el más usado es el FFNN (Feed Foward Neural Network) En este tipo de redes las neuronas se conectan en forma acíclica, de tal forma que vayan formando capas de neuronas conectadas de forma secuencial a otras capas de neuronas, es decir las neuronas de la capa i, puede recibir las señales de las neuronas de la capa i-1 o inferiores, y las señales que salen de la capa i la pueden recibir las neuronas de la capa i+1 o superiores. Bajo este esquema la primera capa en recibir las señales de entrada se llama la capa de entrada y la última capa en recibir las señales de entradas de las otras capas se llama la capa de salida, el resto de capas se llaman capas escondidas (hidden layers), tal como se ve en la figura anterior.

La función g( ) de la neurona representa el umbral de cambio de la neurona de activo a inactivo. Inicialmente para la función g( ) se utilizaba la función escalón (u(x) = 1 para x >0 y 0 para x <= 0) la cual, representaba el cambio de inactivo a activo de la neurona de forma súbita. Pero esta función u(x) no posee la característica de ser diferenciable en todos los puntos, la cual es una característica que necesitan los algoritmos que modifican (entrenan) a la red neuronal. Es por eso que como función de activación g( ) la función sigmoidal g(x) = sgm(x) = $1/(1 + e^{-y})$ o cualquier función que tenga la forma de S. (Figura 9)

Figura 9
Algoritmos de la mineria de datos
Función de activación



Estas funciones de tipo S deben ser diferenciables en todos los puntos y realizar la transición en forma continua del estado Inactivo (0) al estado Activo de la neurona(1).

Una red neuronal se puede considerar como un modelo de regresión generalizado, la cual trata de explicar un determinado conjunto de valores de salida Y en función de un conjunto de valores de entrada X . Este modelo tiene como parámetros el conjunto de todos los pesos Wi que tiene la red neural, mientras la red neuronal tenga más pesos, el modelo a ajustar será más complejo.

El algoritmo mas popular para ajustar a una red neuronal del tipo FFNN es el algoritmo de contrapropagación (backpropagation). Este algoritmo se basa en la búsqueda por medio del método de la gradiente de los pesos óptimos W* que hagan mínimo el error de entrenamiento, el cual es el error cuadrático entre la salida que arroja la red y_i a partir de sus datos de entrada x_i y la repuesta deseada d_i del supervisor, es decir: $\Sigma (y_i - d_i)^2$

Los pasos que tiene que ejecutar este algoritmo son los siguientes:

1.- Inicialice los pesos Wi con pequeños valores aleatorios.

2.- Escoja aleatoriamente un dato de entrada X (estos vienen del conjunto de datos de entrenamiento).

3.-Propagar hacia delante las señales de entrada por todas las capas de la red de acuerdo con la expresión $x_i^{l+1} = g(\Sigma w_{ij} x_j^l)$, en la cual salida de la capa l será la entrada de la capa l+1.

4.- Calcular los deltas δ_i^L de cada neurona i de la capa de salida L.

$$\delta_i^L = g'(h_i^L) * (d_i^L - y_i^L)$$

Donde h_i^l representa la entrada total sobre la neurona i de la capa l, y g´ es la derivada de función de activación g( ) (de ahí que se necesita de una función de activación g( ) que sea diferenciable en todos los puntos).

5.- Calcule el error cuadrático obtenido por la red $E = \Sigma (\delta_i^L)^2$. Si este error es menor a cierto valor E_{max} o si se ha cumplido un número máximo de iteraciones entonces haga FIN en caso contrario continúe con el siguiente paso.

6.- Calcular los deltas de las capas interiores de la red, por medio de la propagación hacia atrás de los deltas encontrados en la capa de salida:

$$\delta_i^l = g'(h_i^l) * \Sigma w_{ij}^{l+1} \delta_i^{l+1}$$ Para l = (L-1), ... ,1

6.- Actualización de los pesos existentes usando el delta de los pesos

$$\Delta w_{ij}^l = \eta y_i^{l-1} \delta_i^l$$ Donde η es la constante de aprendizaje del algoritmo

7.- Vaya al paso 2

Ventajas y Desventajas de las Redes Neuronales

En términos generales las redes neuronales tienen la ventaja de poder clasificar regiones de extrema complejidad con la que se pueden encontrar cuando se requiere reconocer una voz o una imagen. Además, las redes neuronales pueden manejar una gran cantidad de atributos, y una vez que la red neuronal esta entrenada, es decir, están especificados sus pesos su ejecución en base a los datos de entrada, es bastante rápida.

La desventaja principal de las redes neuronales es que el conocimiento de la red viene expresado en forma de pesos numéricos lo cual hace difícil su interpretación por parte del usuario. Otra desventaja es el tiempo largo de entrenamiento que necesita el algoritmo de Backpropagation.

CLASIFICACION BAYESIANA

Este tipo de clasificador esta basado en la regla de Bayes

$$p(y_j \mid x) = \frac{p(x \mid y_j)\, p(y_j)}{p(x)}$$

Donde:
$p(y_j \mid x)$ = probabilidad que ocurra la clase y_j dado el dato x (distribución a posteriori)
$p(x \mid y_j)$ = probabilidad que el dato x ocurra dado que venga de la clase y_j (distribución a priori)
$p(y_j)$ = probabilidad de ocurrencia de la clase y_j,
$p(x)$ = probabilidad de ocurrencia del dato x

El valor y_j (y = j) dado el valor x se considera que es una variable aleatoria que tiene una distribución condicional $p(y_j \mid x)$, pero esta distribución tiene valores de $p(y_1 \mid x)$, $p(y_2 \mid x)$ ... $p(y_n \mid x)$ tanto como posibles resultado tenga la variable y. Entonces la predicción de un valor y_j dado el valor x se lo obtiene seleccionado el índice i que de el mayor posible para la probabilidad $p(y_j \mid x)$ es decir se selecciona la clase y_j más probable dado un valor de x

$y_j = f(x) \equiv x \in y_j \equiv p(y_j \mid x) >= p(y_i \mid x)$
Para todo i , j

Teóricamente dada cualquier distribución conjunta de X y Y en la cual existe una dependencia desconocida entre las variables X y Y, el clasificador bayesiano es el clasificador que va tener el menor error en la predicción de Y dado X. Y=f(X).

El error del clasificador bayesiano viene dado por la expresión:

$$E_B = \int 1 - \max_k(p(y_k \mid x)) \, p(x) \, dx$$

Este error viene a ser una cota inferior para los errores de predicción de cualquier clasificador. El clasificador bayesiano se considera entonces el mejor clasificador posible.

Inclusive se ha llegado a demostrar [5] que el clasificador de los k-vecinos más cercanos se aproxima asintóticamente al clasificador bayesiano cuando el número de datos N tiende a infinito, igual resultado se ha demostrado para las redes neuronales[6].

Trabajar con clasificadores bayesiano sería lo ideal pero el problema que es difícil estimar adecuadamente las probabilidades a priori ($p(x \mid y_j)$) a partir solo de los datos de prueba. Esto hace que se tenga que o usar métodos clasificatorios alternativos tales como redes neuronales, árboles de decisión, etc. o asumir ciertos supuestos sobre la distribución a priori que simplifica su estimación. El segundo caso es lo que conoce como clasificador bayesiano simplificado (Naive Bayes)

Clasificador Bayesiano Simplificado (Naive Bayes)

Este clasificador asume que los atributos de X (x1, x2, .. xn) son variables aleatorias independientes por lo tanto se puede estimar la distribución a priori $p(x \mid y_j)$ de la siguiente forma

$$p(x|y_j) = p(x_1|y_j) * p(x_2|y_j) * ....* p(x_n|y_j)$$

Cada uno de los $p(x_i|y_j)$ puede ser estimado por medio de un histograma de los valores x_i para cada clase y_j . y este histograma puede ser fácilmente calculado a partir de los datos de prueba.

5.2 ALGORITMOS NO SUPERVISADOS

Dado un conjunto de variables aleatorias x1, x2, .. xN para los cuales no existe ninguna variable Y que clasifique a estas variables. Entonces sólo se puede aplicar los algoritmos de tipo no supervisado los que encarga de estimar o de explorar ciertas propiedades de la distribución conjunta de x1, x2,... xN es decir P(x1, x2,... xN)

Ejemplos de Algoritmos No Supervisado

Reglas de Asociación
Agrupamiento (Clustering)

REGLAS DE ASOCIACION

Los almacenes minorista siempre están interesados en asociaciones entre diferentes productos que compra los clientes.

- Alguien que compra pan es probable que compre leche.
- Una persona que compra vodka es probable que compre jugo de toronja.

El conocimiento de estas asociaciones permite que los proveedores mejore su oferta a los clientes, ofreciendo productos que están relacionados con su necesidad actual; Ej. Cuando un cliente compra un determinado libro por Internet, la página Web le puede sugerir otros libros asociados con el primer libro.

Las reglas de asociación vienen expresadas en términos de antecedente y consecuente.

Antecedente $\Rightarrow$ consecuente
Pan $\Rightarrow$ Leche
Vodka $\Rightarrow$ Jugo Toronja

Una regla de asociación se encuentra a partir de los registros guardados en la base de datos y con un nivel de soporte y de confianza definidos por el usuario.

Soporte: es una medida que indica la fracción de la población de registros satisfacen al antecedente y consecuente de la regla.

—Ej. Suponga que el 0.001% de todas las compras incluye leche y jabones. El soporte de la regla leche ⇒ jabón *es* bajo.

—

—Se desea tener reglas con alto nivel de soporte
•Reglas con bajo nivel de soporte por lo general no son muy útiles.

Confianza: es la probabilidad de que el consecuente sea verdad dado que ocurrió el antecedente.

Pr(Concecuente|Antecedente) =

$$\frac{Pr(\text{consecuente} \wedge \text{Antecedente})}{Pr(\text{Antecedente})}$$

—Ej. La regla pan ⇒ leche *tiene una confianza* del 80%. El 80% de las compras que incluye pan también incluye leche.

—Se desea tener reglas con alto nivel de confianza.

Ejemplo

Transaccion Id	Items Comprados
1	{Leche Fideo Queso}
2	{Carne, Fideo, Tomate}
3	{Leche, Azúcar, Pan}
4	{Tomate, Pan, Queso}
5	{Tomate, Queso, Leche}
6	{ Leche, Tomate, Pan}
7	{Leche, Fideo, Pan}

Para el par (Leche-Pan) el soporte es de 3/7 = 42%, ya que en sólo 3 transacciones aparecen la combinación pan y leche.

La confianza para Leche => Pan es Soporte({Leche,Pan})/Soporte({Leche})=(3/7) /(5/7)=60%

La confianza para Pan => Leche es Soporte({Leche,Pan})/Soporte({Pan})=(3/7) /(4/7)=75%

Entonces podemos construir las siguientes reglas:

Leche => Pan con 42% soporte y 60% confianza.

Pan => Leche con 42% soporte y 75% confianza.

Búsqueda de reglas de asociación

Para encontrar estas reglas primero tiene que encontrase los conjuntos de valores x1, x2,..xM que cumple con un determinado nivel de soporte dado por el usuario. Esto define los conjuntos de **I**tems **M**ás **F**recuentes (IMF). El Algoritmo a priori se encarga de encontrar los IMF de una Base de Datos.

Una vez encontrado los IMF. Para cada IMF se busca algún par de subconjuntos de IMF A y B, tal que la regla A => B cumpla con un nivel de confianza dado por el usuario

Algoritmo a priori

Fk : Conjunto de los IMF de tamaño k
Ck : Conjunto de los candidatos a IMF de tamaño k

F1 = {Todos los items}
Para (k=1; Fk != 0; k++)
Haga
{
Ck+1 = Generar nuevos candidatos desde Fk, **para cada** Registro t en la base de datos.**Haga** Incremente el Soporte de cada uno de los candidatos en Ck+1 que contienen a t Fk+1 = Candidatos in Ck+1 con un Soporte mínimo
}
Repuesta = La unión de todos los Fk

Este algoritmo encuentra los IMF, que son los picos modales de la distribución conjunta de las variables aleatorias x1, x2, ... xN que representa los N productos que puede ofrecer un almacén minorista.

Los IMF puede interpretarse como una canasta básica de productos que requiere un determinado segmento de mercado.

A partir de un IMF del tipo {X1, X2, .. Xm} se construye las siguientes reglas de asociación, las cuales tienen un determinado soporte.

$X_{i1}, X_{i2}, .. X_{ia} \Rightarrow X_{j1}, X_{j2} .. X_{ic}$

Tal que el conjunto de índices del antecedente {i1, ... ,ia} sea disjunto con el conjunto de índice del consecuente {j1, .., jc} y que la unión de ambos conjuntos de el conjunto de índices totales {j1, .., jm}.

A partir de este conjunto de reglas se calcula el nivel de confianza que tiene cada una de las

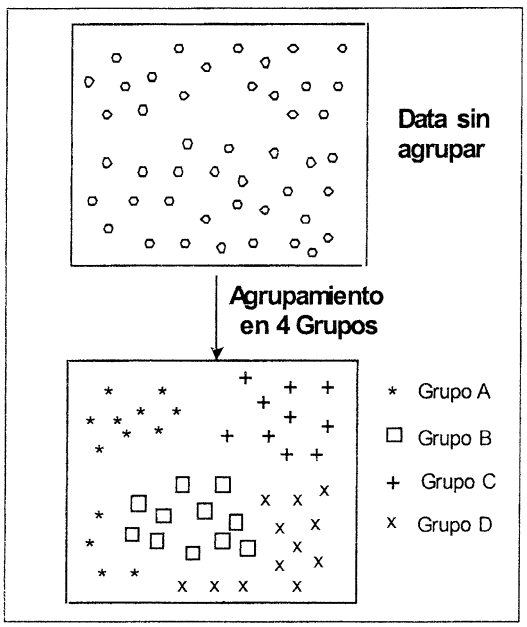
reglas, y se selecciona las reglas que tiene el nivel de confianza especificado por el usuario.

Las reglas mas útiles son las que nos dicen relaciones nuevas entre los datos.

AGRUPAMIENTO (Clustering)

Clustering: Construye grupos de datos a partir de un conjunto de datos de tal forma que los datos que pertenezcan a un mismo grupo (cluster) sean mas parecido entre ellos que los datos que pertenecen a clusters diferentes. Tal como se ve en Figura 10.

Figura 10
Algoritmos de la mineria de datos
Agrupamiento



Para lograr este agrupamiento se debe utilizar algún tipo de métrica (distancia) d(x,y) la cual indica el grado de parecido que va haber entre un dato y otro.
En términos generales el problema de cluster se puede plantear de la siguiente forma:

Dado un número entero k y dado un conjunto de puntos D con una métrica d(x,y). Encuentre una partición C1, C2, ... Ck para el conjunto D de tal forma que se minimice la expresión:

$$W(c) = \sum_{i=1}^{k} \sum_{xi,\, xj \in Ck} d(xi, xj) \qquad (1)$$

Esta cantidad representa la suma total del grado de diferencia que tiene los elementos de un mismo cluster.

Esta última expresión es equivalente a la expresión:

$$W(c) = \sum_{i=1}^{k} \sum_{xi \in Ck} d(x_i, c_k) \qquad (2)$$

donde c_k es el vector medio o centroide asociado con el k-ésimo cluster, que podría interpretarse como el centro geométrico del cluster

$c_k = (1/nk) * \Sigma\ x_i$

donde nk es el número de elementos dentro del cluster k y $xi \in Ck$

Minimizar en forma directa esta expresión es complicada, ya que los espacios de búsqueda de esta expresión son combinatorios.

Por lo tanto se aplica un procedimiento de búsqueda heurístico llamado k-means, el cual es un procedimiento que inicialmente elige en forma aleatoriamente k centroides y a partir de estos centroides se agrupan los datos que están más cerca a cada centroide y se forman los cluster, y una vez formado cada cluster, se vuelve a calcular los centroides de cada cluster, y estos servirán para selección de otros clusters en la siguiente iteración. El objetivo de este algoritmo es ir modificando la posición inicial de cada centroide de tal forma que no exista cambio de cluster cada vez que se calcule un nuevo grupo de centroide.

Los pasos de este algoritmo son los siguientes:

```
For i=1 To k
  Elija aleatorimante k puntos r(i) desde el
conjunto D
Endfor

While (Exista cambios en los cluster Ck )

   Formar los nuevos clusters
   For i=1 To k
      Ci = { x ϵ D | d( r(i), x) <= d(r(j), x)
           para todo j=1,..k j=I }
   Endfor

  Con los nuevos cluster calcule los nuevos
centroides

   For i=1 To k

      r(i) = (1/nk) * Σ xi donde nk es el
      numero de elementos dentro del cluster k
      y xi ϵ Ck

   Endfor

End
```

6. DIFERENCIAS ENTRE LA MINERIA DE DATOS Y LA ESTADISTICA

La diferencia principal entre la Estadística y la Minería de datos es la naturaleza de los datos que procesa:

Mientras para la Estadística los datos provienen por lo general de muestras aleatorias de una población, cuya distribución se desconoce. Los datos que procesa los algoritmos de la Minería de Datos no provienen de muestras aleatorias sino que son datos que fueron ingresados en un determinado periodo de tiempo para un propósito diferente que el análisis de los datos, por ejemplo datos de Facturación, Consumos, Pedidos de Compra, etc. Estos datos sirven principalmente para mantener operando a las empresas. Estos datos no son muestra aleatorias sino se puede considerar como muestras de "oportunidad" o de "conveniencia"

Los algoritmos de Data Mining deben tener la capacidad de manejar grandes volúmenes de datos, es por eso que algunos algoritmos clásicos de la Estadística se han tenido que rehacerse pensando en manejar grandes volúmenes de datos, ejemplo: El algoritmo de agrupamiento clustering (k-.MEANS) es muy costoso ejecutarlo cuando se tiene que procesar bastantes datos con muchas dimensiones y para tal efecto se ha creado versiones escalables de este algoritmo que realizan la tarea de agrupamiento en menor tiempo para un mayor volumen de datos, ejemplos de estas versiones son CLARANS, DBSCAN, BIRCH, CLIQUE CURE, ROCK

7. HERRAMIENTAS COMPUTACIONALES PARA TRABAJAR EN LA MINERIA DE DATOS

En el mercado comienza aparecer productos que tiene incorporado en forma nativa los métodos de minería de datos. Ejemplos de estos productos son los siguientes:

El producto Oracle 9i tiene los siguientes algoritmos:

- Adaptive Bayes Network (árboles de decisión)
- Naive Bayes (Clasificador Bayesiano simplificado)
- *k*-Means (Clustering)
- O-Cluster (Clustering jerárquico)
- a priori (Reglas de asociación)

La marca SPSS tiene una línea de producto llamada CLEMENTINE la cual tiene un lenguaje gráfico para estructurar de forma integrada todas las fases del proceso que se realizan en un proyecto de la Minería de Datos. Este lenguaje tiene iconos que sirven para configurar cada paso del proceso de Data Mining .

También en Internet existen librerías gratuitas de clases desarrolladas en Java tales como las librerías WEKA y XELOPES que permite el desarrollo de aplicaciones de Minería de Datos basados en Java.

La lista de herramientas existentes en la cual se puede realizar trabajos en Minería de Datos es muy amplia, una referencia mas extensa a cerca de estos productos se la encuentra en [8].

8.- CONCLUSIONES

1. La unión de esta técnicas de reconocimientos de patrones con las tecnologías de bases de datos es lo que se ha venido a llamar como Minería de datos. Con esta integración las técnicas de reconocimiento de patrones pueden hacer uso de las tecnologías que tiene los motores de bases de datos para manejar un volumen de datos muy grande en un tiempo razonable de procesamiento.

2. Como podemos ver, la minería de datos es una mezcla compacta de los que es la Estadística y la Informática, debido a esto el perfil de la carrera de ingeniería en Estadística e Informática, está llamada a dominar a esta tecnología en nuestro medio más que cualquier otra carrera tradicional de computación, ya que para la formación de buenos mineros de datos se necesita sólidas bases de Estadística y de Informática.

3. Y debido al éxito que esta teniendo las aplicaciones de la minería de datos en otros países tarde o temprano se va popularizar su uso en nuestro medio y va a ser una excelente oportunidad para un Ingeniero en Estadística e Informática posicionarse como un minero de datos.

REFERENCIAS BIBLIOGRAFICAS

1. **HAND D., MANNILA H. y SMYTH P.** *"Principles of Data Mining"* MIT Press. 2001

2. **PRUDSYS**: Documentación de la librería de clases XELOPES. 2002

3. **WIDROW B.** *"30 Years of Adaptive Neuronal Networks: Perceptron, Madline, and Backpropagation"* Proceedings of the IEEE September 1990

4. **JAIN ANIL K.**. *"Artificial Neuronal Networks: A tutorial"* Computer IEEE Maro 1996

5. **COVER T. y HART P.**. *"Nearest Neighbor Patter Clasification"* IEEE trans Inf. Teory Vol 13 pp 21-27 1967

6. **LIPPMAN R.**. *"An Introduction to Neuronal Computing"* IEEE ASSP Magazine pp 4-22 Abril 1987

7. **BEZDEK J.** *"Computing with Uncertainty"* IEEE Communications magazine Septiembre 1992

8. **KDNUGGETS**. (2003). *"Data Mining, Knowledge Discovery"*, Genomic Mining, Web Mining http://www.kdnuggets.com/ . (Abril 2003)