Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

Una Nueva Técnica para Clasificación Utilizando Teoría de Patrones

Tesis que presenta:

Luis Eulalio Real Novo

para obtener el grado de:

Maestro en Ciencias

en la especialidad de:

Ingeniería Eléctrica

Directores de Tesis

Dr. Ricardo Vilalta López
Dr. Félix Francisco Ramos Corchado

Guadalajara, Jalisco, Agosto de 2009.

Una Nueva Técnica para Clasificación Utilizando Teoría de Patrones

Tesis de Maestría en Ciencias
Ingeniería Eléctrica

Por:
Luis Eulalio Real Novo
Ingeniero Electrónico en Computación
Centro de Enseñanza Técnica Industrial 1999-2004

Becario de Conacyt, expediente no. 227322

Directores de Tesis
Dr. Ricardo Vilalta López
Dr. Félix Francisco Ramos Corchado

CINVESTAV del IPN Unidad Guadalajara, Agosto de 2009.

Agradecimientos.

A Dios a quien dedico este trabajo y debo todo lo que soy y lo que tengo.

A mi esposa Alejandra que es mi impulso y fuente de energía.

A mis padres y hermanos por el ánimo y la confianza.

A mis asesores Ricardo Vilalta y Félix Ramos por su guía y ayuda en este trabajo.

A mis compañeros de computación por el apoyo durante esta maestría.

Al CONACYT por el apoyo económico recibido.

Resumen.

En este documento se demuestra que se puede utilizar la Teoría de Patrones como método de aprendizaje supervisado, proponiendo un método para clasificar datos en conjuntos donde se encuentran mapeos de paridad en presencia de ruido o ausencia de datos en el conjunto de entrenamiento.

Esto se hace por que los métodos actuales de Reconocimiento de Patrones, no son capaces de trabajar eficientemente en el caso dado cuando el patrón de paridad está completo, obteniéndose el peor desempeño en algoritmos del estado del arte, tales como c4.5 o backpropagation. Que además son incapaces de generalizar un concepto cuando se tiene ruido o el patrón está incompleto en el conjunto de entrenamiento. El método propuesto consta de modelo y algoritmo. Para el modelo, se utilizo únicamente Teoría de Patrones y la implementación del algoritmo se desarrolló en matlab. Así podemos clasificar datos hasta con un setenta y cinco por ciento del patrón de paridad.

Utilizando la metodología propuesta se pueden resolver problemas en el ámbito médico, financiero, económico, industrial y cualquier otra área en la que se aplican los demás métodos de aprendizaje de Reconocimiento de Patrones. Ya que aunque el caso de estudio involucra bases de datos binarias, en información real existen comportamientos de paridad en bases de datos reales o enteros, que si no se pueden ver a primera vista, se pueden localizar modelando la información de la manera adecuada.

Abstract.

This thesis proves that Pattern Theory can be used as a supervised learning method, proposing a new method for classify data in datasets where parity mapping is found, in presence of noise or incomplete patterns in the training set.

This is done because none of the actual methods of Pattern Recognition can efficiently deal with datasets where the mentioned problem is present, when a complete parity mapping is found the worst performance is obtained from the state of the art methods that can acquire the concept, such as C4.5 and backpropagation. In addition these methods cannot generalize the concept if the parity pattern is incomplete or when noise is present in the training set. The proposed method it is composed of a Pattern Theory based model and an algorithm developed that is implemented in matlab. This way we can classify data up to 75 percent of the parity pattern.

Using the proposed methodology, problems can be solved in medical, financial, economics, industry and any other environment where Pattern Recognition methods can be applied. Although the study case involves binary datasets, in real datasets exist parity behavior on real or integer data, that if cannot be seen at first sight, it can be found modeling the information in the right way.

Índice.

1. Introducción. 1
1.1. Descripción del problema. 1
1.2. Objetivos. 2
1.3. Propuesta. 2
1.4. Estructura del documento. 2

2. Estado del Arte. 4
2.1. Introducción. 4
2.2. Redes Bayesianas de creencia. 4
2.3. Algoritmo C4.5. 6
2.4. Algoritmo Backpropagation. 9
2.5. Teoría de Patrones. 11
2.6. Conclusiones. 12

3. Propuesta. 13
3.1. Introducción. 13
3.2. Un modelo para describir la paridad. 14
3.3. Generalización del modelo de paridad. 17
3.4. Refinamiento del modelo de paridad generalizado. 19
3.5. Conclusiones. 21

4. Implementación y experimentos. 22
4.1. Introducción. 22
4.2. Algoritmo. 22
4.3. Experimentos en bases de datos artificiales. 25
4.4. Experimentos en bases de datos reales. 27
4.5. Discusión de los resultados. 30

5. Conclusiones y trabajo futuro. 32
5.1. Conclusiones. 32
5.2. Trabajo futuro. 33

Bibliografía. 34

Lista de tablas.

3.1. Universo para patrón de paridad con tres características. 14

3.2. Creación del conjunto de generadores a partir de la partición inicial. 16

3.3. Relación entre el número de características y la cantidad de objetos en el universo de paridad. 17

4.1. Estadísticas de los experimentos en bases de datos artificiales. 26

4.2. Primera interpretación para el atributo 1. 28

4.3. Primera interpretación para el atributo 2. 28

4.4. Primera interpretación para el atributo 3. 28

4.5. Estadísticas de los experimentos en bases de datos reales con la primera interpretación. 28

4.6. Segunda interpretación para los tres atributos. 29

4.7. Estadísticas de los experimentos en bases de datos reales con la segunda interpretación. 30

Lista de figuras.

2.1. Red Bayesiana de compañía aseguradora. 5

2.2. Red Bayesiana para patrón de paridad de tres características. 6

2.3. Desarrollo de un árbol de decisión usando el algoritmo C4.5. 7

2.4. Podado de un árbol de decisión por elevación de subárbol. 8

2.5. Red neuronal artificial de tres capas, con cinco entradas y una salida. 10

3.1. Estructura de generadores en Teoría de Patrones. 14

3.2. Partición inicial para dominios de tres características. 15

3.3. Conjunto de generadores para dominios de tres características. 16

3.4. Recreación del patrón de paridad sobre la gráfica de la base de datos. 17

3.5. Estructura de generadores para patrón de paridad generalizado. 18

3.6. Grafos de 2, 3 y 4 dimensiones para el modelo de paridad generalizado. 19

3.7. Partición inicial para el modelo de paridad generalizado simple. 20

3.8. Grafos de 2, 3 y 4 dimensiones para el modelo de paridad generalizado simple. 20

4.1. Algoritmo para clasificación mediante Teoría de Patrones. 23

4.2. Ejemplo de conexión entre generadores del vecindario. 24

4.3. Tipos de conexiones entre generadores. 24

4.4. Ejemplo de conexión de X hacía un vecino. 25

4.5. Resultados de los experimentos en bases de datos artificiales. 27

4.6. Resultados de los experimentos en bases de datos reales; primera interpretación. 29

4.7. Resultados de los experimentos en bases de datos reales; segunda interpretación. 30

Capítulo 1
Introducción

El objetivo de este capítulo es introducir al lector en el trabajo realizado, iniciando con la presentación del problema, luego brindando un resumen que describa la situación actual en el ámbito de Reconocimiento de Patrones, seguido por los objetivos de la tesis, una visión general de la propuesta y para finalizar, la estructura del presente documento.

1.1. Descripción del problema.

Entre la diversidad de clasificadores para aprendizaje supervisado de los que disponemos en Reconocimiento de Patrones podemos encontrar todo tipo de herramientas matemáticas, desde estadísticas como la teoría de decisión de Bayes y métodos estocásticos hasta árboles de decisión y redes neuronales. Sin embargo no existe hasta el momento un clasificador que utilice Teoría de Patrones. En esta tesis se demuestra que utilizando la Teoría de Patrones definida por Ulf Grenander [25], podemos clasificar objetos en bases de datos que no son clasificables por los métodos existentes.

Para demostrar esto, el caso de estudio elegido es el problema de paridad. Este problema esta definido por la regla de paridad [31], que se establece como sigue: Para un objeto que puede ser representado en un vector de características binarias y que pertenece a una de dos clases posibles, suponiendo que a cada característica y a la clase les asignamos valores de 0 o de 1, si una cantidad par de características es igual a 1 la clase a la que pertenece el objeto es 0, mientras que si la cantidad de características igual a 1 es impar la clase es 1, así tomando los valores de las características y el de la clase, siempre se tiene una cantidad par de valores 1. En una base de datos con objetos de dos características esta regla define la tabla de verdad del operador XOR lógico.

Cuando este comportamiento se encuentra en una base de datos, se dice que existe un mapeo de paridad, también llamado patrón de paridad. Este patrón representa un problema para los clasificadores del estado del arte de Reconocimiento de Patrones, ya que solo unos cuantos (ej. backpropagation, c4.5) pueden adquirir el patrón de paridad completo y generalizar los conceptos inherentes al conjunto de datos, asimismo se obtiene el peor desempeño en tiempo de procesamiento dado el caso. Además si el patrón de paridad está incompleto o existe ruido en los datos de entrenamiento, incluso estos métodos son incapaces de generalizar el concepto.

En este documento se presenta un clasificador que trabaja eficazmente con bases de datos en donde existe el comportamiento antes mencionado; basado en un modelo de Teoría de Patrones.

1.2. Objetivos.

El clasificador propuesto en este trabajo no pretende reemplazar a los clasificadores actuales, en su lugar extiende el campo de problemas que pueden ser resueltos por Reconocimiento de Patrones. Los objetivos son los siguientes:

1. Demostrar que utilizando Teoría de Patrones podemos diseñar clasificadores precisos para aprender conceptos que no pueden ser adquiridos otros clasificadores.
2. Desarrollar un clasificador para predecir la clase de un objeto perteneciente a un conjunto de datos en los que se presenta un patrón de paridad.
3. Desarrollar e implementar un algoritmo para mostrar la aplicación del método propuesto.

1.3. Propuesta.

La propuesta tiene dos partes principales, la primera es el modelo que permite identificar el patrón de paridad en un conjunto de datos utilizando Teoría de Patrones, el segundo es el algoritmo que implementa dicho modelo.

En el modelo se explica como están definidas las reglas para obtener el patrón que estamos buscando y el grafo que debe ser seguido para determinar la presencia del patrón.

El algoritmo presenta el uso del modelo en el clasificador y demuestra su aplicación para encontrar mapeos de paridad en bases de datos.

1.4. Estructura del documento.

La estructura del manuscrito es la siguiente:

Capítulo 2 – Estado del Arte, presenta los clasificadores más utilizados y actuales para aprendizaje supervisado, que son capaces de trabajar en conjuntos de datos en donde existen mapeos de paridad.

Capítulo 3 – Propuesta, explica el desarrollo del modelo y su aplicación al problema de paridad.

Capítulo 4 – Implementación y experimentos, muestra el diseño del algoritmo y su implementación, además de presentar los resultados obtenidos.

Capítulo 5 – Conclusiones y trabajo futuro, discute los resultados y expone algunas extensiones útiles para la propuesta.

Capítulo 2
Estado del Arte

En este capítulo se revisan los métodos de Reconocimiento de Patrones mas utilizados actualmente, además se brindan ejemplos de cómo responden cuando se encuentran patrones de paridad en los datos a clasificar. A su vez se habla sobre la Teoría de Patrones que no se utiliza hasta el momento para hacer clasificación.

2.1. Introducción.

Los métodos de Reconocimiento de Patrones mas usados actualmente utilizan todo tipo de herramientas matemáticas y algunos aunque no son tan recientes como otros, no han sido desplazados como es el caso de los clasificadores que utilizan la teoría Bayesiana, como el método de Bayes simple [5] (*Naivë Bayes*) o las redes Bayesianas de creencia (*Bayesian Belief Networks*). Lo que hacen es establecer un patrón estadístico en los datos en base al conjunto de entrenamiento dado. Otros desprecian la probabilidad de ocurrencia y trabajan directamente sobre la estructura de los datos.

En este capítulo revisaremos las Redes Bayesianas de creencia (Bayesian Belief Networks) dentro del grupo de métodos que explotan la estadística, así como los Árboles de Decisión y las Redes Neuronales Artificiales como métodos de clasificación que explotan la estructura de los datos de entrenamiento.

2.2. Redes Bayesianas de creencia.

Las redes Bayesianas [8] (*Bayesian Belief Networks*) representan la distribución de la probabilidad conjunta (*Joint Probability Distribution*) para hacer la predicción de la clase de cada elemento. Este tipo de redes son grafos dirigidos sin ciclos, en donde los vértices son nodos que representan cada uno a un atributo y las aristas dirigidas indican la dependencia entre los atributos, de manera que un nodo que no es descendiente de otro se considera independiente del mismo.

Además de la representación gráfica, hay una tabla de distribución de probabilidad para cada atributo, lo que implica que está relacionada al nodo de ese atributo en la red. Por ejemplo una base de datos de una aseguradora de automóviles, en la que se tienen como atributos: la visibilidad al momento de conducir (buena, mala); el estado del suelo (firme, arenoso, mojado); el

estado del conductor (somnoliento, distraído, atento). Y como clase si ocurre un accidente o no. La red para esta base de datos se muestra en la figura 2.1.

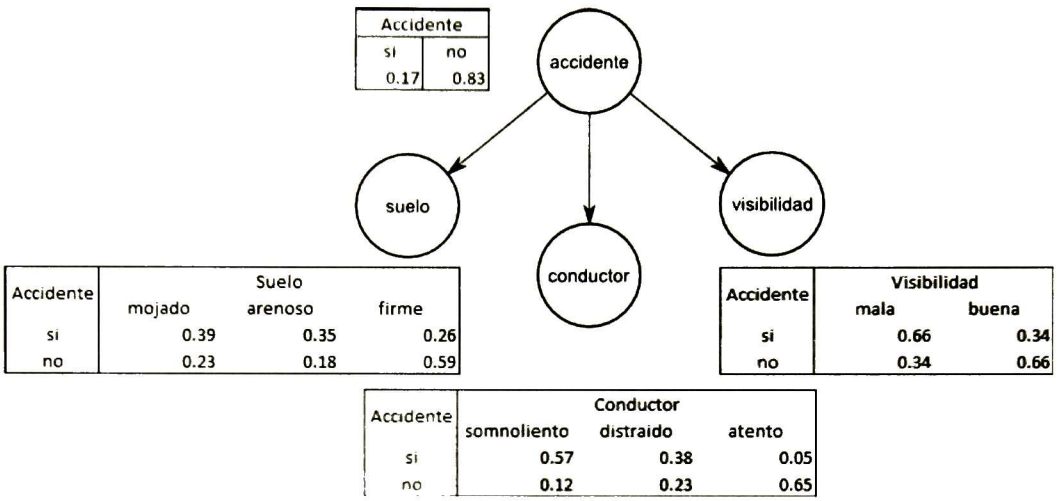

Figura 2.1: Red Bayesiana de compañia aseguradora.

Del lado izquierdo de cada tabla hay una entrada por cada arista que apunta al nodo, y contiene los valores posibles para los nodos que apuntan a él, mientras que del lado derecho aparecen las probabilidades de que ese atributo tome un valor u otro. En otras palabras cada línea del lado derecho describe la distribución de probabilidad para los valores del atributo asociado al nodo. La suma de estos valores es siempre 1.

Para hacer la predicción de la clase, lo que se hace es multiplicar la probabilidad asociada al valor de cada atributo, de esta forma la probabilidad de que ocurra un accidente cuando el conductor esta atento pero la visibilidad es mala y el suelo esta mojado es de 0.0129, mientras que la probabilidad de que no ocurra es de 0.0508, pero estas probabilidades condicionales deben ser normalizadas. Se dividen por la suma de ambas para obtener 0.2025 como la probabilidad de que ocurra un accidente y 0.7975 como probabilidad de evitarlo.

Cuando hablamos de un patrón de paridad cada atributo depende de los demás y la probabilidad de tomar cualquier valor es siempre la misma, lo que indica que es estadísticamente neutral. El problema con los patrones de paridad es que no es posible tomar una decisión ya que las probabilidades para cualquier opción son iguales (figura 2.2).

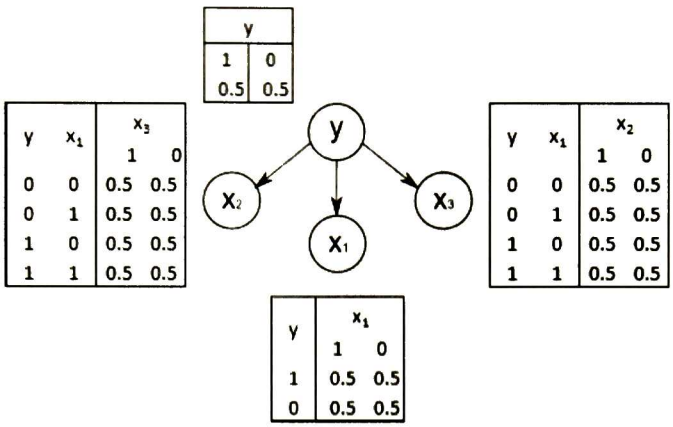

Figura 2.2: Red Bayesiana para patrón de paridad de tres características.

2.3. Algoritmo C4.5.

El algoritmo C4.5 desarrollado por Quinlan [18] es una extensión a su algoritmo anterior ID3 [23] para generar árboles de decisión. Y lo que hace es visitar cada nodo del árbol seleccionando la partición óptima, hasta no poder hacer más particiones. En los árboles de decisión creados por este algoritmo cada nodo interno representa un atributo y las ramas el valor que toma este. Así las hojas del árbol brindan las clases posibles para un objeto.

Para construir el árbol primero se elige el mejor atributo *t* para formar la raíz, después se separan los datos en subconjuntos $\{Ss_1, Ss_2, .., Ss_n\}$ que contengan el mismo valor para *t*; mismos que serán los nodos hijos de la raíz. A continuación se repite el procedimiento para cada nuevo nodo hasta que todos los nodos tengan la misma clase o existan muy pocas muestras. En la figura 2.3 se presenta un ejemplo de este procedimiento, paso a paso.

Para seleccionar la mejor partición el algoritmo utiliza la Ganancia de Información (*Information Gain*), también nombrada Reducción de Entropía. La entropía [4] es el grado de incertidumbre inherente a un conjunto de datos y está definida como:

$$H(X) = -\sum_j p_j log_2(p_j)$$

Donde X es una variable aleatoria, y p_j la proporción de los elementos pertenecientes a una clase en relación al conjunto completo. Utilizando esta medida obtenemos la Ganancia de Información en un conjunto dado. Definido como sigue:

$$IG(A) = H(S) - \sum_k \left(\frac{S_k}{S}\right) H(S_k)$$

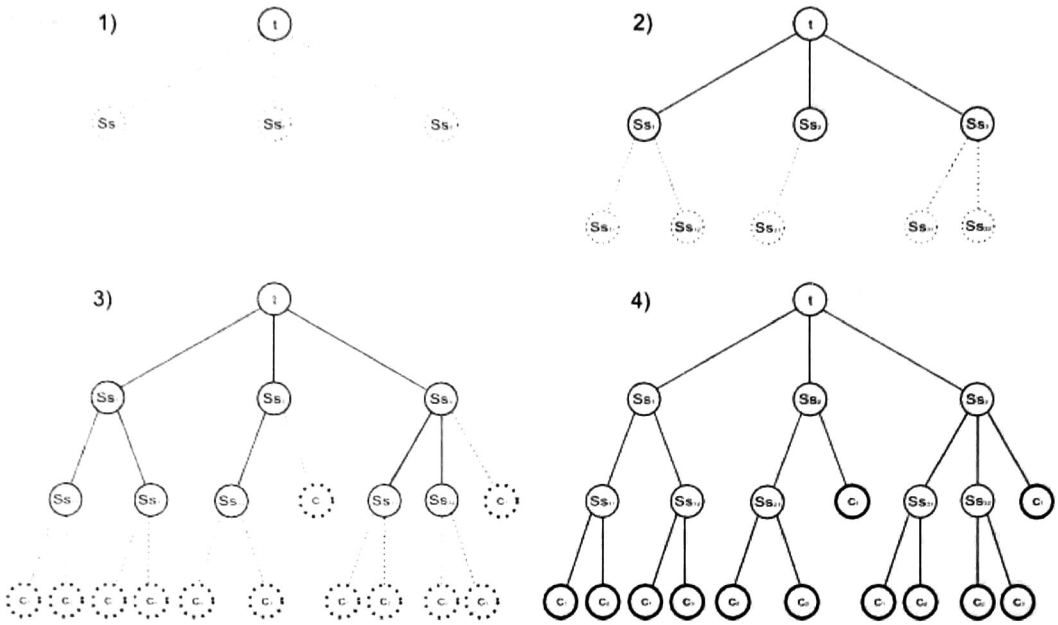

Figura 2.3: Desarrollo de un árbol de decisión usando el algoritmo C4.5.

En la que H(S) es la entropía del conjunto completo y H(S_k) la entropía de un subconjunto k después de particionar S para los valores del atributo A. Se calcula este valor para todas las particiones posibles y se elige la partición que brinde la mayor ganancia. De esta manera se continúa hasta hacer todas las particiones posibles; esto nos lleva a un problema llamado overfitting [12], que indica que el ruido en el conjunto de datos de entrenamiento es capturado como parte del concepto a aprender. La manera de lidiar con el problema es por medio del *podado* (pruning) del árbol.

Existen principalmente dos tipos de podado [13], aquel en el que se detiene el crecimiento del árbol antes de crear todas las ramificaciones posibles llamado *preprunning*, y el que elimina ramificaciones después de haber construido el árbol completo; *postpruning*. En el primer tipo se ahorra trabajo evitando construir partes que después serán desechadas, pero esto evita en ocasiones ver la relación entre atributos, dado que uno solo no puede brindar información valiosa pero combinado con otro pueden ser determinantes para encontrar la clase del objeto a analizar.

El algoritmo C4.5 utiliza *postpruning* en el cual a su vez encontramos dos principales métodos, reemplazo de subárbol (*subtree replacement*) y elevación de subárbol (*subtree raising*), en el primero se inicia el análisis a partir de las hojas y sube hacía la raíz, revisando en cada nodo si puede ser reemplazado por alguna de sus hojas, en caso de que la información del atributo no sea determinante para predecir la clase del objeto; para el segundo que es el utilizado en el algoritmo en cuestión, un subárbol toma el lugar del nodo padre. Esta operación es más compleja que la primera y requiere reclasificar los nodos hijo o las hojas de la ramificación que es remplazada. Se

puede ver un ejemplo de esta operación tomando el árbol completo de la figura 2.3 en la siguiente imagen.

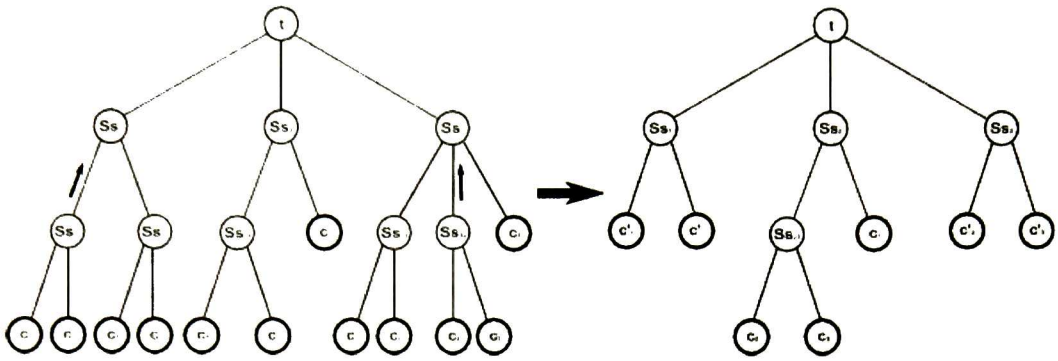

Figura 2.4: Podado de un árbol de decisión por elevación de subárbol.

Se puede notar que las clases de las hojas en los nodos elevados están primadas, lo que indica que no son las mismas que las que tenía el nodo originalmente. Este efecto se debe a la reevaluación necesaria del subárbol que será eliminado y todos sus descendientes para elevar a su nodo sucesor.

Un problema que también se encuentra en los árboles de decisión es cuando faltan valores de atributos. La manera en que se trabaja con valores faltantes cuando se considera que es información relevante para la clasificación, es considerando como un valor posible el valor faltante y construyendo el árbol tomando en cuenta su existencia. Cuando se forma una ramificación considerando un valor faltante se le da un peso, ese peso baja a las demás ramificaciones que parten de ahí hasta las hojas. Por lo que cuando se decide una clase se tiene un grado de certeza correspondiente.

Cuando se utiliza el algoritmo C4.5 en bases de datos con mapeos de paridad es necesario construir el árbol completo y no puede ser podado, ya que de hacerlo se perdería información valiosa para la clasificación. Este desemboca en el peor desempeño del algoritmo en todos los casos. Pero si además faltan valores de los atributos la certeza de clasificación disminuye considerablemente. En las pruebas realizadas por Christopher Thornton [6] en una base de datos con cuatro características; un universo de dieciséis posibles datos. Al eliminar un solo elemento del conjunto de entrenamiento, imposibilito al algoritmo para generalizar el concepto a partir de la información disponible.

Este en general es el problema que presentan los patrones de paridad cuando se pretende clasificar utilizando el algoritmo C4.5.

2.4. Algoritmo Backpropagation.

El algoritmo backpropagation utiliza Redes Neuronales Artificiales, que fueron desarrolladas siguiendo el modelo de funcionamiento de las redes de neuronas en la naturaleza (cerebros animales). Cada neurona tiene un funcionamiento simple pero conectada a otras neuronas su potencia se incrementa, al grado de que en conjunto pueden realizar reconocimiento de patrones como identificar a una persona por el timbre de su voz o por las expresiones que usa comúnmente, e incluso vista a grandes distancias por la manera de caminar.

Las entradas son recolectadas por las neuronas inferiores y procesadas por una función hacía las demás neuronas para después obtener una respuesta de salida. Esto hace de las redes neuronales artificiales un método mucho más robusto que los anteriores y por consiguiente menos susceptible fallas por ruido en la información. La contraparte es que su funcionamiento es menos transparente y requieren mayor tiempo de entrenamiento que otros métodos.

Una desventaja en estas redes es que todos los valores de entrada deben ser números reales entre 0 y 1, por lo que en el caso de valores categóricos es necesario asignarles un valor numérico para trabajar con ellos. Estas redes constan de dos o más capas y los elementos de cada capa se encuentran desconectados entre sí y conectados a todos los nodos de la siguiente capa; aquí la información fluye en un solo sentido. Para la mayoría de los problemas se utilizan tres capas: una de entrada, una de salida y una intermedia llamada capa oculta. La cantidad de nodos en la capa de entrada depende del número de características de los datos. Mientras que la cantidad de capas intermedias y el número de nodos en las mimas puede variar a conveniencia del usuario. Considerando que solo se cuenta con una capa, el número de nodos en ella se incrementa si se quiere tener mayor poder y flexibilidad en la red, sin embargo como en el caso de los árboles de decisión esto puede provocar *overfitting*; dado el caso la cantidad de nodos se debe reducir. Cuando ocurre lo contrario, si la precisión de las predicciones es baja entonces se incrementa la cantidad de nodos en la capa oculta.

Tomando como ejemplo la red de la figura 2.5 el valor de cada uno de los nodos es la suma del producto del valor cada nodo y el arco que lo conecta. Este valor es denominado *net*.

$$net_j = \sum_i W_{ij}x_{ij} = W_{0j}x_{0j} + W_{1j}x_{1j} + \cdots + W_{kj}x_{kj}$$

Donde W es el valor del peso del i-ésimo nodo de entrada al nodo actual y x el valor de ese nodo. En el caso de W_0x_0 es una entrada constante que convencionalmente toma el valor de 1. El valor net_j es usado como entrada para la función de activación, de manera que si supera determinado umbral la neurona artificial dispara.

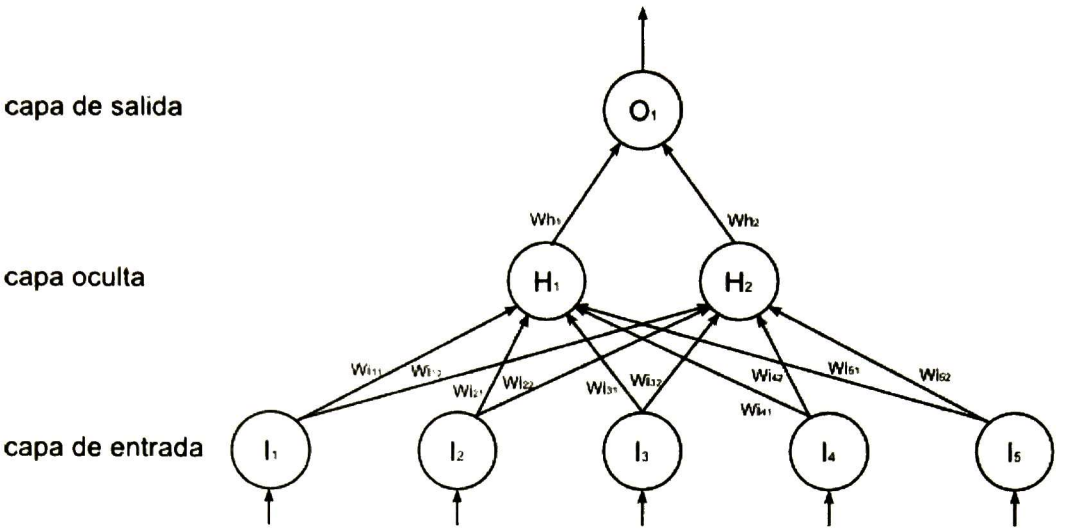

Figura 2.5: Red neuronal artificial de tres capas, con cinco entradas y una salida.

Al entrenar una red neuronal artificial se le da como entrada el conjunto de datos de entrenamiento, a continuación se procesan por la red los elementos; se compara el resultado obtenido para cada uno y la verdadera clase del objeto para obtener el error en la predicción; el error de la red es la suma de los errores de todas las predicciones realizadas. Comúnmente se utiliza la suma de los cuadrados de los errores. A continuación se reevalúan los pesos en los arcos para reducir el error de clasificación, para hacerlo se utiliza el método del gradiente descendente, en donde se estima el valor óptimo de los pesos en los arcos por medio de una derivada parcial del error con respecto a cada peso en los arcos, para obtener la clase buscada y después se actualizan los valores en los pesos.

En resumen, el algoritmo evalúa los elementos de la base de datos de entrenamiento, calcula el error en la clasificación y regresa un porcentaje de la responsabilidad del error a cada uno de los nodos, ahí se actualizan los pesos de los arcos y se vuelve realizar la clasificación. Como se puede notar no es fácil ver lo que ocurre dentro de la red y la complejidad para apreciarlo aumenta con la cantidad de capas ocultas y los nodos contenidos en las mismas. Esto hace difícil determinar los problemas de funcionamiento cuando se trabaja con una base de datos que presenta un patrón de paridad y cuando el patrón se encuentra completo se pueden clasificar los objetos sin problemas, pero cuando existe ruido o ausencia de datos se pierde la precisión de las predicciones.

2.5. Teoría de Patrones.

Propuesta por Ulf Grenander [26], describe el conocimiento del mundo como patrones simples que a su vez forman patrones más complejos, por medio de un lenguaje formal que llama álgebra de patrones; utiliza a su vez estadística, entropía de la información y la topología de los patrones para encontrar variables ocultas en el mundo real. Propiamente la teoría de patrones no se ha utilizado hasta el momento para hacer reconocimiento de patrones, en su lugar es enfoca principalmente en analizar y describir la estructura de los patrones encontrados en la naturaleza.

Si bien la teoría de patrones no forma parte del estado del arte de reconocimiento de patrones, se presenta aquí para mostrar el punto de desarrollo en que se encuentra actualmente. En esta teoría se consideran dos tipos de patrones principalmente, los patrones abiertos y los cerrados, que se subdividen en otras tantas categorías como patrones ornamentales, de movimiento, en plantas, celulares, de lenguaje, en tejidos, y algunos más.

Para representar los patrones se utilizan los llamados generadores que son elementos que se unen a otros para formar una configuración, que indica una regularidad en la información. Las diferentes configuraciones posibles forman un espacio de estructuras regulares denominado imagen, las imágenes representan la formalización de las observaciones. De forma que las configuraciones son las descripciones matemáticas que no se pueden ver directamente pero que se pueden apreciar en las imágenes. Como en el caso de un patrón acústico, en una gráfica podemos ver el ritmo que existe en una canción (la imagen) pero en la configuración es en donde vemos su descripción formal.

En los elementos de una imagen se establecen clases de equivalencia que determinan el patrón subyacente, a esta relación se le llama regla de identificación. Con estas clases de equivalencia se puede reproducir la imagen observada y encontrar otras que si bien no están presentes en el momento también pueden ser formadas por los elementos de dicha imagen. Así es como se da paso a los patrones, el siguiente escalón en la descripción. Un patrón es un conjunto de imágenes, como una imagen un conjunto de configuraciones.

Los patrones tienen una probabilidad de ocurrencia, dada en una distribución de probabilidad, al igual que los generadores en las configuraciones que dan paso al patrón. Pero en los patrones y su apreciación existe ruido, ya hablemos de un sensor como una cámara de video o fotográfica o el ojo humano, existen factores de restan precisión a la observación, en teoría de patrones es llamado deformación, y permite extraer el patrón de la imagen deformada. Por ejemplo una hoja bajo el agua en un arrollo no se ve de la misma manera que en el exterior, la refracción de la luz modifica la imagen, pero se puede reconocer que es una hoja e incluso el tipo de árbol al que pertenece por que el patrón permanece ahí; es lo que reconoce el cerebro al procesar la imagen de la hoja.

Una vez que se tiene la representación de un patrón en estructuras regulares se puede utilizar con diversos fines. Ya sea restauración de una imagen distorsionada o de baja calidad, reconocimiento de patrones para identificación de objetos, extrapolación para realizar predicciones, segmentación para identificar dentro de una imagen objetos de interés y comprensión del patrón propiamente.

Entre los últimos trabajos realizados con teoría de patrones se encuentra el desarrollado por Ulf Grenander, Anuj Srivastava y Sanjay Saini [28], donde presentan un marco matemático para analizar el crecimiento de objetos biológicos. Tales como órganos, células y tumores. Este trabajo propone un modelo de Crecimiento por Difeomorfismo Iterado Aleatorio, en el que el crecimiento total es expresado como una serie de deformaciones locales más pequeñas. El análisis se concentra en una región y se consideran las regiones circundantes que también representan un crecimiento o un deterioro, y la áreas ajenas a la región que se analiza en el momento son empujadas hacía fuera o hacia dentro. Este método provee una mejor descripción de los organismos biológicos que los usados hasta el momento en los que la representación se efectúa por medio de figuras geométricas mas simples (ej. ovoidales).

Las principales áreas en las que se han presentado trabajos utilizando teoría de patrones son biología, matemáticas y visión artificial. Pero es una herramienta relativamente nueva, considerando que las primeras publicaciones se presentaron a partir de 1993.

2.6. Conclusiones.

No existe un método de aprendizaje automatizado que se pueda considerar como el mejor, pues aunque existe algunos mas robustos que otros, o más tolerantes al ruido en la información; bastante común en las bases de datos del mundo real. En ocasiones se puede intentar clasificar un conjunto de datos con un método muy complejo cuando uno mas sencillo es suficiente y ahorra recursos computacionales.

Lo anterior se considera un problema para el que no existe una solución definitiva, ya que también se tiene que considerar la manera en la que se adquiere la información, el como se modela para ser procesada, la forma en que se interpretará la respuesta brindada por el clasificador, entre otros. Si la teoría de patrones se enfoca en describirlos, la pregunta que salta a la mente es: ¿Por qué no se ha utilizado hasta el momento la teoría de patrones como herramienta de clasificación?

Capítulo 3
Propuesta

En este capítulo se presenta el modelo que describe el patrón de paridad que buscamos en las bases de datos a clasificar, mostrando primero el modelo general de la paridad usando Teoría de Patrones, para seguir con un refinamiento del mismo.

3.1. Introducción.

Los clasificadores utilizados en aprendizaje se pueden dividir en dos grupos en base a la manera en la que encuentran relaciones en los datos para aprender. Están los que explotan la estadística de la información y los que trabajan sobre la estructura de los datos. Nuestro modelo se encuentra en el segundo grupo puesto que los mapeos de paridad son estadísticamente neutrales. En otras palabras la probabilidad de que el objeto pertenezca a una clase entre las posibles siempre es la misma.

Volviendo al caso mencionado en el primer capítulo; para elementos de dos características y dos clases posibles, con un patrón de paridad (el comportamiento del XOR lógico); si desconocemos una característica la probabilidad de que la clase sea del primer tipo o del segundo es la misma 0.5, mientras que si desconocemos ambas características seguimos teniendo la misma probabilidad de ocurrencia en la clase para los posibles elementos del dominio.

Con este modelo recrearemos el patrón de paridad para determinar la existencia o ausencia del mismo, ensamblando generadores para coincidir con el patrón. Así podemos determinar si el patrón está completo o si solo una fracción de este está presente.

Para describir un patrón utilizando el álgebra de patrones provista por la Teoría de Patrones inicialmente se definen los generadores (figura 3.1), que son las unidades en dicha álgebra y servirán para encontrar los puntos de unión entre los elementos de la base de datos; esto es, si la aparición de un nuevo elemento concuerda con el patrón en base a los elementos existentes se conecta a alguno de ellos y se continúa uniendo generadores hasta completar el patrón. Cada generador tiene una estructura de lazos (b_0, b_1, .., b_n) con valores (β_0, β_1, .., β_n) asociados a cada uno de ellos.

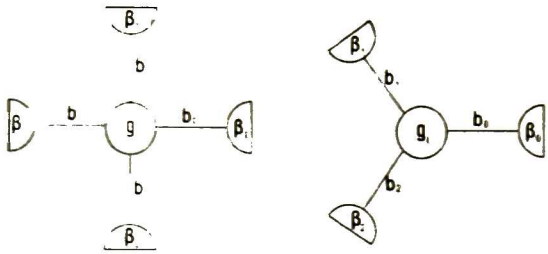

Figura 3.1: Estructura de generadores en Teoría de Patrones.

Las conexiones entre los generadores están dadas por funciones que determinan la dirección y posición posibles para cada tipo de generador, ya que si un lazo con valor x apunta al norte y necesitamos la conexión hacia el este requerimos una transformación de ese generador de $3\pi/2$ rad lo que sería un generador diferente; Entonces la función de "Similitud" Sy la función de "Grupo de estructura de lazos" BSG, nos brindan a partir de una cantidad limitada de generadores una gama mucho mas amplia. Además se requiere una función para determinar cuando una conexión es válida o no, llamada ρ e indica con un 1 la posibilidad de conexión entre dos lazos con valores β_x y β_y. Estas conexiones siguen un grafo seleccionado *a priori*, que indica al patrón buscado. A continuación veremos estas definiciones.

3.2. Un modelo para describir la paridad.

Inicialmente vamos a definir el modelo para un caso particular, para hacerlo sencillo utilizaremos objetos de tres características binarias y dos clases posibles. Por lo que los elementos del dominio serán los mostrados en la tabla 3.1.

Características			Clase	Vector
x_1	x_2	x_3	y	X
0	0	0	0	[(0,0,0),0]
0	0	1	1	[(0,0,1),1]
0	1	0	1	[(0,1,0),1]
0	1	1	0	[(0,1,1),0]
1	0	0	1	[(1,0,0),1]
1	0	1	0	[(1,0,1),0]
1	1	0	0	[(1,1,0),0]
1	1	1	1	[(1,1,1),1]

Tabla 3.1: Universo para patrón de paridad con tres características.

Para el modelo se requieren tantos generadores como elementos existan en el dominio, ya que cada generador representará un elemento del mismo. Para conectar cada generador con sus vecinos utilizaremos tres lazos. El conjunto de generadores (ec. 1) se compone de dos clases de

equivalencia, G^1 (ec. 2) que contiene los generadores positivos asignados a la clase 0 y G^2 (ec. 3) que contiene los generadores negativos relacionados a la clase 1.

$$G = \{G^1 \cup G^2\} \quad [1]$$
$$G^1 = \{g_1, g_4, g_6, g_7\} \quad [2]$$
$$G^2 = \{g_2, g_3, g_5, g_8\} \quad [3]$$

La estructura de los generadores es la misma para todos (ecs. 4 y 5). Solo difieren en el valor de los lazos; los positivos tienen valores 0 mientras que los negativos tienen 1 (ec. 6).

$$B_s(g_i) = \{b_0, b_1, b_2\} \quad [4]$$
$$B_v(g_i) = \{\beta_0, \beta_1, \beta_2\} \quad [5]$$
$$\beta_i = \begin{cases} 0, class = + \\ 1, class = - \end{cases} \quad [6]$$

Al tener la misma estructura podemos utilizar solamente un generador de cada tipo (ec. 7, figura 3.2) y en base a esa partición obtener los demás generadores. Para determinar la posición de los generadores; dada por los valores de las características del objeto al que está ligado el generador; necesitamos la función de *similitud* Sy la función de *grupo de estructura de lazos* BSG para definir la dirección que tomarán en cada posición.

$$G_0 = \{g_1, g_2\} \quad [7]$$
$$\begin{cases} S = \{s_i | i = 1, \dots, 4\} \\ S: (x_1, x_2, x_3) \rightarrow (x_1 + \Delta_1, x_2 + \Delta_2, x_3 + \Delta_3) \\ s_i = (\Delta_1, \Delta_2, \Delta_3) | \Delta_j \in \{0,1\} \end{cases} \quad [8]$$
$$\begin{cases} BSG = \{\mu_i | \mu = 1, \dots, 4\} \\ BSG: (x_1, x_2, x_3) \rightarrow (x_1 \cdot \delta_1, x_2 \cdot \delta_2, x_3 \cdot \delta_3) \\ \mu_i = (\delta_1, \delta_2, \delta_3) | \delta_j \in \{0, \pi\} \end{cases} \quad [9]$$

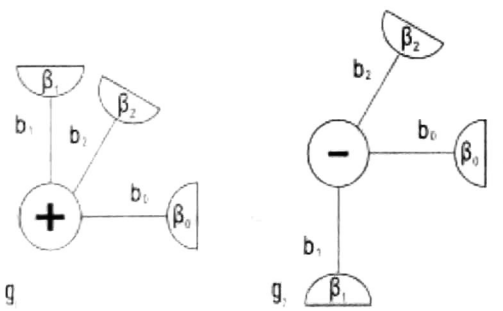

Figura 3.2: Partición inicial para dominios de tres características.

La función de *similitud* es un conjunto de sumas en base dos, que al aplicar la función a un generador cualquiera se obtienen todos los demás de la misma clase, pero con la misma dirección que el original. Para dar la dirección correcta se utiliza en conjunto a la función anterior (figura 3.2)

la función de *grupo de estructura de lazos* que realiza rotaciones en los ejes de cada una de las características, con giros de cero o π radianes. Así, a partir de G_0 construimos el conjunto completo (figura 3.3).

Generador	Características	S	BSG	Resultado
g_1	[0,0,0]	(0,0,0)	(0,0,0)	g_1
g_1	[0,0,0]	(1,1,0)	(π,π,0)	g_7
g_1	[0,0,0]	(1,0,1)	(π,0,π)	g_6
g_1	[0,0,0]	(0,1,1)	(0,π,π)	g_4
g_2	[0,0,1]	(0,0,0)	(0,0,0)	g_2
g_2	[0,0,1]	(1,1,0)	(π,π,0)	g_8
g_2	[0,0,1]	(1,0,1)	(π,0,π)	g_5
g_2	[0,0,1]	(0,1,1)	(0,π,π)	g_3

Tabla 3.2: Creación del conjunto de generadores a partir de la partición inicial.

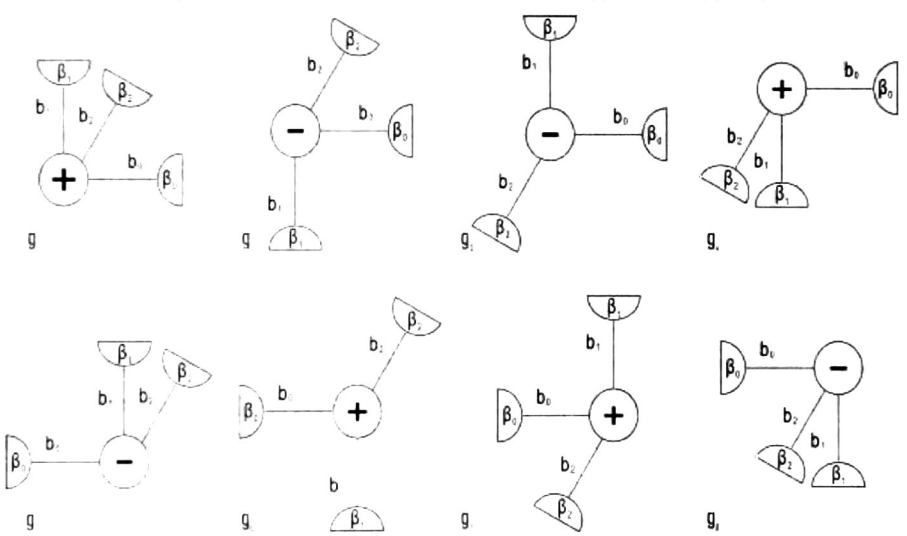

Figura 3.3: Conjunto de generadores para dominios de tres características.

Ahora solo falta indicar cuando dos lazos pueden conectarse y el grafo que indica el patrón a buscar. La función de conexión ρ define cuando dos valores de lazo permiten la conexión. Si las conexiones anteriores forman el grafo dado, se dice que el patrón existe. El grafo σ utilizado en este modelo es un cubo. En la figura 3.4 se muestra la gráfica de los puntos de nuestra base de datos (a), a la que aplicando el patrón del modelo e indicando al grafo del patrón (b) confirmamos si la conexión de los generadores recrea el grafo (c).

$$\rho: \beta \times \beta \rightarrow \{True, False\}$$

β	0	1
0	F	T
1	T	F

[10]

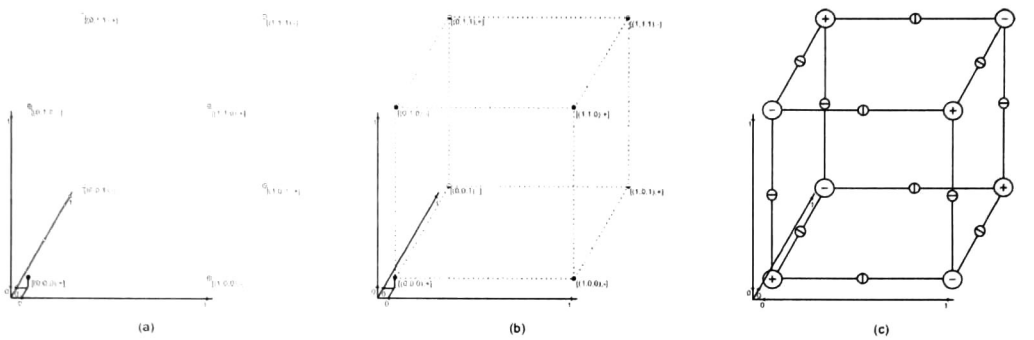

Figura 3.4: Recreación del patrón de paridad sobre la gráfica de la base de datos.

Aún cuando el patrón no se encuentre completo podemos determinar el porcentaje encontrado utilizando este método. Habiendo visto los principios de funcionamiento del modelo pasamos a la siguiente sección, donde se hará una generalización de este modelo para elementos con n cantidad de características.

3.3. Generalización del modelo de paridad.

Como se mencionó en el apartado anterior, en este modelo vamos a representar cada objeto del conjunto de datos con un generador. En la tabla 3.3 vemos la cantidad de generadores que necesitamos para elementos con determinado número de características.

Características	Elementos	Generadores	Lazos
1	1	1	1
2	4	4	2
3	8	8	3
...	...	...	...
n	2^n	2^n	n

Tabla 3.3: Relación entre el número de características y la cantidad de objetos en el universo.

La cantidad de lazos necesaria para conectar un generador con sus vecinos es igual a la cantidad de características, sin estar los valores de estas ligados en forma alguna al valor de los lazos. Así tendremos lazos de dos tipos; uno para cada clase, que llamaremos positivo y negativo, la siguiente figura muestra la estructura que tendrán los generadores del modelo. Mientras que las ecuaciones 11 a 13 indican los generadores de los que dispondremos para una cantidad de entradas determinada en el conjunto de generadores G.

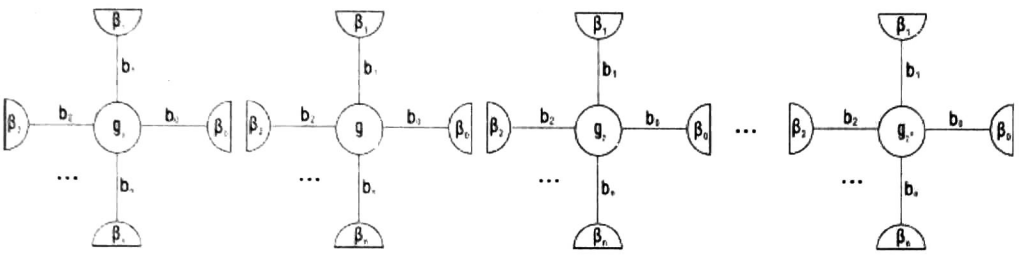

Figura 3.5: Estructura de generadores para patrón de paridad generalizado.

$$G = \{G^1 \cup G^2\} \quad [11]$$

$$G^1 = \{g_1, g_2, \dots, g_{2^{n-1}}\} \quad [12]$$

$$G^2 = \{g_{2^{n-1}+1}, g_{2^{n-1}+2}, \dots, g_{2^n}\} \quad [13]$$

Puesto que la mitad de los generadores serán de la clase positiva y la mitad de la clase negativa, el conjunto de generadores puede ser dividido en dos. G^1 contiene los generadores positivos y G^2 los generadores negativos. La estructura de los generadores, cantidad de lazos (ec. 14) y los valores para estos (ec. 15) son iguales entre todos los positivos e igual para los negativos, solo difieren los valores de los lazos entre clases como se puede notar en el modelo para 3 características, pero usaremos en este caso para los generadores positivos 1, mientras que los valores para los generadores negativos serán 2 (ec. 16).

$$B_s(g) = \{b_1, b_2, \dots, b_n\} \quad [14]$$

$$B_v(g) = \{\beta_1, \beta_2, \dots, \beta_n\} \quad [15]$$

$$\beta_i = \begin{cases} 1, class = + \\ 2, class = - \end{cases} \quad [16]$$

Por cuestiones de eficiencia no trabajaremos con el conjunto completo de generadores. Al tener la misma estructura para todos los generadores positivos y negativos, podemos encontrar todos los generadores a partir de dos, uno de cada clase. El partición inicial G_0 es igual que la del modelo de tres características (ec. 7) y las funciones de *similitud* (ecs. 8,9) y *grupo de estructura de lazos* (ecs. 10,11).

$$\begin{cases} S = \{s_i | i = 1, \dots, 2^{n-1}\} \\ S: (x_1, \dots, x_n) \rightarrow (x_1 + \Delta_1, \dots, x_n + \Delta_n) \\ s_i = (\Delta_1, \dots, \Delta_n) | \Delta_j \in \{0,1\} \end{cases} \quad [17]$$

$$\begin{cases} BSG = \{\mu_i | \mu = 1, \dots, 2^{n-1}\} \\ BSG: (x_1, \dots, x_n) \rightarrow (x_1 \cdot \delta_1, \dots, x_n \cdot \delta_n) \\ \mu_i = (\delta_1, \dots, \delta_n) | \delta_j \in \{0, \pi\} \end{cases} \quad [18]$$

También la función de conexión de lazos es igual al modelo anterior (ec. 10). Así mismo el grafo es un hipercubo de n dimensiones (figura 3.6).

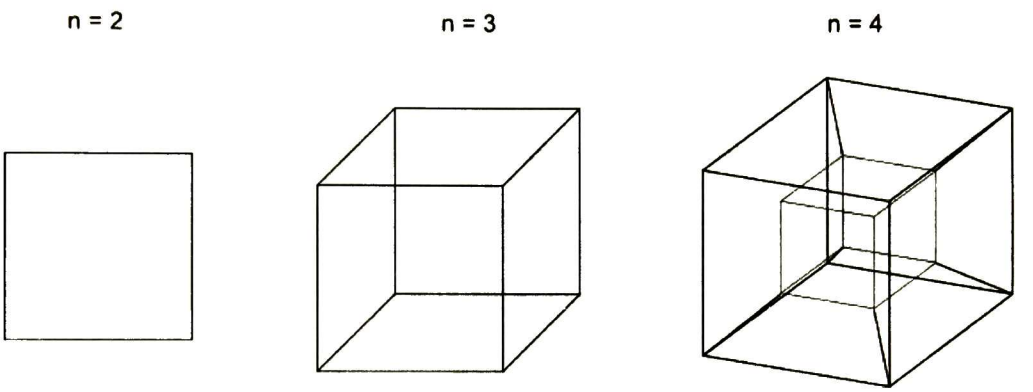

Figura 3.6: Grafos de 2, 3 y 4 dimensiones para el modelo de paridad generalizado.

El problema con este modelo es que cuando la cantidad de características es muy grande en número de lazos de incrementa y la cantidad de conexiones necesarias crece de manera exponencial. Por lo que es necesaria la refinación del modelo.

3.4. Refinamiento del modelo de paridad generalizado.

Para hacer el modelo mas eficiente es necesario reducir la cantidad de lazos para cada generador. Tomando en cuenta que el grafo de conexión es un hipercubo, sabemos que existe un camino Hamiltoniano; a cada vértice llega solo una arista y solo una arista sale de él. Las clases de equivalencia se definen como sigue

$$G^1 = \{g_i | i = 2a, a \in \mathbb{N}, a \leq 2^{n-1}\} \quad [19]$$

$$G^2 = \{g_j | j = 2b + 1, b \in \mathbb{N}, b \leq 2^{n-1}\} \quad [20]$$

Así mismo, cada generador tendrá dos lazos únicamente y la estructura como en los casos anteriores es igual para todos (ecs. 19 y 20), la función de conexión ρ es igual que para el modelo de paridad generalizado. En la figura 3.7 se pueden ver los generadores de la partición inicial.

$$B_s(g) = \{b_1, b_2\} \quad [21]$$

$$B_v(g) = \{\beta_1, \beta_2\} \quad [22]$$

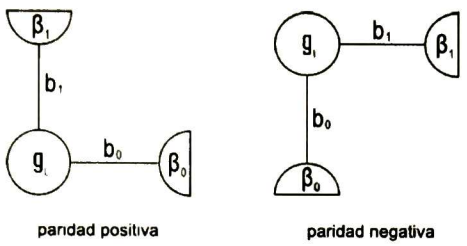

Figura 3.7: Partición inicial para el modelo de paridad generalizado simple.

Las funciones de *similitud* y *grupo de estructura de lazos* son iguales a las descritas anteriormente (ecs. 17 y 18). Y los grafos del camino Hamiltoniano para dos, tres y cuatro características se presentan en la siguiente figura.

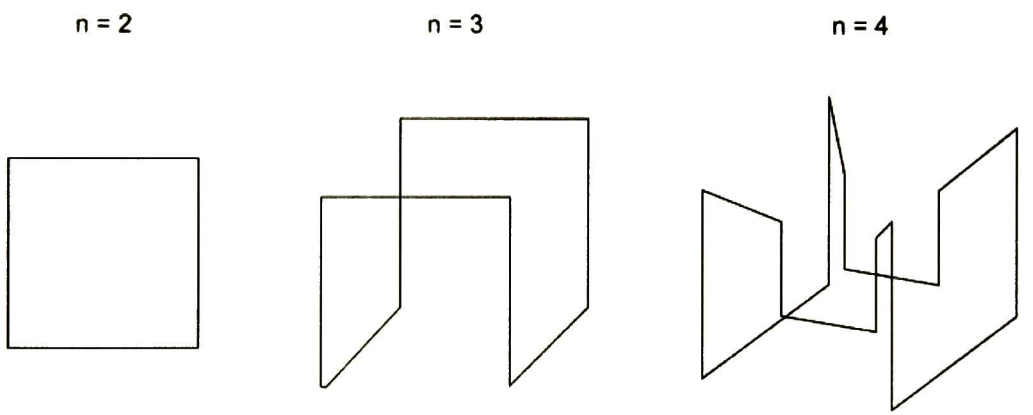

Figura 3.8: Grafos de 2, 3 y 4 dimensiones para el modelo de paridad generalizado simple.

El camino Hamiltoniano se puede encontrar utilizando el código de Gray, que enumera los valores binarios en una secuencia en donde el siguiente número en la serie es un bit diferente al inmediato anterior.

3.5. Conclusiones.

Lo que hace que este método funcione a diferencia de los existentes en Reconocimiento de Patrones cuando existe un mapeo de paridad, es que aunque como otros que trabajan sobre la estructura de los datos, aquí no se necesita el patrón completo. Es posible encontrar fracciones del mapeo ya sea por que no se encuentran todos los datos o por presencia de ruido. Es posible determinar el porcentaje del patrón encontrado, ya que una base de datos podría dado el caso tener un comportamiento de paridad doble.

La paridad doble se refiere a que en determinada región de los datos, aquellos objetos con una cantidad par de características iguales a 1 pertenecen a la clase positiva, pero fuera de esa región los objetos con cantidad par de características iguales a 1 pertenecen a la clase negativa. En este caso el modelo encontraría la región que apoya la paridad positiva, que es la referida en la introducción y la región que contiene el patrón de paridad negativa, que se comporta de forma inversa a la especificada.

Además es importante notar que aunque la cantidad de generadores que componen el modelo crece de manera exponencial al número de características, no es necesario trabajar con todos los generadores, debido al apoyo de las funciones de *similitud* y *grupo de estructura de lazos*. Por que cada vez que fuera necesario un generador se puede obtener en línea a partir de los dos generadores iniciales.

Capítulo 4
Implementación y experimentos

En este capítulo se presenta el desarrollo del algoritmo para utilizar el modelo propuesto, así como los resultados obtenidos en los experimentos realizados y su discusión.

4.1. Introducción.

En el capítulo 3 se presentó un modelo para definir el comportamiento de los datos cuando existe un patrón de paridad, y encontrar ese patrón en partes o completo. Pero es necesario utilizarlo de la forma adecuada para clasificar datos de manera eficiente. Dado que trabajamos con un problema que crece exponencialmente a medida que se incrementa la cantidad de características no podemos darnos el lujo de analizar el conjunto de entrenamiento para cada nuevo dato que se va a clasificar, pues sería sumamente costoso en tiempo de procesamiento.

Incluso sabemos que en muchas bases de datos solo existe paridad en determinadas regiones, así que los elementos que brindan la información más valiosa para realizar la clasificación son aquellos que se encuentran más cerca del dato a clasificar. Por lo tanto trabajaremos con el vecindario del objetivo.

También necesitamos un factor de certidumbre para permitir al algoritmo trabajar en presencia de ruido; común en bases de datos reales. Estos dos parámetros serán configurables por el usuario para definirlos según sea el caso.

4.2. Algoritmo.

Para utilizar el modelo como herramienta de clasificación usaremos la partición inicial y la función de *Similitud* con la que se obtendrán todos los generadores y asumiremos automáticamente la orientación de los generadores. Más adelante se verá la intención de esto.

En la figura 4.1 se muestra el algoritmo propuesto. El cual toma como entrada la base de datos a clasificar, el conjunto de entrenamiento, el tamaño del vecindario y el umbral de decisión que indica que porcentaje del patrón debe encontrarse en el vecindario para decidir si un elemento forma parte de un patrón de paridad o no.

En las dos primeras líneas del algoritmo se definen el conjunto G, que contiene la partición inicial del espacio completo de generadores. Y el conjunto S, con el que se obtienen los generadores necesarios basados en la partición inicial conforme se requiera.

Algorithm for Classification via Pattern Theory
input: training dataset T, unknown dataset D, threshold τ, quantity of neighbors k
output: D'
PatternTheoryClassifier(T, D, τ, k)
1: *let G be the set of generators from the parity pattern for the number of features in D*
2: *let S be the set of similarities corresponding to G*
3: *foreach $X \in D$*
4: *let $N = \{\eta_1, \dots, \eta_k\}$ be the neighborhood of X, where $N \subseteq T$*
5: *foreach neighbor η_j*
6: *attach a generator g_l to η_j where $x \in \eta_j = s_l \cdot g_1$*
7: *let f be the number of neighbors that can be connected to η_j*
8: *replace $\eta_j = \{x, y\}$ by $\eta'_j = \{x, y, p\} \mid p = (f+1)/k$*
9: *let h^+ be the set of positive neighbors that can be connected to X*
10: *let h^- be the set of negative neighbors that can be connected to X*
11: *take the bigger h^c*
12: *if $|h^c|/k > \tau$*
13: *create $(x', y', p') \in T' \mid x' = x, y' \leftarrow c, p' = p \in h^c$*
14: *else*
15: *create $(x', y', p') \in T' \mid x' = x, y' \leftarrow i, p' = 0$*
16: *return D'*

Figura 4.1: Algoritmo para clasificación mediante Teoría de Patrones.

A continuación se realiza el análisis de la base de datos, miembro por miembro. Lo primero que se necesita es el vecindario del objetivo (línea 4); a los elementos del vecindario se les etiquetarán los generadores correspondientes a su posición y clase (línea 5). A partir de este punto se trabaja con las etiquetas en lugar de los datos propiamente. Para cada elemento del vecindario se revisan las conexiones posibles (línea 7) con los demás elementos del mismo (figura 4.2), y se agrega a su vector característico la proporción del patrón encontrado para cada uno (línea 8). De manera que si el patrón está completo en el vecindario, todos sus elementos tendrán un valor de certidumbre igual a 1.

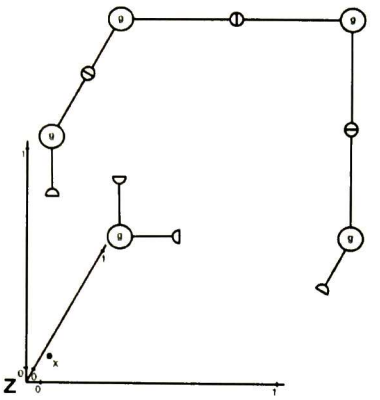

Figura 4.2: Ejemplo de conexión entre generadores del vecindario.

Ya que los generadores pueden no encontrarse juntos en el espacio se hará la conexión o bien directamente (figura 4.3a) o por desdoble (figura 4.3b); en este caso lo que se intenta ver es si al reproducir los generadores necesarios para que el generador *a* alcance al generador *b* es posible realizar la conexión, en otras palabras que no exista un conflicto en la conexión cuando uno apoya la clase positiva y el otro la clase negativa.

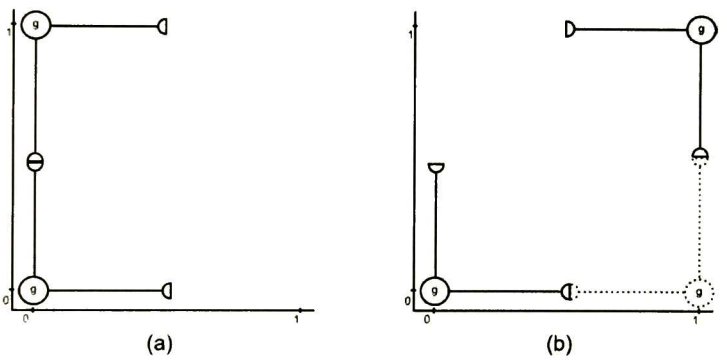

Figura 4.3: Tipos de conexiones entre generadores.

Después se conectarán uno a uno los elementos del vecindario al objetivo X, de la misma forma en la que se llevo a cabo la conexión entre los vecinos. Asumiendo para X la clase necesaria para realizar la conexión (figura 4.4). Entonces se dividen los generadores del vecindario en dos subconjuntos mediante la siguiente regla: si el generador apoya la clase negativa de X (es necesario que X sea negativa para realizar la conexión) toma lugar en h^-, mientras que si apoya la clase positiva (X se considera positiva para realizar a conexión) va a h^+

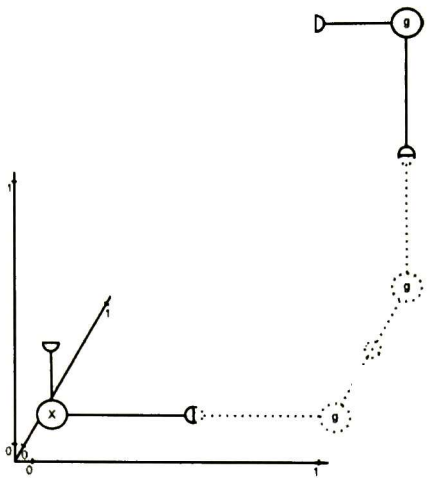

Figura 4.4: Ejemplo de conexión de X hacía un vecino.

En seguida se toma el mayor subconjunto, en caso de que se formaran ambos o el único existente, y se analiza si la proporción de vecinos que apoyan la clase del subconjunto seleccionado es superior al umbral. De ser así, se agrega a X la clase del subconjunto y el valor *p* como factor de certidumbre de cualquiera de los elementos de h^c (la proporción de conexión es la misma en todos los elementos puesto que ellos conforman la conexión). Si no se supera el umbral el objetivo se marca como indeterminado.

4.3. Experimentos en bases de datos artificiales.

Para probar el algoritmo inicialmente se realizaron experimentos con bases de datos artificiales. Es importante notar que se realizaron experimentos en bases de datos con 8, 12, 16 y 20 características binarias, en donde el patrón estaba completo y luego con ausencia de datos. Estos experimentos no se discutirán debido a que siempre se obtuvo la clasificación de todos los objetos con un 100% de certeza y sin errores.

Debido a lo anterior y para demostrar la eficacia del algoritmo se creó una base de datos con objetos de ocho características binarias. A esta base de datos se le metió ruido en diferentes cantidades, desde un 10% hasta un 50% incrementado por decenas, para así tener cinco bases de datos diferentes. En cada base de datos se hicieron pruebas tomando de forma aleatoria el 10%, 20% y 30% de los datos como conjunto de entrenamiento.

En total se realizaron quince experimentos con datos artificiales, cada uno tres veces. La información mostrada en la tabla 4.1 son las medias de las tres corridas de prueba en las cinco bases de datos. Para los experimentos el vecindario fue de diez elementos con un umbral de aceptación de 0.75.

conjunto para clasificación		conjunto de entrenamiento		datos con ruido		datos clasificados		errores de clasificación	
porcentaje	elementos	elementos	porcentaje	elementos	porcentaje	elementos	porcentaje	elementos	porcentaje
230	90%	26	10%	3	10%	228	99%	0	0%
230	90%	26	10%	5	20%	164	71%	0	0%
230	90%	26	10%	8	30%	116	50%	0	0%
230	90%	26	10%	11	40%	31	13%	1	3%
230	90%	26	10%	13	50%	25	11%	24	96%
205	80%	51	20%	5	10%	201	98%	0	0%
205	80%	51	20%	10	20%	135	66%	0	0%
205	80%	51	20%	15	30%	113	55%	0	0%
205	80%	51	20%	20	40%	28	14%	28	100%
205	80%	51	20%	25	50%	14	7%	14	100%
179	70%	77	30%	8	10%	167	93%	0	0%
179	70%	77	30%	15	20%	139	78%	0	0%
179	70%	77	30%	23	30%	74	41%	0	0%
179	70%	77	30%	31	40%	29	16%	29	100%
179	70%	77	30%	38	50%	14	8%	14	100%

Tabla 4.1: Estadísticas de los experimentos en bases de datos artificiales.

Las columnas expresan información por pares, en cada par de columnas la de la izquierda presenta el porcentaje y la de la derecha la cantidad de datos correspondiente a este. La gráfica correspondiente a la tabla anterior se muestra a continuación. En el primer par de columnas se encuentra el conjunto de datos a clasificar, en el siguiente par el conjunto de entrenamiento. El porcentaje de datos con ruido es respecto a la base de datos original (antes de dividirla en datos para clasificar y conjunto de entrenamiento). El punto de referencia de los datos clasificados es el conjunto de datos para clasificar. En el último par de columnas se encuentran los porcentajes y datos clasificados erróneamente.

En la tabla se puede apreciar no solo que al incrementar la cantidad de ruido la eficacia de clasificación disminuye; además, mientras mayor es el porcentaje de datos de entrenamiento con ruido también disminuye la eficacia de clasificación. Esto es por que la cantidad de datos con ruido se incrementa aumentando la probabilidad de encontrar ruido en el vecindario. La figura 4.5 contiene la gráfica correspondiente a la tabla anterior.

En esta figura se muestra el porcentaje de datos a clasificar en el eje vertical contra el porcentaje de ruido en el eje horizontal. Las tres barras corresponden al porcentaje de datos tomados para entrenamiento con 10%, 20% y 30% respectivamente.

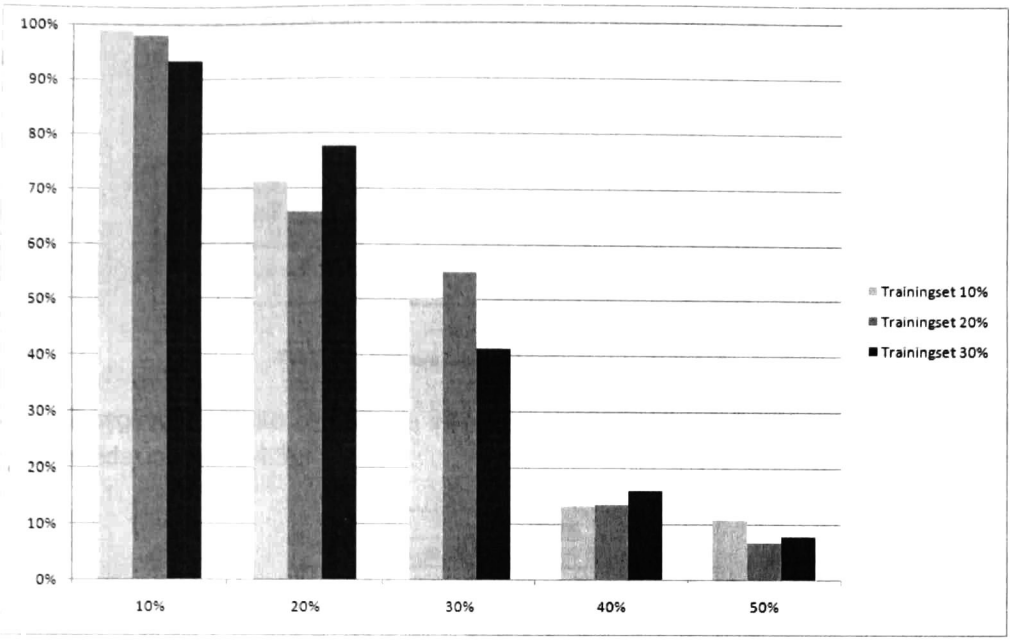

Figura 4.5: Resultados de los experimentos en bases de datos artificiales.

4.4. Experimentos en bases de datos reales.

Para los experimentos con datos reales se busco una base de datos con dos clases, y datos enteros para facilitar su interpretación binaria. La base de datos seleccionada fue "Haberman's Survival Data" del repositorio de la UCI [2]. La base de datos contiene información de un estudio conducido entre los años 1958 y 1970 en el hospital Billings de la Universidad de Chicago a los pacientes vivos que fueron sometidos a cirugía para cáncer de pecho.

La base de datos contiene 306 instancias, con cuatro atributos cada una (incluyendo la clase). De las tres características, la primera es la edad del paciente cuando fue sometido a la cirugía, la segunda el año en que se realizó la operación, para terminar con el número de nodos axilares positivos detectados (ganglio linfático en el área de la axila a la que el cáncer se ha diseminado [17]). La clase puede tener dos valores: 1, el paciente sobrevivió cinco años o mas; 2, el paciente falleció en el transcurso de los cinco años subsecuentes a la cirugía.

Ya que el algoritmo trabaja con datos binarios se hizo una interpretación de los datos, y como todos los atributos son enteros se pudo interpretar la información en cantidades binarias sin mayor complicación. Se realizaron dos modificaciones, la primera interpretación fue una discretización, en donde cada bit representa un rango y toma valor 1 cuando el valor original del atributo cae dentro.

El primer atributo cuyos valores oscilan entre 30 y 83 se dividió en 11 rangos de 5 unidades (tabla 4.2).

		rango										
		30-34	35-39	40-44	45-49	50-54	55-59	60-64	65-69	70-74	75-79	80-84
atributo 1	30	1	0	0	0	0	0	0	0	0	0	0
	35	0	1	0	0	0	0	0	0	0	0	0
	40	0	0	1	0	0	0	0	0	0	0	0
	...	...	...	...	...	...	...	...	...	...	...	...
	84	1	0	0	0	0	0	0	0	0	0	1

Tabla 4.2: Primera interpretación para el atributo 1.

El segundo atributo comprende valores entre 58 y 69; se dividió en 6 rangos de dos unidades cada uno (tabla 4.3).

		rango					
		58-59	60-61	62-63	64-65	66-67	68-69
atributo 2	58	1	0	0	0	0	0
	60	0	1	0	0	0	0
	62	0	0	1	0	0	0
	...	...	...	...	...	...	...
	68	0	0	0	0	0	1

Tabla 4.3: Primera interpretación para el atributo 2.

El último atributo se dividió en 11 rangos, en grupos de 5 unidades, con valores entre 0 y 54 (tabla 4.4).

		rango										
		0-4	5-9	10-14	15-19	20-24	25-29	30-34	35-39	40-44	45-49	50-54
atributo 3	0	1	0	0	0	0	0	0	0	0	0	0
	5	0	1	0	0	0	0	0	0	0	0	0
	10	0	0	1	0	0	0	0	0	0	0	0
	...	...	...	...	...	...	...	...	...	...	...	...
	52	0	0	0	0	0	0	0	0	0	0	1

Tabla 4.4: Primera interpretación para el atributo 3.

Con esta interpretación obtenemos vectores característicos de 29 atributos binarios incluyendo la clase. Una vez interpretada la información se puede procesar con el algoritmo brindando los resultados mostrados en la tabla 4.5.

conjunto para clasificación		conjunto de entrenamiento		datos clasificados		aciertos		errores	
elementos	porcentaje	elementos	porcentaje	elementos	porcentaje	elementos	porcentaje	elementos	porcentaje
275	90%	31	10%	182	66%	137	50%	45	16%
245	80%	61	20%	83	34%	66	27%	17	7%
214	70%	92	30%	108	50%	85	40%	23	11%
184	60%	122	40%	98	53%	81	44%	17	9%

Tabla 4.5: Estadísticas de los experimentos en bases de datos reales con la primera interpretación.

En la tabla anterior se debe notar que conforme aumenta el número de elementos en el conjunto de entrenamiento peor es el desempeño del algoritmo en cuanto a clasificación se refiere. Esto se debe a que, al igual que en las bases de datos artificiales, la probabilidad de que el conjunto de entrenamiento contenga datos con ruido se incrementa con su tamaño.

La siguiente figura muestra la gráfica correspondiente a la tabla. El eje vertical representa el porcentaje de datos clasificados, mientras que el eje horizontal corresponde al porcentaje de datos tomados para el conjunto de entrenamiento.

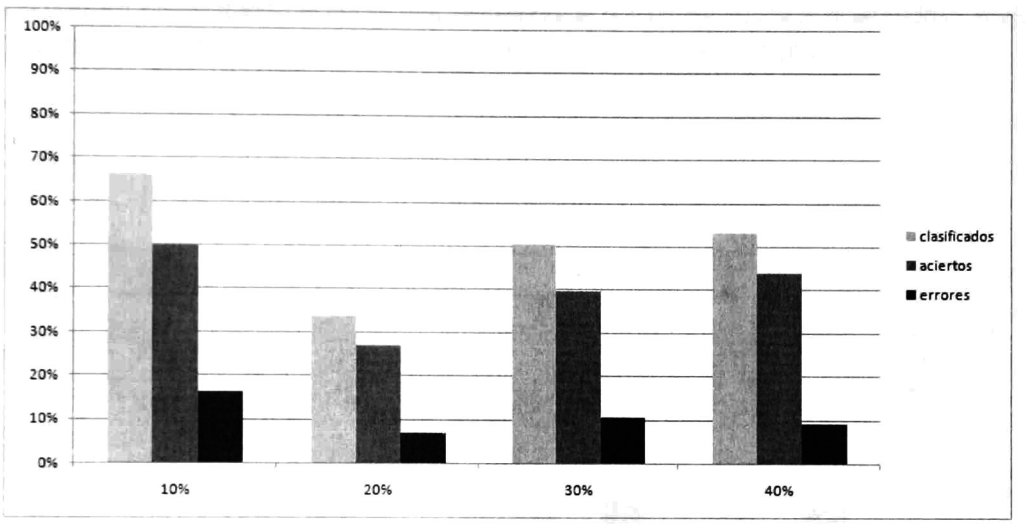

Figura 4.6: Resultados de los experimentos con bases de datos reales; primera interpretación.

La segunda interpretación es una conversión de los valores enteros a binarios (tabla 4.6). Dado que el primer atributo oscila en un rango de 30 a 83 se utilizaron 6 bits. El segundo atributo toma valores entre 58 y 69, para este se utilizaron 4 bits. El último atributo se interpreta en 6 bits al igual que el primero y tiene valores entre 0 y 54.

atributo 1	interpretación	atributo 2	interpretación	atributo 3	interpretación
30	[0,0,0,0,0,0]	58	[0,0,0,0]	0	[0,0,0,0,0,0]
31	[0,0,0,0,0,1]	59	[0,0,0,1]	1	[0,0,0,0,0,1]
32	[0,0,0,0,1,0]	60	[0,0,1,0]	2	[0,0,0,0,1,0]
...	...	...	...	...	...
84	[1,1,0,1,0,1]	69	[1,0,1,1]	54	[1,1,0,1,0,0]

Tabla 4.6: Segunda interpretación para los tres atributos.

En la tabla 4.7 se concentran los resultados de los experimentos para esta interpretación. Al igual que para la primera interpretación se hicieron experimentos con conjuntos de entrenamiento conteniendo el 10%, 20%, 30% y 40% del total de los datos. Los datos de entrenamiento fueron seleccionados de manera aleatoria, como en los casos anteriores.

conjunto para clasificación		conjunto de entrenamiento		datos clasificados		aciertos		errores	
elementos	porcentaje	elementos	porcentaje	elementos	porcentaje	elementos	porcentaje	elementos	porcentaje
275	90%	31	10%	19	7%	8	3%	11	4%
245	80%	61	20%	22	9%	15	6%	7	3%
214	70%	92	30%	37	17%	9	4%	28	13%
184	60%	122	40%	35	19%	14	8%	21	11%

Tabla 4.7: Estadísticas de los experimentos en bases de datos reales con la segunda interpretación.

En la figura 4.7 se puede apreciar la gráfica de resultados. Como se ve el desempeño en este caso es muy bajo y significa que la pérdida de información es muy costosa. Como en la gráfica de la primera interpretación el eje vertical representa el porcentaje de datos clasificados y el eje horizontal el porcentaje de datos contenidos en el conjunto de entrenamiento.

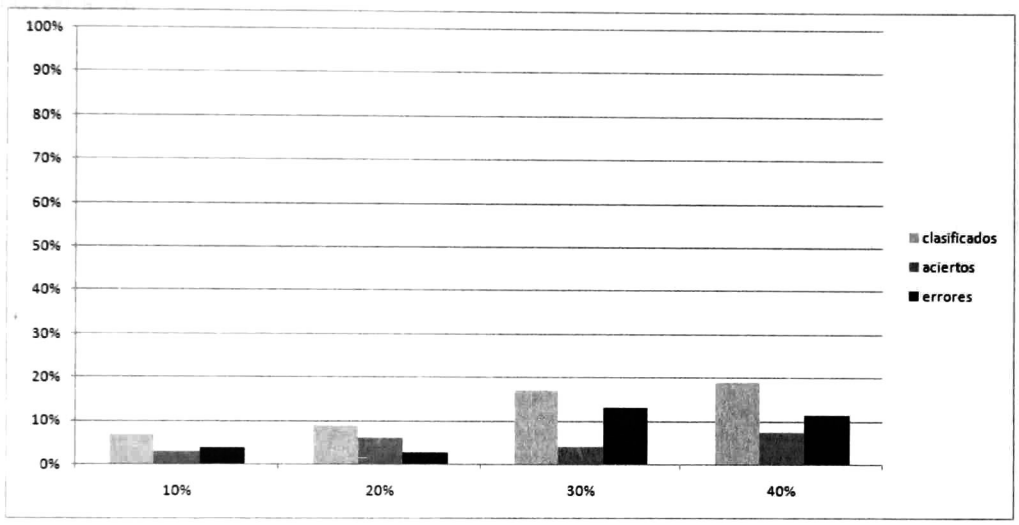

Figura 4.7: Resultados de los experimentos con bases de datos reales, segunda interpretación.

4.5. Discusión de los resultados.

Comparando el desempeño obtenido en los experimentos con datos artificiales y reales se deben puntualizar las siguientes ideas:

El desempeño es por mucho superior en los datos artificiales debido a que en ese caso la información fue concebida originalmente de manera binaria, mientras que en el caso de los datos reales depende en gran medida de la interpretación binaria que se de a los datos. Muestra de esto se ve entre las dos diferentes interpretaciones a las que fue sujeta la base de datos de Hambersman.

Aunque el algoritmo es susceptible al ruido en la información, respecto al mapeo de paridad; se obtuvo una eficacia aceptable hasta en un 30% de datos con ruido. Hablando en datos artificiales en donde tenemos la certeza de que existe el patrón.

La interpretación binaria directa de datos enteros genera perdida de información, al grado de no permitir la clasificación por este método.

Dicho lo anterior podemos asegurar que este algoritmo es un buen mecanismo de apoyo para otros métodos de Reconocimiento de Patrones y puede ser usado en una etapa de post-procesamiento, una vez que los datos ya fueron clasificados se puede mejorar esta clasificación ya sea corrigiendo la decisión original o aumentando el factor de certeza en la clase determinada originalmente.

Capítulo 5
Conclusiones y trabajo futuro

En este capítulo se discuten las conclusiones a las que se llega a partir del trabajo realizado y se exponen puntos en los que se debe trabajar en el futuro para mejorar el método.

5.1. Conclusiones.

El trabajo presentado en los capítulos anteriores demuestra que aún se puede extender el estado del arte de Reconocimiento de Patrones en cuanto a los problemas que es capaz de resolver, o en otras palabras los conceptos que se pueden adquirir con una computadora por estos métodos. Tomando como ejemplo el caso de estudio adoptado en el desarrollo de esta tesis, el problema de paridad no se ha considerado de importancia hasta el momento; en gran medida por la manera en la que se modela la información de las bases de datos disponibles. Sin embargo aunque no es tan común encontrar datos en representación binaria tampoco es imposible, por ejemplo el MMPI-2 (Inventario Multifásico de la Personalidad de Minnesota [10]), que es un instrumento psicológico para encontrar rasgos de personalidad de un individuo e incluye 567 reactivos de respuestas si o no, lo que brinda una base de datos con características binarias.

También se demostró que este clasificador trabaja bien con datos enteros y de igual forma lo puede hacer con datos reales, definiendo rangos de valores interpretados en bits. Si bien no hace una clasificación óptima funge como refuerzo y corrector para otros clasificadores que encuentran problemas cuando existe un mapeo de paridad en los datos.

Además se vio que la dificultad para adquirir un concepto por lo métodos convencionales cuando subyace en el un mapeo de paridad está directamente relacionada a que estos son estadísticamente neutrales. Esta característica no es exclusiva de la información binaria, pudiendo presentarse también en datos enteros, e incluso podemos encontrar bases de datos en las que la neutralidad estadística esta dada en un parte de las características o en alguna región de los datos.

Otro punto importante a notar es el llamado Error de Bayes [30], que indica una región en los datos que no se puede clasificar debido a la ambigüedad existente en las clases de los datos encontrados en dicha región. Si dentro de esta región de error encontramos un Error de Bayes podemos fácilmente clasificar los elementos que se encuentran dentro con este método.

Por esta razón se puede decir que la metodología presentada no está limitada a información binaria, solo es cuestión de cambiar el modelo para procesar tipos de datos diferentes. Pero la intención principal de este trabajo es demostrar como utilizar Teoría de Patrones para clasificación. En conclusión, este trabajo abre la puerta a mejores clasificadores, y amplía el espectro de problemas que se pueden resolver con aprendizaje automatizado.

5.2. Trabajo futuro.

Entre las mejoras y complementos que se pueden realizar a este trabajo encontramos las siguientes mencionadas.

- Obtener una base de datos generada originalmente con información binaria, como la mencionada en la sección anterior del MMPI-2, para realizar experimentos en bases de datos reales sin tener que hacer la interpretación de los datos, conservando así la información relevante por completo.

- Codificar el algoritmo como un módulo para trabajar en conjunto con otros clasificadores y revisar la mejora en el desempeño antes y después de utilizar el algoritmo. Así como hacer una tabla estadística y su gráfica de barras correspondiente como se realizo para los experimentos mostrados.

- Automatizar la interpretación de los datos enteros a binarios y buscar mejores interpretaciones, para asegurar que se conserve de la información relevante al realizar la interpretación de los datos.

- Modificar el modelo para trabajar con información en datos enteros, lo cual impactaría directamente la gama de problemas con los que se puede trabajar ya que no estaría restringido a patrones de paridad sino que trabajaría con patrones de neutralidad estadística.

Bibliografía.

[1] Andrew Webb (2002). *Statistical Pattern Recognition,* Second Edition. Inglaterra: John Wiley and Sons.

[2] Asuncion, A. & Newman, D.J. (2007). UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

[3] Bir Bhanu, Yingqiang Lin, Krzysztof Krawiec (2005). *Evolutionary Synthesis of Pattern Recognition Systems*. USA: Springer Science.

[4] Chang, C.-I.; Du, Y.; Wang, J.; Guo, S.-M.; Thouin, P.D. (2006). *Survey and comparative analysis of entropy and relative entropy thresholding techniques*. En Vision, Image and Signal Processing, Volume 153, Issue 6, (pp. 837-850).

[5] Christopher M. Bishop (2006). *Pattern Recognition and Machine Learning* (pp. 378-383). USA: Springer Science.

[6] Christopher Thornton (2003). *Parity: The Problem that Won't Go Away*. Cognitive and Computing Sciences. University of Sussex. UK.

[7] D. Michie, D. J. Spiegelhalter, C. C. Taylor (1994). *Machine Learning, Neural and Statistical Classification.* USA: Prentice Hall.

[8] Daniel T. Larose (2005). *Discovering Knowledge in Data: An introduction to Data Mining*. USA: Wiley-Interscience.

[9] David J. C. MacKay (2003). *Information Theory, Inference and Learning Algorithms*. Inglaterra: Cambridge.

[10] Emilia Lucio Gómez-Maqueo (2003). *Uso e Interpretación del MMPI-2 en Español*. México: El Manual Moderno.

[11] Graham J. Williams & Simeon J. Smoff (2006). *Data Minning: Theory, Methodology, Techniques, and Applications*. Alemania: Springer-Verlag.

[12] Ian H. Witten & Eibe Frank (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (pp. 227-233), Second Edition. USA: Elsevier.

[13] Johannes Fürnkranz (1994). *Pruning Methods for Rule Learning Algorithms.* Austrian Research Institute for Artificial Intelligence. Austria.

[14] Keinosuke Fukunaga (1990). *Introduction to Statistical Pattern Recognition*, Second Edition. USA: Academic Press.

[15] Luc Devroye, László Györfi, Gábor Lugosi (1996). *A Probabilistic Theory of Pattern Recognition.* USA: Springer-Verlag.

[16] Marcus A. Maloof (2006). *Machine Learning and Data Mining for Computer Security.* USA: Springer-Verlag.

[17] National Cancer Institute (2009). Cancer Dictionary. USA: NCI. [http://www.cancer.gov/Templates/db_alpha.aspx?CdrID=45845&lang=spanish].

[18] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* USA: Morgan Kaufmann Publishers.

[19] Richard O. Duda, Peter E. Hart, David G. Stork (2000). *Pattern Classification* (p. 12), Second Edition. USA: Wiley-Interscience.

[20] Richard O. Duda, Peter E. Hart, David G. Stork (2000). *Pattern Classification* (p. 30), Second Edition. USA: Wiley-Interscience.

[21] S. B. Kotsiantis (2007). *Supervised Machine Learning: A Review of Classification Techniques* (pp. 249-268). USA: Informatica 31.

[22] S. J. Hanson, W. Remmele, R. L. Rivest (1993). *Machine Learning From Theory to Applications: Cooperative Research at Siemens and MIT.* Alemania: Springer-Verlag.

[23] Tom M. Mitchel (1997). *Machine Learning* (pp. 52-81). USA: McGraw Hill.

[24] Trevor Hastie, Robert Tibshirani, Jerome Friedman (2003). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* USA: Springer.

[25] Ulf Grenander (1994). *General Pattern Theory: A Mathematical Study of Regular Structures.* USA: Clarendon Press.

[26] Ulf Grenander (1996). *Elements of Pattern Theory.* Inglaterra: John Hopkins University Press.

[27] Ulf Grenander and Michael I. Miller (2007). *Pattern Theory: From Representation to Inference.* USA: Oxford University Press.

[28] Ulf Grenander, Anuj Srivastava, and Sanjay Saini (2007). *A Pattern-Theoretic Characterization of Biological Growth.* IEEE Transactions on Medical Imaging, Vol. 26, No. 2.

[29] Vladimir N. Vapnik (1998). *Statistical Learning Theory.* USA: Wiley-Interscience.

[30] Vojislav Kecman (2001). *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models.* USA: MIT Press.

[31] Wegener, Ingo and Randall Pruim (2005). *Complexity Theory* (p. 260). Alemania: Springer-Verlag.

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Una Nueva Técnica para Clasificación Utilizando Teoría de Patrones

del (la) C.

Luis Eulalio REAL NOVO

el día 26 de Agosto de 2009.

Dr. Juan Manuel Ramírez Arredondo
Investigador CINVESTAV 3C
CINVESTAV Unidad Guadalajara

Dr. Luis Ernesto López Mellado
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Félix Francisco Ramos Corchado
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Mario Angel Siller González Pico
Investigador CINVESTAV 2A
CINVESTAV Unidad Guadalajara