TESIS DEFENDIDA POR

Miguel Angel Palacios Alonso

Y aprobada por el siguiente comité:

Dr. Carlos Alberto Brizuela Rodríguez

Codirector del Comité

Dr. Luis Enrique Sucar Succar

Codirector del Comité

Dr. Vitali Kober

Miembro del Comité

Dr. Hugo Homero Hidalgo Silva

Miembro del Comité

Dr. Joaquín Alvarez Gallegos

Miembro del Comité

Dr. Pedro Gilberto López Mariscal

Coordinador del programa en Ciencias de la Computación

Dr. David Hilario Covarrubias Rosales

Encargado del Despacho de la Dirección de Estudios de Posgrado

22 de Febrero del 2008.

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN
SUPERIOR DE ENSENADA

PROGRAMA DE POSGRADO EN CIENCIAS
EN CIENCIAS DE LA COMPUTACIÓN

Aprendizaje evolutivo del clasificador Bayesiano simple dinámico

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

MAESTRO EN CIENCIAS

Presenta:

Miguel Angel Palacios Alonso

Ensenada, Baja California, México. Febrero del 2008.

RESUMEN de la tesis que presenta **Miguel Angel Palacios Alonso**, como requisito parcial para obtener el grado de MAESTRO EN CIENCIAS en CIENCIAS DE LA COMPUTACIÓN. Ensenada, B. C. Febrero del 2008.

Aprendizaje evolutivo del clasificador Bayesiano simple dinámico

Resumen aprobado por:

Dr. Carlos Alberto Brizuela Rodríguez
Codirector de Tesis

Dr. Luis Enrique Sucar Succar
Codirector de Tesis

Muchos problemas tales como el reconocimiento de voz, reconocimiento del habla, procesamiento de imágenes y muchas otras tareas han sido tratadas con modelos ocultos de Markov, estos problemas pueden ser tratados también con una extensión del clasificador bayesiano simple (CBS) conocido como el clasificador bayesiano simple dinámico (CBSD).

El CBS trabaja bien en conjuntos de datos con atributos independientes. Sin embargo, su rendimiento disminuye cuando los atributos son dependientes o cuando uno o más atributos irrelevantes son dependientes de algún atributo relevante. Por lo tanto, para incrementar la exactitud de este clasificador se requiere de un método para el diseño de la estructura de una red que pueda capturar las dependencias entre atributos y elimine atributos irrelevantes.

El CBSD es un método probabilístico para clasificación de procesos dinámicos que involucran relaciones temporales, supone que los procesos que modela son estacionarios (los parámetros no cambian con el tiempo) y Markovianos (la probabilidad del estado futuro es independiente del pasado dado el presente), además de que supone independencia condicional de los atributos dada la clase.

En el problema de aprendizaje del CBSD deben tomarse en cuenta además de la suposición de independencia heredada por el CBS, las relaciones temporales que describe el proceso dinámico. Por lo tanto, realizando mejora estructural, se requiere un método de aprendizaje que permita eliminar atributos irrelevantes, determinar la asociación de atributos (dependientes) correspondientes a los nodos hijo y determinar el número óptimo de estados del nodo clase oculto.

En este trabajo se propone un algoritmo de optimización evolutivo para resolver este problema de diseño. Se propone un nuevo esquema de codificación y sus respectivos operadores genéticos, los cuales son extensiones naturales de codificación y operadores para problemas de agrupamiento. La metodología diseñada es aplicada para resolver el problema de reconocimiento de nueve ademanes de la mano derecha.

Los resultados experimentales muestran que la red evolucionada tiene una exactitud de clasificación más alta que el clasificador bayesiano simple dinámico básico con una mejora promedio de 2.7% en el reconocimiento de los ademanes.

Palabras clave: Clasificador bayesiano simple, Redes Bayesianas Dinámicas, Cómputo evolutivo, Reconocimiento de ademanes, Algoritmos Genéticos.

ABSTRACT of the thesis presented by **Miguel Angel Palacios Alonso**, as a partial requirement to obtain the MASTER SCIENCE degree in COMPUTER SCIENCES. Ensenada, B. C. February 2008.

Evolutionary learning of dynamic Bayesian naive classifier

Abstract approved by:

Dr. Carlos Alberto Brizuela Rodríguez
Thesis codirector

Dr. Luis Enrique Sucar Succar
Thesis codirector

Many problems such as voice recognition, speech recognition, image processing and others have been dealt with Hidden Markov Models, these kind of problems can also be dealt with an extension of the naive Bayesian classifier known as dynamic naive Bayesian classifier.

The naive Bayesian classifier works well on data set with independent attributes. However, when attributes are dependent of each other or when one or more irrelevant attributes are dependent of relevant ones their performance decrease considerably. Therefore, to increase this classifier accuracy we need a method to design network structures that can capture the dependencies and get rid of irrelevant attributes.

The dynamic naive Bayesian classifier is a probabilistic method for dynamic processes classification that involve temporal relations, it assumes that the processes are stationary (the parameters do not change with time) and Markovians (the probability of the future state is independent from the past in view of the present), besides that conditional independence of the attributes is supposed given the class.

In the learning problem, besides the independence assumption, temporal relations describing the dynamical process should also be considered. Therefore, the learning method should be capable of eliminating irrelevant attributes, determining the attributes associations that corresponds to the children nodes and also determining the optimal number of states for the hidden class node.

We propose an evolutionary optimization algorithm to solve this design problem. We introduce a new encoding scheme and new genetic operators which are natural extensions of previously proposed encoding and operators for grouping problems. The design methodology is applied to solve the recognition problem for nine hand gestures.

Experimental results show that the evolved network has higher average classification accuracy than the basic dynamic naive Bayesian classifier with an average improvement of 2.5% in gestures recognition.

Keywords: Naive Bayes Classifier, Dynamic Bayesian Networks, Evolutionary Computation, Gesture Recognition , Genetic Algorithms.

A la persona que más admiro,
una heroína de la vida real:
mi madre

Agradecimientos

A mi madre, por siempre mostrarme el cómo a través de un medio de enseñanza infalible ... el ejemplo. Gracias mamá por todo el cariño y apoyo que me has brindado.

Gracias a mis hermanos por el apoyo incondicional que cada uno me ha mostrado a lo largo de mis estudios y de mi vida, gracias Paty, gracias Chayo, gracias Adan.

A mis sobrinas Maribel y Chayo y mi sobrino Julio por poner mucha de la alegria que se vive en casa.

A mi asesor el Dr. Carlos Brizuela por apoyarme en todo momento, por darme la libertad y confianza de desenvolverme en este trabajo de tesis.

A mi asesor el Dr. L. Enrique Sucar por aceptarme como tesista y por estar siempre en la disposición de ayudarme.

A mi comite de tesis Dr. Vitaly Kover, Dr. Hugo Hidalgo Silva y Dr. Joaquín Alvarez Gallegos por todos sus comentarios.

A mis compañeros y amigos de la generación 2005 (Jacobo, Mario, Caloca, Torito, Argelia, Noe, Bernardino, Ariel, Edna, Dayra, Gilberto, Giovana) por compartir conmigo esta aventura.

A mis compas de la generación 2006 por arroparme en el último tramo de este camino.

Un agrademiento especial a Hector H. Avilés por que desde el primer momento en que inicie este trabajo estuvo siempre presente y en disposición de ayudarme.

También un agradecimiento especial a Everardo por que con sus comentarios ayudó a aclarar muchas de mis dudas y a hacer más ligero el camino.

A todo CICESE por reforzar en mi la idea de que la calidad científica y la calidad humana no tienen por que ir en direcciones opuestas.

A CONACYT por el apoyo económico otorgado para la culminación de este trabajo.

A todos,
GRACIAS por ESTAR ... y por SER.

Ensenada, México
22 de Febrero del 2008.

Miguel Angel Palacios Alonso

Tabla de Contenido

Capítulo Página

Tabla de Contenido (Continuación)

Tabla de Contenido (Continuación)

Lista de Figuras

Lista de Figuras (Continuación)

Lista de Tablas

Capítulo I

Introducción

I.1 Motivación y antecedentes

La clasificación es una tarea básica en el análisis de datos y reconocimiento de patrones. Un clasificador, es una función que asigna una clase a casos descritos por un conjunto de atributos (Friedman *et al.*, 1997).

Uno de los clasificadores más utilizado es el Clasificador Bayesiano Simple (CBS), el cual supone independencia condicional de los atributos dada la clase. Este es ampliamente utilizado debido a que es fácil de construir y de entender (Duda *et al.*, 2001; Martínez, 2006).

El CBS es sorprendentemente efectivo en la práctica; su decisión de clasificación puede ser a menudo correcta aún si sus estimaciones de probabilidad son inexactas (Rish, 2001). Sin embargo, es sabido que cuando los atributos que maneja el CBS son dependientes, o cuando uno o más atributos irrelevantes tienen algún grado de dependencia de atributos relevantes, el rendimiento del CBS disminuye considerablemente (Pazzani, 1996).

Para diseñar clasificadores más exactos, se ha propuesto la relajación de las suposiciones de independencia y diferentes métricas han sido evaluadas para el aprendizaje de clasificadores bayesianos. Cuando se intenta relajar las suposiciones de independencia de los atributos dada la clase, de manera indirecta se realiza un proceso que es conocido como aprendizaje estructural. Por lo que mejorar la estructura del CBS produce mejoras en la exactitud.

Buscando la mejora de la estructura en el CBS se han propuesto distintas opciones como permitir enlaces entre los nodos hijo (Friedman *et al.*, 1997) o bien métodos en los que variables dependientes son fusionadas como un mismo nodo (Pazzani, 1996).

Avilés-Arriaga *et al.* (2003) proponen una extensión del CBS para poder modelar procesos dinámicos, el llamado Clasificador Bayesiano Simple Dinámico (CBSD), el cual muestra mejor rendimiento que los Modelos Ocultos de Markov (MOM) cuando el número de muestras de entrenamiento es pequeño. Siendo la base del CBSD, el CBS hereda sus características a éste.

El CBSD puede ser visto como una Red Bayesiana Dinámica y desde esta perspectiva pueden ser utilizados algoritmos existentes de aprendizaje paramétrico y estructural. En el problema de aprendizaje del CBSD deben tomarse en cuenta, además de la suposición de independencia heredada por el CBS, las relaciones temporales que describen el proceso dinámico. Se requiere entonces un método capaz de resolver la existencia de un nodo oculto. Por lo tanto, realizando mejora estructural, un método de aprendizaje para el CBSD debe determinar tanto la asociación de atributos (dependientes) correspondientes a los nodos hijo, así como el número de estados del nodo oculto.

Avilés-Arriaga (2006) presenta un sistema de reconocimiento visual de ademanes aplicado al control de un robot móvil. Se comparan cuatro tipos de modelos utilizados en la fase de clasificación, el CBSD, los MOMs, redes bayesianas y las redes lógico-probabilistas. La estructura de todos los modelos fueron definidas con base en el conocimiento del experto. Avilés-Arriaga *et al.* (2006) muestran la conveniencia de utilizar el CBSD, ya que requiere menos cálculos que el MOM, al mismo tiempo que mantiene un porcentaje de reconocimiento competitivo con los MOMs. Presenta evidencia empírica donde muestra que además de información de movimiento se requiere información de postura para incrementar el porcentaje de clasificación, aun con ademanes similares.

Martínez (2006) presenta una metodología para el aprendizaje del CBS y del CBSD

basado en búsqueda estándar, donde además de determinar la estructura del clasificador (dinámico o estático) obtiene la discretización correspondiente en caso de que se utilicen variables continuas. La búsqueda está dividida básicamente en tres etapas secuenciales, la determinación de los intervalos de las variables continuas, la determinación del número de estados del nodo clase, y finalmente la mejora estructural (asociación de variables dependientes y eliminación de variables irrelevantes).

Siguiendo los enfoques propuestos por Sucar *et al.* (1993, 1994) y Pazzani (1996) podemos identificar dependencias entre atributos, por lo que cada uno de los nodos hijo del CBSD pueden ser vistos como grupos conteniendo uno o más atributos. Sin embargo este problema crece exponencialmente con el número de atributos. Por lo tanto, no podemos explorar exhaustivamente el espacio de solución, aún para un número pequeño de atributos, por lo que se requiere una alternativa a la fuerza bruta para encontrar un agrupamiento G óptimo o cercano al óptimo. Además el manejo de información incompleta complica el problema de aprendizaje. Algoritmos como EM (Expectation-Maximization) (Dempster *et al.*, 1977) son necesarios para estimar los datos faltantes. Lo que provoca un espacio de búsqueda grande y multimodal (Friedman, 1998). Algoritmos determinísticos son propensos a obtener optimos locales. Múltiples reinicios del proceso de aprendizaje se han sugerido como una forma de tratar con este problema.

Una opción para enfrentar el problema con óptimos locales, es utilizar un método de búsqueda estocástico. Este trabajo explora el uso de cómputo evolutivo, en particular de Algoritmos Genéticos (AGs), para el aprendizaje del CBSD.

Los AGs han demostrado ser un enfoque general y flexible que se adapta a cualquier problema de búsqueda y optimización. Una de las principales diferencias con respecto a los métodos clásicos, es que los AGs utilizan una población de soluciones por iteración, en lugar de una sola solución. Para mover la población de soluciones sobre el espacio de búsqueda, los AGs se basan en el principio de selección natural, donde los más aptos

sobreviven y prosperan.

I.2 Planteamiento del problema

Debido a que el CBSD, al igual que el CBS, está basado en la suposición de independencia de los atributos dada la clase, la existencia de dependencias entre los atributos y la existencia de atributos irrelevantes disminuye su rendimiento. Además deben considerarse las relaciones temporales del CBSD, la cuales son no observables y deben estimarse. Por lo tanto, para diseñar un clasificador más exacto es necesario considerar los siguientes problemas:

- Selección de atributos
- Manejo de dependencias en atributos
- Manejo de variables ocultas

Esto es, el diseño de un algoritmo para el aprendizaje del CBSD debe considerar las dependencias que pueden existir entre los datos disponibles (atributos), así como seleccionar sólo la información relevante. Una vez identificadas las dependencias, éstas deben ser representadas de alguna manera en la estructura del modelo obtenido. Si consideramos que la representación de dependencias es por medio de unión de atributos (Pazzani, 1996), el problema de búsqueda que identifica dependencias y elimina atributos irrelevantes se vuelve exponencial con respecto al número de atributos disponibles.

Por otro lado, en procesos del mundo real, generalmente se pueden identificar sus salidas. Sin embargo el origen de dichas salidas es desconocido. En una base de datos a esta falta de la información se le conoce como variable oculta, es decir, una variable para la cual no se ha tenido lectura de ninguno de sus valores en cada una de las muestras tomadas de los datos. La existencia de este tipo de variables requiere que se utilicen

métodos más costosos para obtener los parámetros cuando el proceso es modelado. El CBSD al modelar procesos dinámicos cuenta con un nodo oculto para representar este proceso. La determinación del mejor número de estados que puede tomar este nodo es un problema a resolver.

I.3 Objetivos

I.3.1 Objetivo general

Desarrollar un algoritmo evolutivo para el aprendizaje de CBSDs, este algoritmo considera el problema tratado con variables aleatorias discretas y el nodo clase oculto. El algoritmo propuesto debe ser capaz de determinar el número de estados del nodo oculto, la asociación de atributos dependientes y eliminación de atributos irrelevantes.

I.3.2 Objetivos específicos

- Diseñar una representación adecuada para evolucionar el CBSD.
- Diseñar operadores genéticos para la representación propuesta.
- Diseñar una función de aptitud que guíe la evolución del CBSD.
- Evaluar el algoritmo propuesto en el reconocimiento de ademanes de la mano derecha.

I.4 Propuesta de solución

Debido a la naturaleza exponencial del problema de aprendizaje estructural se propone utilizar un algoritmo evolutivo para determinar soluciones cercanas al óptimo para el

número de estados del nodo oculto y la asociación de atributos. Se propone evolucionar CBSDs con el fin de tener un diseño que agrupe atributos dependientes y elimine atributos irrelevantes. Se propone una codificación especial, los operadores genéticos correspondientes y una función objetivo que considera la exactitud de clasificación sobre un conjunto de prueba parcial.

I.5 Metodología de investigación

Para realizar este trabajo de investigación se aplicó la siguiente metodología:

- **Investigación del tema**.

  Con el fin de identificar las características de una Red Bayesiana (RB) se hizo una revisión del estado del arte de redes bayesianas (RBs) y Redes Bayesianas Dinámicas (RBDs). Se analizó el uso de una RB como clasificador, las ventajas y desventajas del Clasificador Bayesiano Simple (CBS) y cada una de sus variantes. Se revisaron los trabajos presentados por Avilés-Arriaga (2006) y Martínez (2006). Avilés-Arriaga (2006) realiza un sistema de reconocimiento visual de ademanes completo como se muestra en la Figura 21, donde el experto define la estructura del clasificador. Martínez (2006) realiza un proceso de aprendizaje similar al presentado en este trabajo, pero realizando búsqueda estándar y llevando a cabo discretización para CBS estáticos y dinámicos.

- **Propuesta del enfoque evolutivo**.

  Se diseñó un algoritmo para el aprendizaje del CBSD basado en cómputo evolutivo. Se analizó y adaptó la representación basada en grupos propuesta por Falkenauer (1994) para representar un individuo. Se analizaron diferentes métricas para evaluar la calidad de una red, de tal forma que pudiese ser utilizada como la función

de aptitud del enfoque evolutivo. Se definieron las partes básicas restantes del enfoque evolutivo como son: la inicialización de la población, la selección de padres y sobrevivientes, los operadores genéticos (cruzamiento y mutación) y se definió el criterio de paro.

- **Análisis y preparación de datos**.

  En este trabajo se utiliza la base de datos proporcionada por Avilés-Arriaga (2006) correspondiente a nueve ademanes de la mano derecha realizadas por un usuario. Para poder ser interpretados por el sistema la información obtenida del sistema visual es procesada para obtener la variables aleatorias discretas correspondientes al proceso modelado.

- **Experimentación**

  Con el fin de evaluar el comportamiento del sistema de aprendizaje del clasificador bayesiano simple dinámico (CBSD) se realizaron 4 experimentos en el reconocimiento de nueve ademanes de la mano.

I.6 Organización de la tesis

El resto del documento está organizado de la siguiente manera:
En los capítulos II al IV se presentan el marco teórico necesario para la comprensión del problema.
En el Capítulo II se presentan los problemas de inferencia y aprendizaje para redes bayesianas estáticas y dinámicas.
En el Capítulo III se presentan los diferentes tipos de clasificadores bayesianos existentes en la literatura y la definición del CBSD.
En el Capítulo IV se presentan los conceptos básicos utilizados en cómputo evolutivo y de cómo estos son utilizados para dar solución a nuestro problema.

En el Capítulo V se presenta la metodología propuesta para el aprendizaje del CBSD.

El Capítulo VI presenta los experimentos en el reconocimiento visual de ademanes y los resultados obtenidos.

En el Capítulo VII se discuten las conclusiones y el trabajo futuro.

Capítulo II

Redes bayesianas

II.1 Introducción

Los modelos gráficos probabilísticos son modelos de interacciones ("causales") entre un conjunto de variables aleatorias (Kjærulff y Madsen, 2005). Este tipo de modelos capturan un conjunto de propiedades de independencia condicional asociada con las variables representadas en la red. Son capaces de representar de manera gráfica la distribución de probabilidad conjunta que recae sobre los datos modelados.

Muchas situaciones de la vida real pueden ser modeladas como un dominio de entidades representadas como variables aleatorias en una red probabilística. Dicha red representa y procesa conocimiento probabilístico, describe el conocimiento del dominio de un problema de una manera precisa. La representación gráfica utilizada por este tipo de modelos es intuitiva y fácil de comprender, haciéndola una herramienta para la comunicación de conocimiento entre expertos, usuarios y sistemas.

Un modelos gráfico probabilístico se construye en dos fases:

- Fase 1. Se define la estructura cualitativa del modelo usando un lenguaje gráfico. Este paso consiste en identificar variables y relaciones entre variables.

- Fase 2. Consiste en calcular los parámetros definidos por la representación cualitativa obtenida en el paso anterior.

II.2 Redes bayesianas

Un modelo de interacción probabilística entre un conjunto de variables aleatorias puede ser representado como una distribución de probabilidad conjunta. Considerando el caso donde las variables aleatorias son discretas, el tamaño de la distribución de probabilidad conjunta crece exponencialmente con el número de variables, por lo que se requiere de una representación más compacta para el razonamiento acerca del estado de sistemas grandes y complejos que involucran un gran número de variables.

Las Redes Bayesianas (RBs) son populares dentro de la comunidad de inteligencia artificial, facilitan una representación eficiente y son apropiadas para la representación de conocimiento en situaciones que involucran razonamiento bajo incertidumbre (Kjærulff y Madsen, 2005).

Una RB permite realizar inferencia probabilística para predecir la salida de algunas variables con base en las observaciones de otras. Las RBs son utilizadas en sistemas de clasificación y diagnóstico. Por ejemplo, MUNIN es utilizado para diagnóstico de enfermedades en músculos y nervios , y PATHFINDER es utilizado para diagnóstico de enfermedades del nódulo linfático (Jensen, 1997). También son utilizadas en recuperación de información (Heckerman y Horvitz, 1998) y en localización y solución de problemas en impresoras (Heckerman y Wellman, 1995).

Una RB es un Grafo Dirigido Acíclico (GDA) y conexo (Pearl, 1988) en el cual:

- los nodos representan variables,
- los arcos indican la existencia de influencias ("causales") directas entre la variables enlazadas,
- la fortaleza de estas influencias son expresadas por probabilidades condicionales.

Una RB puede ser denotada como $R = (\mathcal{G}, \Theta)$, donde $\mathcal{G}$ es el GDA y Θ son los

parámetros asociados a cada uno de los nodos, es decir las tablas de probabilidad condicionales.

Dependiendo del dominio modelado $\mathcal{G}$ puede subdividirse en tres tipos (Neapolitan, 1990) (ver Figura 1). Un árbol es la estructura más simple, donde existe un nodo raíz y todos los demás nodos pueden tener a lo más un padre. Un poli-árbol puede contener nodos con dos o más padres. Una red multiconectada, puede contener más de una trayectoria entre parejas de nodos.

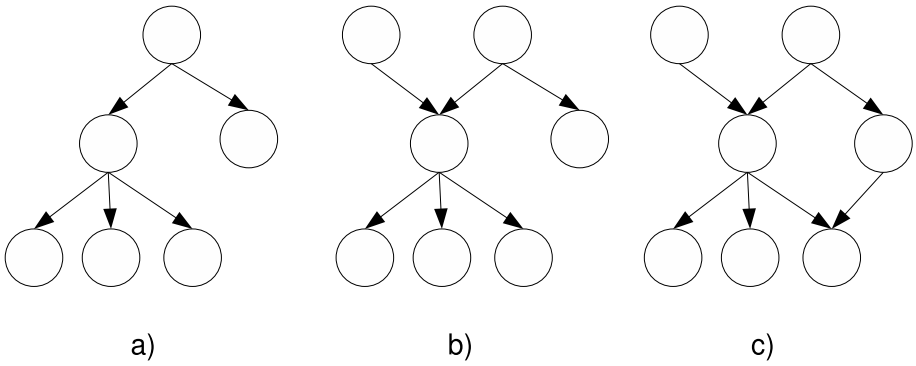

Figura 1. Estructuras de RB. a) Árbol, b)Poli-árbol, c)Red.

Las RBs también pueden denotarse en la literatura como redes de creencia, redes causales y mapas de conocimiento.

La semántica de una RB exige una clara correspondencia entre la topología de un GDA y las relaciones de dependencia representadas por ella. La representación gráfica de una RB permite descomponer (factorizar) la función de distribución de probabilidad en términos de relaciones de independencia condicional definida sobre subconjuntos de variables. Una RB puede ser vista como un instrumento de inferencia para deducir nuevas relaciones de independencia desde las ya construidas en la red.

Dado que una RB es un modelo probabilístico puede entonces describirse en dos niveles:

- el nivel cualitativo o gráfico (fase 1) que muestra la estructura de las variables en el grafo.

- el nivel cuantitativo (fase 2) que corresponde a las probabilidades condicionales del modelo, es decir la probabilidad de un nodo dados sus padres.

Sea $X = \{X_1, X_2, \ldots, X_n\}$ un conjunto de n variables aleatorias y $P(X)$ la función de distribución de probabilidad definida sobre X. Sin conocimiento alguno acerca de las dependencias entre variables aleatorias, la distribución de probabilidad de X puede expresarse a través de la regla de la cadena como sigue (Papoullis y Pillai, 2002):

$$\begin{aligned} P(X_1, X_2, \ldots, X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_2, X_1) \\ &\ldots \quad P(X_n|X_{n-1}, \ldots, X_1) \end{aligned} \tag{1}$$

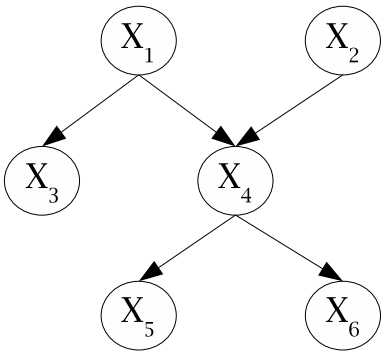

Figura 2. Ejemplo de una RB con seis variables o nodos en la red. Representación gráfica de dependencias en $P(X_1, X_2, X_3, X_4, X_5, X_6)$. Cada nodo en la red representa una variable aleatoria.

Por otro lado, si tenemos algún conocimiento del proceso a modelar, resolver la ecuación (1) resultaría en una serie de cálculos innecesarios. El conocimiento disponible puede ser reflejado e interpretado de manera gráfica por medio de una RB. Por ejemplo, supongamos que tenemos la RB de la Figura 2, esta red contiene seis nodos. Si

consideramos que hay una relación de uno a uno entre los nodos de la red y las variables aleatorias del modelo entonces tenemos que el fenómeno está integrado por seis variables aleatorias. Esta misma red muestra también las suposiciones de independencia entre variables; estas suposiciones son mostradas a través de los arcos que conectan a cada nodo/variable, por lo que se puede observar que la variable X_4 es influenciada por X_1 y X_2. En otras palabras, dados los valores de X_1 y X_2, X_4 es independiente condicionalmente de sus no descendientes. La probabilidad conjunta que refleja la RB de la Figura 2 sobre las variables $\{X_1, X_2, X_3, X_4, X_5, X_6\}$ puede obtenerse de la siguiente manera:

$$P(X_1, X_2, X_3, X_4, X_5, X_6) = P(X_6|X_4)P(X_5|X_4)P(X_4|X_1, X_2)P(X_3|X_1)P(X_2)P(X_1) \tag{2}$$

Tenemos entonces que cada factor del lado derecho es una función de probabilidad condicional, que denota la dependencia probabilística entre las variables del modelo y que permite describir las relaciones "causales" y de correlación entre las variables.

A partir de la RB de la Figura 2 puede interpretarse el modelo definido en la ecuación (2) y viceversa.

El nivel cualitativo de la red, es decir, la información reflejada por el grafo de la RB es generalizado por la siguiente ecuación (Pearl, 1988):

$$P(X_1, X_2, \ldots, X_n) = \prod_{m=1}^{n} P(X_m|Pa(X_m)) \tag{3}$$

donde $Pa(X_m)$ representa el subconjunto de X cuyos elementos están directamente influenciando a X_m. El subconjunto $Pa(X_m)$ usualmente es llamado los "ancestros" o "padres" de X_m.

II.2.1 Inferencia

La tarea de inferencia consiste en deducir cuál es la distribución sobre un conjunto particular de variables aleatorias dado que conocemos los estados de algunas otras variables

en la red (Pavlovic, 1999).

Sea $E = \{E_1, \ldots, E_n\}$ el conjunto de variables evidencia, e, un evento particular observado, X la variable consultada, y $Y = \{Y_1, \ldots, Y_n\}$ las variables ocultas. El objetivo de la tarea de inferencia es encontrar la distribución de probabilidad $P(X|e)$. Observe que X puede ser parte del conjunto evidencia o del conjunto de variables ocultas.

De aquí en adelante los datos son evidencia, esto es, instancias de algunas o de todas las variables aleatorias describiendo el dominio.

Cualquier probabilidad condicionada o marginal se puede obtener a partir de la probabilidad conjunta. La probabilidad conjunta de las variables de una RB es el producto de todas las probabilidades condicionales incluidas en la red. Desafortunadamente, aunque este método para el cálculo de las probabilidades a posteriori de las variables parece el más inmediato, su complejidad crece exponencialmente con el número de nodos de la red (Neapolitan, 1990).

Explotando independencias locales, el algoritmo de paso de mensajes (Pearl, 1988) para inferencia en RBs determina $P(X|e)$ para todos los valores e de la variable E de la red. El paso de mensajes es realizado iniciando mensajes desde cada variable instanciada hacia sus vecinos, a su vez éstos pasan mensajes a sus vecinos. La evidencia puede llegar en cualquier orden. Este método es aplicable a RBs con estructura de árbol y poli-árbol.

Debido a que la inferencia puede verse como una marginalización de distribuciones conjuntas, el método de eliminación de variables consiste en distribuir sumas dentro de productos, esto para evitar cálculos innecesarios. Este algoritmo requiere como entrada un orden de eliminación, el cual es un problema NP-completo (Yannakakis, 1981).

Uno de los algoritmos basados en eliminación de variables es el algoritmo de agrupamiento (Murphy, 2002). Uno de los pasos en este tipo de algoritmos es la construcción de un árbol de unión (llamado asi por que los nodos que lo conforman son agrupaciones o cúmulos de variables). El modo general de construir un árbol de unión consiste en

moralizar el grafo de la RB (insertar enlaces entre los padres de un nodo) y triangularizar el grafo no dirigido asociado resultante, cada clique se convierte en un cúmulo del árbol de unión. Después de asignar cada tabla de probabilidades condicionales de la red a un cúmulo e introducir la evidencia, el cálculo de las probabilidades a posteriori se realiza mediante el intercambio de mensajes entre cúmulos vecinos. Este algoritmo permite inferencia en RB multiconectadas.

Los esquemas antes mencionados son métodos de inferencia exacta. En el peor caso, la inferencia exacta es NP-difícil (Dagum y Luby, 1993), por lo que se necesita recurrir a aproximaciones. Algunos métodos de inferencia aproximada son: propagación de creencias con bucles (Pearl, 1988), propagación de la esperanza (Minka, 2001), métodos variacionales (Jordan *et al.*, 1998) y métodos de muestreo (Neal, 1993).

II.2.2 Aprendizaje

El aprendizaje bayesiano puede describirse como: dado un conjunto de entrenamiento $D = \{D_1, D_2, \ldots, D_n\}$, encontrar la red que se ajuste mejor a D.

El aprendizaje puede ser de dos tipos:

- Parámetrico: consiste en obtener las probabilidades condicionales (θ) de cada uno de los nodos.
- Estructural: consiste en obtener la estructura del GDA de la RB, este caso incluye el aprendizaje parámetrico.

II.2.2.1 Estimación de parámetros

El aprendizaje de parámetros consiste en determinar las probabilidades condicionales de un modelo cuya estructura es conocida. Dependiendo de la naturaleza de lo datos estos pueden dividirse en dos:

- Aprendizaje con datos completos. Los valores de todas las variables fueron observados en cada uno de los eventos de la base de datos.

- Aprendizaje con información incompleta y/o variables ocultas. Se le llama información incompleta cuando en el conjunto de muestras algunos de los valores de algunas variables no fueron observados. Las variables ocultas son variables que no fueron observadas en ninguna de las muestras que conforman la base de datos.

En la Figura 3 se muestra un ejemplo de dos bases de datos, una con datos completos y otra con información incompleta y una variable oculta.

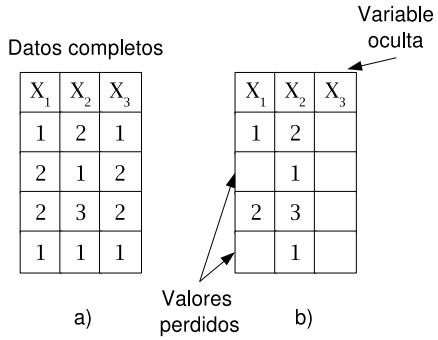

Figura 3. Dos bases de datos con tres variables aleatorias en a) los datos están completos, en b)X_1 contiene información incompleta, X_3 es una variable oculta.

1. **Estimación de parámetros con datos completos**

   Sea $D = \{D_1, \ldots, D_M\}$ un conjunto de datos, entonces la verosimilitud logarítmica del conjunto de datos se define como:

$$logP(D|\Theta, \mathcal{M}) = log \prod_{m=1}^{M} P(D_m|\Theta, \mathcal{M})] = \sum_{i=1}^{n} \sum_{m=1}^{M} logP(X_i|Pa(X_i), D_m) \quad (4)$$

   donde $\mathcal{M}$ es el modelo, Θ son los parámetros asociados y $Pa(X_i)$ es el conjunto de padres del nodo X_i.

En el caso de distribuciones multinomiales, donde definimos $\theta_{ijk} = P(X_i = k|Pa(X_i) = j)$, la verosimilitud logarítmica se convierte en

$$L(\Theta) = \sum_i \sum_m log \prod_{j,k} \theta_{ijk}^{I_{ijkm}} \quad (5)$$

$$= \sum_i \sum_m log \sum_{jk} I_{ijkm} log\theta_{ijk} \quad (6)$$

$$= \sum_{ijk} N_{ijk} log\theta_{ijk} \quad (7)$$

donde $I_{ijkm} = I(X_i = k, Pa(X_i) = j|D_m)$ es 1 si el evento $(X_i = k, Pa(X_i) = j)$ ocurre en el caso D_m, y de aqui $N_{ijk} = \sum_m I(X_i = k, Pa(X_i) = j|D_m)$ es el número de veces que el evento fue visto en el conjunto de datos de entrenamiento.

El estimador de máxima verosimilitud es simplemente una tabla normalizada conteniendo conteos de cada configuración de X_i, dada cada configuración de sus padres en el conjunto de datos, es decir, $\hat{\theta}$ queda definido de la siguiente manera:

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{\sum_{k'} N_{ijk'}} \quad (8)$$

donde la sumatoria sobre k' incluye a todos los valores que puede tomar X_i.

2. **Estimación de parámetros con variables ocultas: El algoritmo EM (Expectation-Maximization)**

En problemas del mundo real nos encontramos con la existencia de variables ocultas, las cuales no son observables en los datos disponibles para realizar el aprendizaje. Las variables ocultas pueden reducir el número de parámetros requeridos para especificar una RB (Russell y Norvig, 2003). Esto, puede reducir la cantidad de datos necesarios para aprender los parámetros. Sin embargo, las variables ocultas complican el problema de aprendizaje debido a que la verosimilitud logarítmica

no puede ser descompuesta como en la ecuación (7). Con variables ocultas tenemos que:

$$L(\Theta) = logP(D_m|\Theta) = \sum_m log \sum_z P(Z = z, D_m|\Theta) \tag{9}$$

donde Z es el conjunto de variables ocultas, y $\sum_z$ es la suma sobre Z requerida para obtener la probabilidad marginal de los datos.

El algoritmo EM permite obtener los valores probables de las variables ocultas en cada uno de los casos de observación y para ello se auxilia del módulo de inferencia.

La idea básica detrás de EM es aplicar la desigualdad de Jensen (Cover y Thomas, 1991) a la ecuación (9) como sigue (Murphy, 2002):

$$\begin{aligned}
L &= \sum_m log \sum_z P(Z = z, D_m|\theta) && (10)\\
&= \sum_m log \sum_z q(z|D_m) \frac{P(Z = z, D_m|\theta)}{q(z|D_m)} && (11)\\
&\geq \sum_m \sum_z q(z|D_m) log \frac{P(Z = z, D_m|\theta)}{q(z|D_m)} && (12)\\
&= \sum_m \sum_z q(z|D_m) logP(Z = z, D_m|\theta) - \sum_m \sum_z q(z|D_m) logq(z|D_m) && (13)
\end{aligned}$$

donde q es cualquier distribución sobre las variables ocultas.

Este proceso es realizado para obtener una cota inferior sobre la verosimilitud logarítmica, y entonces iterativamente maximizar esta cota inferior. El algoritmo EM alterna entre maximizar la cota inferior con respecto a q y θ, respectivamente, manteniendo al otro fijo. Maximizando la cota inferior con respecto a q se tiene

$$q(z|D_m) = P(z|D_m, \theta) \tag{14}$$

Este es el paso de cálculo del valor esperado (paso E), y hace que la cota se ajuste. Maximizar la cota inferior con respecto a los parámetros θ' es equivalente a maximizar la esperanza de la verosimilitud logarítmica de los datos completos.

Si se utiliza $q(z|D_m) = P(z|D_m, \theta)$ tenemos

$$Q(\Theta'|\Theta) = \sum_m \sum_z P(z|D_m, \theta) log P(Z = z, D_m|\theta') \tag{15}$$

El paso de maximización (paso M) ajusta los parámetros del modelo, es decir, maximiza la verosimilitud logarítmica de los datos considerando que estos están completos. Dempster *et al.* (1977) probaron que seleccionar Θ' de tal forma que $Q(\Theta'|\Theta) > Q(\Theta|\Theta)$ garantiza que $P(D|\Theta') > P(D|\Theta)$, por lo que el algoritmo puede converger a máximos locales.

Para datos multivariados, el valor esperado de la verosimilitud logarítmica de los datos "completados" $Q(\Theta' : \Theta)$ es de la siguiente forma:

$$\begin{aligned} Q(\Theta' : \Theta) &= E[log P(Z, Y|\Theta')] & (16) \\ &= \sum_{ijk} E[N_{ijk}] log \theta'_{ijk} & (17) \end{aligned}$$

donde $E[N_{ijk}] = \sum_m P(X_i = k, Pa(X_i) = j|D_m, \theta)$ es obtenido por inferencia en la red $(\mathcal{G}, \Theta)$, si el conjunto $\{X_i, Pa(X_i)\}$ no es medido completamente, o por conteo en caso contrario. El paso de maximización se reduce a calcular:

$$\hat{\theta}_{ijk} = \frac{E[N_{ijk}]}{\sum_{k'} E[N_{ijk'}]} \tag{18}$$

II.2.2.2 Estimación de la estructura de una red bayesiana

La estructura de la red representa conocimiento causal acerca del dominio que es a menudo proporcionado por un experto, pero esto no siempre es posible por lo que es importante comprender cómo la estructura de una RB puede ser aprendida a partir de datos.

Dados datos de entrenamiento D, el aprendizaje estructural es la tarea de encontrar un conjunto de aristas dirigidas (entre las variables aleatorias) que mejor modele la

densidad verdadera de los datos. Si se consideran modelos de RBs con n variables, el número de posibles estructuras es exponencial. Robinson (1977) mostró que $r(n)$, el número de estructuras posibles para RBs teniendo n nodos, está dado por la fórmula de recurrencia

$$r(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} r(n-i) \tag{19}$$

Chickering (1995) muestra que encontrar la mejor red desde el conjunto de todas las redes en las cuales un nodo puede tener 2 padres o más es NP-difícil.

Existen dos clases de métodos para el problema de estimación de la estructura de una RB (López, 2005)

- Selección del modelo por búsqueda y puntaje. El proceso de búsqueda es controlado por un puntaje, el cual busca maximizarse o minimizarse dependiendo de la medida de calidad que se utilice para evaluar la red obtenida. Las medidas de calidad pueden ser bayesianas, de mínima longitud de codificación y medidas de información teórica (Schwarz, 1978; Heckerman, 1995; Lam y Bacchus, 1994).

- Selección del modelo utilizando análisis de dependencias. La estimación de la estructura de una RB usando el análisis de dependencia se basa en el uso de pruebas sobre subconjuntos de arcos en la red (Martínez, 2006; Pazzani, 1996).

II.3 Redes bayesianas dinámicas

La mayor parte de los acontecimientos que encontramos en la vida diaria no son llevados a cabo en un punto particular en el tiempo, dichos eventos pueden ser descritos a través de múltiples estados de observaciones que producen un juicio de un evento completo. Una RB no es capaz de representar dichas dependencias temporales, por lo que han sido extendidas para modelar procesos temporales.

Las Redes Bayesianas Dinámicas (RBDs) describen un sistema que está cambiando o evolucionando con el tiempo. Este modelo permite a los usuarios monitorear y actualizar el sistema conforme avanza el tiempo. Todos los nodos, enlaces y probabilidades que forman la interpretación estática de un sistema son idénticos a una RB. Las variables aquí pueden ser denotadas como el estado de una RBD, porque incluyen una dimensión de tiempo. Los estados de cualquier sistema descrito como RBD satisfacen dos condiciones (Pavlovic, 1999):

- Describe un proceso de Markov: El estado futuro de un sistema es independiente de su pasado dado el presente.
- Una RBD describe un proceso estacionario, es decir los parámetros no cambian con respecto al tiempo.

Como puede observarse, las cadenas de Markov son un ejemplo específico de RBDs. Sin embargo, los estados de una RBD no necesitan ser directamente observables; pueden ser influenciados por algunas otras variables que un observador puede medir directamente.

Una RBD es una forma de extender RBs, donde cada RB describe el proceso en un tiempo t. La relación temporal entre estas redes se define por funciones de probabilidad condicional establecidas entre algunas de sus variables. La Figura 4 muestra una RBD como una red 2TBN (two Time Bayesian Network) debido a que sólo se requieren dos instancias de la RBD para describirla, la red inicial y la red para cualquier tiempo t (recordemos que una RBD describe un proceso estacionario por lo que los parámetros para $t >= 2$ son los mismos).

Denotaremos por $X_i[t]$ al atributo observado i, donde t indica el tiempo. En esta figura cada RB en el tiempo t está compuesta por los 4 nodos/variables temporales

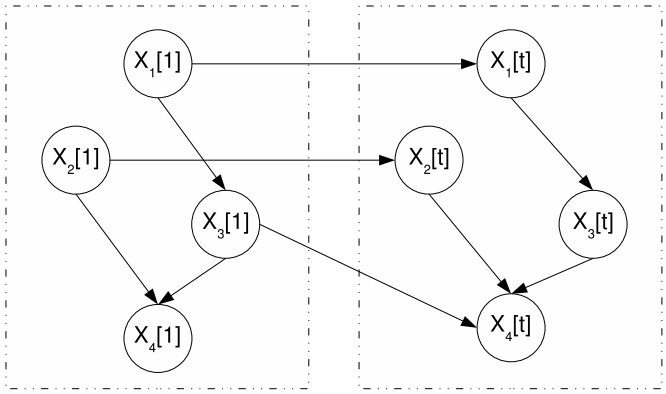

Figura 4. Ejemplo de una RBD representada como una red 2TBN

$\{X_1[t], X_2[t], X_3[t], X_4[t]\}$. Las relaciones de dependencia de cada RB en el tiempo t están contenidas en un mismo rectángulo. Las flechas que inician en un rectángulo y terminan en otro indican las conexiones temporales de la red dinámica.

Una RBD consiste de la función de distribución de probabilidad sobre la secuencia de T nodos ocultos $S = \{S_1, \ldots, S_T\}$ y la secuencia de T variables observables $O = \{O_1, \ldots, O_T\}$ donde T es el número de observaciones de una secuencia. La distibución de probabilidad conjunta en una RBD puede ser expresada de la siguiente manera (Murphy, 2002; Pavlovic, 1999):

$$P(S, O) = P(S_1) \prod_{t=1}^{T-1} P(S_{t+1}|S_t) \prod_{t=1}^{T} P(O_t|S_t) \tag{20}$$

donde $P(S_1)$ es la probabilidad inicial de los estados, $P(O_t|S_t)$ es la función de probabilidad de las observaciones dados los estados. $P(S_{t+1}|S_t)$ es la distribución de probabilidad de transiciones entre los estados a través del tiempo. Observe que para el ejemplo de la Figura 4, $S \cup O = X$.

II.3.1 Inferencia

Una vez que en las RBDs solamente un conjunto de estados puede ser observado en cada tiempo t, tenemos que calcular todos los estados desconocidos en la red. Esto se realiza por un procedimiento llamado inferencia. El proceso de inferencia en una RBD

puede ejemplificarse de manera clara en una red simple como son los modelos ocultos de Markov o el clasificador bayesiano simple dinámico. En la Figura 5 se puede observar un modelo oculto de Markov representado como una RBD.

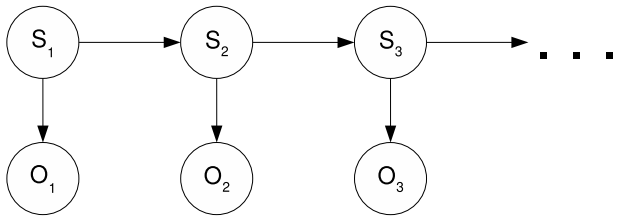

Figura 5. Inferencia en RBDs. Dados los valores de los nodos observados $O = \{O_1, O_2, \ldots\}$ en cada tiempo t, tenemos que estimar los valores de los nodos ocultos S_t . Los nodos S_1, S_2, S_3, ..., son nodos ocultos.

El problema de inferencia en una RBD puede ser visto como el problema de encontrar $P(S_1^T|O_1^T)$ donde O_1^T representa un conjunto finito de T observaciones consecutivas y S_1^T es es el conjunto de variables ocultas correspondientes.

Supongamos que deseamos calcular la probabilidad de la secuencia de observación $O = O_1 O_2 \ldots O_T$ dado el modelo $\mathcal{M}$. Una forma no eficiente de hacer esto es enumerando todas las posibles secuencias de estados de longitud T (el número de observaciones). Sea $S = S_1 S_2 \ldots S_T$ una de estas secuencias, donde S_1 es el estado inicial. La probabilidad de la secuencia de observaciones O para la secuencia de estados S es (Rabiner, 1989):

$$P(O|S, \mathcal{M}) = \prod_{t=1}^{T} P(O_t|S_t, \mathcal{M}) \tag{21}$$

donde suponemos independencia estadística de observaciones. Así, tenemos que:

$$P(O, S|\mathcal{M}) = P(O|S, \mathcal{M})P(S|\mathcal{M}) \tag{22}$$

La probabilidad de O (dado el modelo) es obtenido sumando su probabilidad conjunta sobre todas las posibles secuencias de estado S_t

$$P(O|\mathcal{M}) = \sum_{S_1,S_2,\ldots,S_T} P(O|S,\mathcal{M})P(S|\mathcal{M}) \quad (23)$$

$$= \sum_{S_1,S_2,\ldots,S_T} P(O,S|\mathcal{M}) \quad (24)$$

donde podemos observar que $P(O)$ puede obtenerse marginalizando sobre la ecuación (24).

Para llevar a cabo la tarea de inferencia es necesario llevar a cabo un proceso de dos pasos: propagación de probabilidades en dirección hacia delante (dirección del tiempo) y propagación hacia atrás (Rabiner, 1989).

1. Para el algoritmo hacia adelante es necesario definir la variable $\alpha_t(i)$ como:

$$\alpha_t(i) = P(O_1 O_2 \ldots O_t, S_t = q_i|\mathcal{M}) \quad (25)$$

es decir, la probabilidad de la secuencia de observaciones parciales, $O_1 \ldots O_t$ (hasta el tiempo t) y el estado q_i dado el modelo $\mathcal{M}$. Podemos obtener $\alpha_t(i)$ de la siguiente forma:

(a) Inicialización

$$\alpha_1(i) = P(O_1|S_1 = q_i)P(S_1 = q_i) \quad (26)$$

(b) Inducción

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)P(S_{t+1} = q_j|S_t = q_i)\right] P(O_{t+1}|S_{t+1} = q_j) \quad (27)$$
$$1 \leq t \leq T-1, \quad 1 \leq j \leq N$$

(c) Terminación

$$P(O|\mathcal{M}) = \sum_{i=1}^{N} \alpha_T(i) \quad (28)$$

donde N es el número de estados de la variable oculta.

Con este algoritmo basta para realizar la tarea de inferencia en una RBD, sin embargo, también puede llevarse a cabo a través del algoritmo hacia atrás el cual define la variable β de la siguiente manera:

2. Algoritmo hacia atrás

$$\beta_t(i) = P(O_{t+1}O_{t+2}\ldots O_T, |S_t = q_i, \mathcal{M}) \tag{29}$$

es decir, la probabilidad de la secuencia de observación parcial desde el tiempo $t+1$ hasta T, dado el estado q_i en el tiempo t y el modelo $\mathcal{M}$. El algoritmo es el siguiente:

(a) Inicialización

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \tag{30}$$

(b) Inducción

$$\begin{aligned} \beta_t(i) &= \sum_{j=1}^{N} P(S_{t+1} = q_j|S_t = q_i)P(O_{t+1}|S_{t+1} = q_j)\beta_{t+1}(j) \\ & t = T-1, T-2, \ldots, 1, \quad 1 \leq i \leq N \end{aligned} \tag{31}$$

II.3.2 Aprendizaje

Un enfoque bayesiano para aprendizaje inicia con algún conocimiento a priori acerca de la estructura del modelo (el conjunto de arcos en la red bayesiana) y parámetros Θ del modelo, para el caso de las RBDs $\Theta = \{P(S_1), P(S_t|S_{t-1}), P(O_t|S_t)\}$. El conocimiento inicial es representado a través de una distribución de probabilidad a priori sobre la estructura y parámetros del modelo y se actualiza usando datos disponibles para obtener una distribución a posteriori sobre el modelo y los parámetros. De manera más formal, dada una distribución de probabilidad sobre la estructura del modelo $P(\mathcal{M})$ y una

distribución sobre los parámetros de cada uno de los modelos $P(\Theta|\mathcal{M})$, un conjunto de observaciones O es utilizado para formar una distribución a posteriori sobre el modelo usando la regla de Bayes (Rabiner, 1989; Avilés-Arriaga, 2006):

$$P(\mathcal{M}|O) = \frac{P(O|\Theta, \mathcal{M})P(\Theta|\mathcal{M})P(\mathcal{M})}{P(O)} \tag{32}$$

Cuando se tienen instancias de todas las variables involucradas en el modelo, el aprendizaje o entrenamiento del modelo suele ser trivial. Sin embargo, en muchos problemas reales no es posible contar con los datos completos siempre, como sucede con las variables ocultas S. De esta forma el problema de aprendizaje se complica. Afortunadamente existe un método para aproximar los parámetros de la ecuación (20). La idea principal de este procedimiento es estimar iterativamente los parámetros Θ del modelo. El objetivo es incrementar en cada paso la verosimilitud $P(S, O|\Theta)$. Las ecuaciones de estimación son las siguientes (Rabiner, 1989):

$$P^*(S_1 = q_i) = \frac{P(S_1 = q_i|O)}{P(O)} \tag{33}$$

donde $P^*(S_1 = q_i)$ se puede ver como el valor esperado de iniciar en el estado q_i en el tiempo $t = 1$,

$$P^*(S_{t+1} = q_j|S_t = q_i) = \frac{\sum_{t=1}^{T-1} P(S_{t+1} = q_j, S_t = q_i|O)}{\sum_{t=1}^{T-1} P(O, S_t = q_i)} \tag{34}$$

donde $P^*(S_{t+1} = q_j|S_t = q_i)$ es el valor esperado de transitar al estado q_j en el tiempo $t + 1$ dado que estuvo en el estado q_i en el tiempo t, y para el caso de una variable aleatoria discreta:

$$P^*(O_t = v_k|S_t = q_i) = \frac{\sum_{t=1}^{T} P(O, S_t = q_i)\delta_{O_t, v_k}}{\sum_{t=1}^{T} P(O, S_t = q_i)} \tag{35}$$

donde $P^*(O_t = v_k|S_t = q_i)$ es el valor esperado de observar v_k en el estado q_i; por tanto, sólo las observaciones v_k aportan para esta probabilidad, la función $\delta_{O_t,v_k} = 1$ si $O_t = v_k$ y $\delta_{O_t,v_k} = 0$ en caso contrario. Después de calcular estas ecuaciones, las estimaciones iniciales $P(\cdot)$ son reemplazadas por $P^*(\cdot)$ si $P^*(O) > P(O)$. Este proceso se repite hasta que $P^*(O) - P(O) < \tau$, donde τ es un umbral, o hasta que se haya alcanzado un número máximo de iteraciones. Podemos entonces seleccionar los parámetros $\Theta = \{P(S_1), P(S_t|S_{t-1}), P(O_t|S_t)\}$ de tal forma que $P(O|\mathcal{M})$ es maximizada localmente utilizando un procedimiento iterativo como el método Baum-Welch (método equivalente a EM (Dempster *et al.*, 1977)) o utilizando técnicas de gradiente (Levinson *et al.*, 1983).

II.4 Resumen

Una RB es un GDA conexo que representa una distribución de probabilidad conjunta, los nodos son variables aleatorias y los enlaces entre ellos representan la relación que existe entre ellas y están cuantificadas por una probabilidad condicional. En una RB existen dos procesos importantes, la inferencia, que consiste en deducir cuál es la distribución sobre un conjunto particular de variables aleatorias dado que conocemos los estados de otras variables en la red (Pavlovic, 1999), y el aprendizaje paramétrico y estructural. El aprendizaje parámetrico consiste en estimar las probabilidades condicionales de la red, el aprendizaje estructural consiste en determinar la topología del GDA conexo por lo que incluye al aprendizaje paramétrico.

Una RBD hace dos suposiciones: que el proceso es estacionario (los parámetros no cambian durante el tiempo) y Markoviano (el futuro es independiente del pasado dado el presente).

El problema de aprendizaje con variables ocultas complica el proceso de aprendizaje. Para el caso de RBDs esto es más costoso aún, ya que hay que considerar el aspecto

dinámico en el proceso de aprendizaje. Para llevar a cabo el proceso de aprendizaje de RBDs se pueden utilizar algoritmos como el algoritmo hacia adelante y el algoritmo hacia atrás. Para el problema de variables ocultas y/o datos incompletos se puede utilizar el algoritmo EM.

En el siguiente capítulo se presenta el uso de RBs como clasificadores, en particular el clasificador bayesiano simple (CBS) y el clasificador bayesiano simple dinámico (CBSD), así como el planteamiento del problema en el aprendizaje de CBSDs.

Capítulo III

Clasificadores bayesianos

III.1 Introducción

La clasificación es una tarea básica en el análisis de datos y reconocimiento de patrones que requiere la construcción de un clasificador, éste es una función que asigna una clase a casos descritos por un conjunto de atributos (Friedman *et al.*, 1997). En un sentido amplio, cualquier método que incorpore información desde muestras de entrenamiento en el diseño de un clasificador utiliza aprendizaje (Duda *et al.*, 2001).

El término clasificación puede cubrir cualquier contexto en donde alguna decisión o pronóstico es hecho sobre una base de información disponible. Un procedimiento de clasificación es entonces un método formal para hacer repetidamente tales juicios en nuevas situaciones (Michie *et al.*, 1994).

El proceso de clasificación puede ser de dos tipos:

- Clasificación supervisada: Se parte de una serie de clases o categorias conceptuales prediseñadas a priori, en la que la labor del clasificador es asignar cada conjunto de atributos (que describen un objeto o fenómeno) a la clase o categoría que le corresponda. La construcción de un procedimiento de clasificación a partir de un conjunto de datos para los cuales las clases verdaderas son conocidas son llamadas también reconocimiento de patrones, discriminación o aprendizaje supervisado.

- Clasificación no supervisada: No hay clases previas de clasificación establecidas a priori. Los objetos, descritos por sus atributos, se agrupan en función de una medida de similitud entre ellos y de la información que representan; las clases son

inferidas a partir de los datos.

Este trabajo se encuentra dentro del primero de los casos, clasificación supervisada.

III.2 Redes bayesianas como clasificadores

Utilizando un método de aprendizaje estructural como el presentado en el Capítulo II podemos inducir una RB B, que codifica una distribución $P(A_1, \ldots, A_n, C)$, desde un conjunto de datos de entrenamiento. Podemos entonces utilizar el modelo obtenido de tal forma que dados un conjunto de atributos $a_1, \ldots, a_n$, el clasificador basado en B regrese la etiqueta c que maximiza la probabilidad posterior $P(c|a_1, \ldots, a_n)$.

Desde esta perspectiva un clasificador puede ser visto como una RB. En la siguiente sección se describen los tipos de clasificadores bayesianos.

III.3 Tipos de clasificadores bayesianos

Un clasificador bayesiano obtiene la probabilidad posterior de cada clase C_i usando la regla de Bayes, como el producto de la probabilidad a priori de la clase por la probabilidad condicional de los atributos dada la clase, dividido por la probabilidad de los atributos

$$P(C|A_1, \ldots, A_n) = \frac{P(C)P(A_1, \ldots, A_n|C)}{P(A_1, \ldots, A_n)} \tag{36}$$

Esta fórmula junto con una suposición de independencia condicional entre atributos da origen a uno de los clasificadores más efectivos, en el sentido de que su rendimiento predictivo es competitivo con el estado del arte de clasificadores. Este clasificador es llamado bayesiano simple, el cual se describe a continuación.

III.3.1 Clasificador bayesiano simple

El clasificador bayesiano simple (CBS) aprende desde datos de entrenamiento la probabilidad condicional de cada atributo A_i dada la clase C. La clasificación es hecha

aplicando la regla de Bayes para calcular la probabilidad de C dada una muestra particular de $A_1, \ldots, A_n$ y entonces predecir la clase con la más alta probabilidad a posteriori. El CBS hace dos suposiciones:

- que los atributos son independientes entre sí, dada la clase,
- que los atributos son discretos.

Un clasificador bayesiano simple es una estructura que tiene un solo nodo (la clase) que es padre de todos los otros nodos (los atributos) [Cheng and Greiner, 1999]. En la Figura 6 se muestra la representación gráfica del clasificador bayesiano simple, donde $A_1, \ldots, A_n$ son los nodos hijos y C es el nodo clase. Debido a la suposición de independencia condicional, los enlaces entre nodos hijo (atributos) no son permitidos.

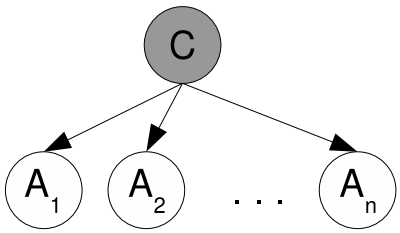

Figura 6. Estructura del clasificador bayesiano simple

La probabilidad conjunta se puede obtener por medio del producto de las probabilidades condicionales individuales de cada atributo dado el nodo clase como se muestra en la siguiente ecuación,

$$P(C, A_1, \ldots, A_n) = P(C) \prod_{i=1}^{n} P(A_i|C) \tag{37}$$

La ecuación (37) muestra de manera clara cómo el clasificador bayesiano simple puede ser interpretado como una RB, ya que muestra la factorización de los nodos hijos dados

sus padres. Aunque la suposición de independencia es generalmente una suposición pobre, en la práctica el CBS a menudo compite con clasificadores más sofisticados.

En un estudio acerca de las características de los datos que afectan el rendimiento del CBS (Rish, 2001), concluyen que este clasificador es sorprendentemente efectivo en la práctica, toda vez que su decisión de clasificación puede ser a menudo correcta aún si sus estimaciones de probabilidad son inexactas.

III.3.2 Clasificador bayesiano simple aumentado a árbol

Con el fin de mejorar el rendimiento del CBS Friedman *et al.* (1997) proponen agregar aristas entre los atributos, manteniendo como en el CBS un enlace entre la clase y los atributos. El clasificador bayesiano simple aumentado a árbol (CBSAA) relaja las suposiciones de independencia condicional, permitiendo a los atributos tener enlaces. En un CBSAA la variable clase no tiene padres y cada atributo tiene como padres a la variable clase y a lo más un atributo, es decir, entre los atributos es permitida una estructura de árbol y cada uno de ellos es influenciado por la variable clase. Una estructura CBSAA se muestra en la Figura 7, donde pueden observarse estas modificaciones. El precio de permitir dependencias, es por supuesto un mayor costo computacional.

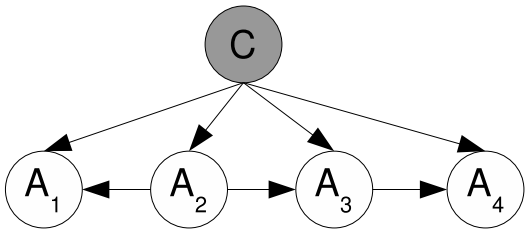

Figura 7. Estructura del clasificador bayesiano aumentado a árbol (CBSAA)

III.3.3 Clasificador bayesiano simple aumentado a red

El clasificador CBSAR generaliza el CBSAA permitiendo que los atributos formen un grafo arbitrario, en vez de un árbol (Cheng y Greiner, 1999). El algoritmo de aprendizaje del CBSAR es igual al algoritmo del CBAA, excepto que CBSAR usa un algoritmo de aprendizaje no restringido en vez de uno restringido para que forme un árbol, como lo hace CBAA (Martínez, 2006). La estructura CBSAR se muestra en la Figura 8, donde puede observarse que los atributos pueden tener más de un padre; es decir, los atributos pueden formar estructuras multiconectadas, pero el nodo clase sigue enlazado a todos los atributos.

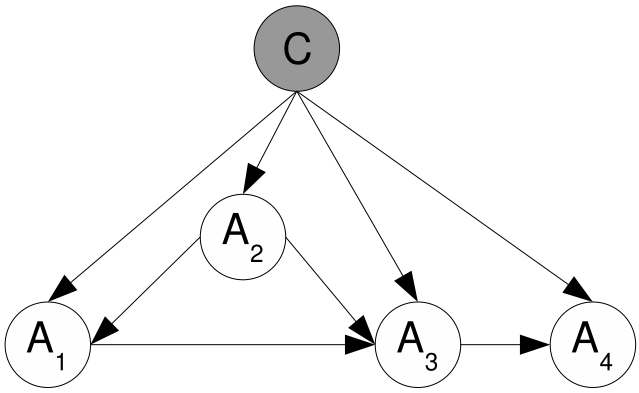

Figura 8. Estructura del clasificador bayesiano aumentado a red (CBSAR)

III.3.4 Clasificador bayesiano general

El algoritmo para aprendizaje de un clasificador bayesiano general (CBG) considera al nodo clasificado como un nodo ordinario, es decir, como se trata el aprendizaje de una RB, el nodo clase no necesariamente está conectado a todos los atributos. La Figura 9 muestra la estructura de este clasificador.

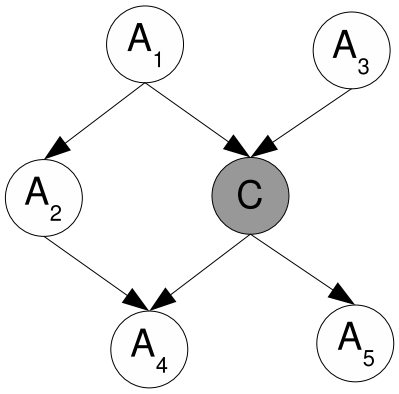

Figura 9. Estructura del clasificador bayesiano general (CBG)

III.3.5 Otros clasificadores bayesianos

Podemos encontrar otras versiones de clasificadores bayesianos, como lo son el clasificador bayesiano k-dependiente (Sahami, 1996) y las multiredes bayesianas (Geiger y Heckerman, 1996). El primer clasificador es una red intermedia entre el CBSAA y CBSAR, ya que permite que cada atributo pueda tener a lo más k atributos como padres, en el caso de las multiredes bayesianas generalizan al CBSAA ya que está conformado por un conjunto de estructuras, una para cada clase. Dependiendo de la clase, se tiene una estructura diferente para la red.

III.4 Problemas en el aprendizaje de clasificadores

En el aprendizaje de una RB, y en particular de un CBS podemos encontrarnos con distintos problemas como son:

1. El manejo de dependencias
2. La existencia de variables ocultas o información incompleta
3. La selección de atributos
4. La discretización

Sucar *et al.* (1993, 1994); Pazzani (1996) proponen una búsqueda de pares de atributos que puedan ser unidos en una misma distribución condicional, buscando con esto una combinación de atributos que permitan obtener mejores resultados de clasificación. Friedman *et al.* (1997) sugieren incluir relaciones de dependencia probabilística entre algunos de los atributos para reflejar correlaciones entre ellos.

Diaz-de-Leon y Sucar (2002) utilizan este último enfoque en una RB derivada del clasificador bayesiano simple para reconocer actividades considerando varias partes del cuerpo. Martínez (2006) realiza un proceso similar al propuesto por Pazzani (1996), el cual realiza aprendizaje estructural que se basa en la unión y/o eliminación de atributos, realizando también un proceso de discretización para poder manejar variables continuas. Avilés-Arriaga (2006) presenta un trabajo para el reconocimiento de ademanes donde se utilizan RBs para detección de piel y para el proceso de clasificación de los ademanes para el control de un robot móvil.

Cuando se tienen instancias de todas las variables involucradas en el modelo, el aprendizaje o entrenamiento del modelo suele ser trivial. En procesos del mundo real, generalmente se pueden identificar sus salidas. Sin embargo el origen de dichas salidas es desconocido. En una base de datos a esta falta de la información se le conoce como variable oculta, es decir, una variable para la cual no se ha tenido lectura de ninguno de sus valores tomados en cada una de las muestras de los datos. La existencia de este tipo de variables requiere que se utilicen métodos más costosos para obtener los parámetros cuando el proceso es modelado.

La mayoría de trabajos existentes para aprendizaje de nodos ocultos se encuentran enfocados al área de aprendizaje paramétrico por lo que la estructura es fija. (Kwoh y Gillies, 1996) proponen la creación de nodos ocultos, los cuales modelan las dependencias entre variables. Para la obtención de la matriz de probabilidades de esos nodos se utiliza un método de gradiente descendiente.

Como ya se mencionó el algoritmo EM permite obtener los parámetros de una red en la cual existen variables ocultas o datos faltantes. Este algoritmo es utilizado por Friedman (1998) para aprendizaje estructural. EL algoritmo realiza los dos pasos de EM, en el primero de ellos completa los datos faltantes por medio de los valores esperados de las variables basado en la estructura y parámetros actuales. En el segundo paso, calcula en base a la estructura actual la calidad de una estructura candidata, es decir, el algoritmo completa los datos utilizando la red actual.

III.5 Clasificador bayesiano simple dinámico

El clasificador bayesiano simple dinámico (CBSD) parte de las mismas suposiciones de una RBD y del CBS, es decir supone que los procesos son estacionarios (los parámetros no cambian con el tiempo) y Markovianos (la probabilidad del estado futuro es independiente del pasado dado el presente), además de que supone independencia condicional de los atributos dada la clase. El CBSD está compuesto por (Avilés-Arriaga, 2006):

- Un conjunto finito $C = \{C[t]|t = 1, \ldots, T\}$, donde cada $C[t]$ es una variable aleatoria que puede tomar su valor de una de las N clases posibles $U = \{c_1, \ldots, c_N\}$ en el tiempo t y,

- un conjunto $A = \{A[t]|t = 1, \ldots, T\}$, donde cada $A[t] = \{A_1[t], \ldots, A_M[t]\}$, de M variables aleatorias que corresponden a los atributos de interés observados del proceso en un tiempo t.

Un CSBD tiene la siguiente función general de probabilidad conjunta:

$$P(A, C) = P(C[1]) \prod_{t=1}^{T} \prod_{m=1}^{M} P(A_m[t]|C[t]) \prod_{t=1}^{T-1} P(C[t+1]|C[t]) \tag{38}$$

donde $P(C[1])$ es la distribución de probabilidad a priori para la variable de clase $C[1]$, $P(A_m[t]|C[t])$ es la función de probabilidad de un atributo dada la clase en el tiempo t

y $P(C[t+1]|C[t]|)$ es la distribución de probabilidad de la transición de estados entre las variables de clase a través del tiempo. El término $\prod_{m=1}^{M} P(A_m[t]|C[t])$ muestra las suposiciones de independencia condicional entre los atributos dada la clase, como se suponen en el clasificador bayesiano simple. La Figura 10 muestra la estructura del CBSD.

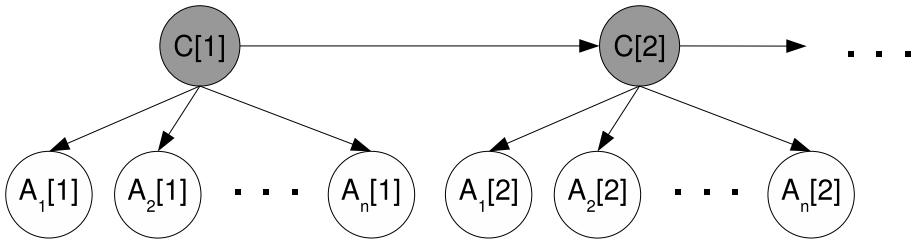

Figura 10. Estructura del clasificador bayesiano simple dinámico (CBSD)

III.5.1 Inferencia

De las figuras 5 y 10 se observa que las estructuras son muy similares. Recordemos también que un Modelo Oculto de Markov (MOM) puede ser visto, al igual que un CBSD, como una RBD, por lo que los algoritmos de inferencia pueden derivarse directamente de los algoritmos Hacia-Adelante y Hacia-Atrás de los MOMs.

Algoritmo hacia adelante

Para el algoritmo hacia adelante es necesario definir la variable $\alpha_t(i)$ como:

$$\alpha_t(i) = P(A[1], \ldots, A[t], C[t] = c_i|\mathcal{M}) \tag{39}$$

Podemos obtener $\alpha_t(i)$ de la siguiente forma:

1. Inicialización

$$\alpha_1(i) = P(C[1] = c_i) \prod_{m=1}^{M} P(A_m[1]|C[1] = c_i) \tag{40}$$

2. Inducción

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) P(C[t+1] = c_j | C[t] = c_i)\right] \prod_{m=1}^{M} P(A_m[t+1] | C[t+1] = c_j) \tag{41}$$

$$1 \leq t \leq T-1, \quad 1 \leq j \leq N$$

3. Terminación

$$P(A|\mathcal{M}) = \sum_{i=1}^{N} \alpha_T(i) \tag{42}$$

donde M es el número de atributos y N el número de estados.

Con este algoritmo basta para realizar la tarea de inferencia en el CBSD, sin embargo, también puede llevarse a cabo a través del algoritmo hacia atrás el cual define la variable β como se muestra a continuación.

Algoritmo hacia atrás

El algoritmo hacia atrás define la variable

$$\beta_t(i) = P(A[t+1], \ldots, A[T], | C[t] = c_i, \mathcal{M}) \tag{43}$$

es decir, calcula la probabilidad de la secuencia de observación parcial desde el tiempo $t+1$ hasta T, dada la clase c_i en el tiempo t y el modelo $\mathcal{M}$. El algoritmo es el siguiente:

1. Inicialización

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \tag{44}$$

2. Inducción

$$\beta_t(i) = \sum_{j=1}^{N} P(C[t+1] = c_j | C[t] = c_i) \prod_{m=1}^{M} P(A[t+1] | C[t+1] = c_j) \beta_{t+1}(j) \tag{45}$$

$$t = T-1, T-2, \ldots, 1, \quad 1 \leq i \leq N$$

El proceso de clasificación con estos modelos es el mismo que para los modelos ocultos de Markov.

III.5.2 Aprendizaje

III.5.2.1 Aprendizaje paramétrico

En el proceso de aprendizaje de un CBSD se busca determinar cuales son los parámetros del nodo clase oculto en $t = 1$ ($P(C[1])$), las probabilidades de transición ($P(C[t]|C[t+1])$) y las probabilidades de los atributos dada la clase ($P(A_i[t]|C[t])$), por lo que podemos utilizar el método presentado en el Capítulo II para RBDs, donde podemos observar que sólo es necesario aplicar en la ecuación (35) la suposición de independencia de los atributos, por lo que tendríamos la siguiente expresión:

$$P^*(A[t] = v_k|C[t] = c_i) = \frac{\sum_{t=1}^{T} P(A, C[t] = c_i)\delta_{A[t],v_k}}{\sum_{t=1}^{T} P(A, C[t] = c_i)} \tag{46}$$

III.5.2.2 Aprendizaje estructural

Comúnmente no se realiza aprendizaje estructural en el CBSD, ya que se considera una estructura restringida; es decir, siempre hay un solo nodo padre (la clase) y un conjunto de nodos hijo (los atributos), por lo que la estructura permanece sin cambios en el proceso de aprendizaje. Sin embargo, en este trabajo se siguen los enfoques propuestos por Sucar *et al.* (1993, 1994) y por Pazzani (1996).

Sucar *et al.* (1993) proponen la metodología QUALQUANT (Orientación Cualitativa Cuantitativa), la cual busca construir y mejorar la estructura de una RB. Se obtiene un conjunto de reglas basadas en el conocimiento del experto, a partir de las cuales se construye la estructura inicial de la red. Posteriormente, basado en técnicas estadísticas, mejora la estructura inicial de la red. Este mecanismo utiliza un conjunto de datos para validación de dependencias, para lo cual se calcula la correlación entre variables. Si se encuentra una correlación baja no implica necesariamente independencia, pero sí indica que se puede suponer independencia. En cambio, si la correlación es alta, indica que no

son independientes.

Pazzani (1996) presenta un método basado en ajustes donde identifica dos desventajas del CBS: que los atributos contenidos en un conjunto de datos pueden no ser condicionalmente independientes dada la clase, o bien que uno o más atributos irrelevantes pueden tener algún grado de dependencia de atributos relevantes y que ésto puede afectar la exactitud del clasificador. Sin embargo, Pazzani (1996) indica que dichos problemas (dependencias y datos irrelevantes) pueden ser detectados a partir de los datos. Por ejemplo, en la Figura 11a se tiene un dominio de 4 variables A_1, A_2, A_3 y A_4 y una variable a predecir C, donde se supone que la variable A_2 no es relevante para C, y que las variables A_1 y A_3 son condicionalmente dependientes dado C. Por lo que el modelo quedaría tal como se observa en la Figura 11b.

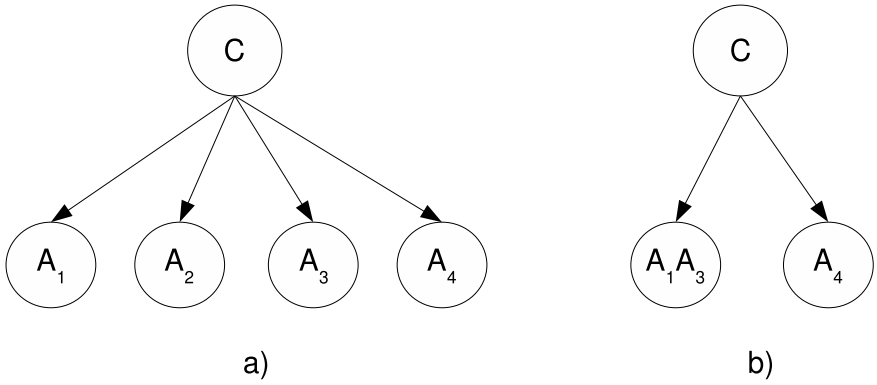

Figura 11. a) Estructura inicial del CBS. b) Estructura modificada después de la eliminación del atributo A_2 y la unión de los atributos A_1, A_3.

Martínez (2006) propone dos métodos, uno para el aprendizaje de clasificadores bayesianos estáticos (ACBE) y otro para el aprendizaje de clasificadores bayesianos dinámicos (ACBD).

El método ACBE incluye cuatro etapas, i) Inicialización, ii) Discretización, iii) Mejora estructural y iv) Clasificación. Las etapas de discretización y mejora estructural

se repiten hasta que la exactitud de la clasificación no puede ser mejorada. La discretización se basa en el principio de Longitud de Descripción Mínima (LDM) (Lam y Bacchus, 1994), donde el número de intervalos que minimiza la LDM se obtiene para cada atributo. Para tratar con atributos dependientes y atributos irrelevantes elimina y/o une atributos, basado en medidas de información mutua condicional y evaluando la exactitud de la clasificación después de cada operación.

El método ACBD incluye cinco etapas, i) Inicialización, ii) Discretización, iii) Determinación del nodo clase oculto, iv) Mejora estructural y v) Clasificación dinámica. El método de discretización es similar al del método ACBE; la evaluación de las estructuras en la fase de mejora estructural considera que la red tiene una estructura de árbol y son evaluadas a través de una medida de calidad basada en el principio LDM, la determinación del mejor número de estados para el nodo clase oculto se basa en el algoritmo EM y las estructuras resultantes se evalúan con base a la medida de calidad. Finalmente la Clasificación dinámica es evaluada.

III.5.2.3 El problema de aprendizaje del CBSD

Siguiendo los enfoques propuestos por Sucar *et al.* (1993, 1994) y Pazzani (1996) podemos identificar dependencias entre atributos, por lo que cada uno de los nodos hijo del CBSD pueden ser vistos como grupos conteniendo uno o más atributos. La estructura podría representarse entonces como se muestra en la Figura 12. Entonces, el problema es decidir cual agrupamiento G utilizar y el número de estados en el nodo oculto, una vez que el óptimo será el agrupamiento de atributos dependientes y la eliminación de atributos irrelevantes. Si utilizamos fuerza bruta para determinar esta estructura óptima (agrupamiento), en un problema con n atributos, entonces se requiere buscar en un espacio de solución de tamaño dado por la siguiente ecuación:

$$|\mathcal{G}'_n| = \left(\sum_{i=1}^{n} \begin{pmatrix} n \\ n-i \end{pmatrix} B_i \right) * n_s \tag{47}$$

donde B_i es el número de Bell de i elementos (Cameron, 1994) y n_s es el número de estados en el nodo clase oculto. Por ejemplo, para una red con siete atributos y ocho estados para el nodo oculto, tenemos que

$$|\mathcal{G}'_7| = \left(\sum_{i=1}^{7} \begin{pmatrix} 7 \\ 7-i \end{pmatrix} B_7\right) * 8 = 28973 \quad (48)$$

con el mismo número de estados para la variable oculta, pero con ocho atributos, tenemos que $|\mathcal{G}'_8| = 119042$, con nueve atributos $|\mathcal{G}'_9| = 402969$. Recordemos que este conteo es para un solo modelo, pero un clasificador completo se encuentra conformado por un número de modelos igual al número de categorias a clasificar. No es difícil ver que $|\mathcal{G}'_n|$ crece exponencialmente con n. Por lo tanto, no podemos explorar exhaustivamente el espacio de solución, aún para un número pequeño de atributos, por lo que se requiere una alternativa a la fuerza bruta para encontrar un agrupamiento G óptimo o cercano al óptimo.

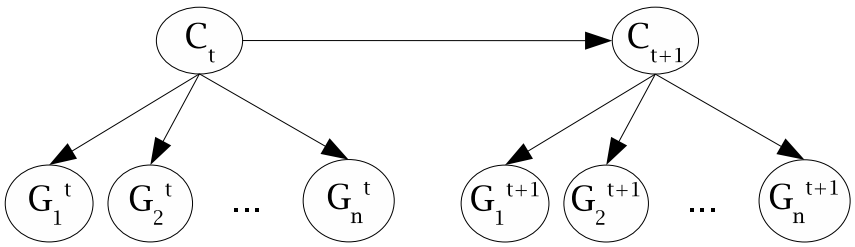

Figura 12. Estructura del CBSD donde cada nodo G_i^t corresponde a un agrupamiento de atributos

III.6 Resumen

El CBS ha sido utilizado ampliamente debido a que es un método de clasificación eficiente, fácil de aprender y con gran exactitud en muchos dominios. El CBS ofrece una semántica simple y clara de los atributos y su relación con respecto a la clase. Aunque

la suposición de independencia es casi siempre violada en la práctica, el CBS da buenos resultados.

Actualmente existe una tendencia muy importante en el estudio de los clasificadores bayesianos, tanto en la mejora de la estructura relajando las suposiciones de independencia, como en la identificación de las métricas adecuadas para evaluar el aprendizaje. Buscando la mejora de la estructura en el CBS se han propuesto distintas opciones como por ejemplo el CBSAA, CBSAR, CBSG y otros más.

Una extensión al CBS es el CBSD el cual mantiene las mismas suposiciones pero modela procesos dinámicos, lo cual permite verlo como una RBD, en particular como un MOM. Algoritmos de aprendizaje para este tipo de modelos pueden ser utilizados entonces para aprender el CBSD.

Siguiendo los enfoques propuestos por Sucar *et al.* (1993, 1994) y Pazzani (1996) podemos identificar dependencias entre atributos, por lo que cada uno de los nodos hijo del CBSD pueden ser vistos como grupos conteniendo uno o más atributos. Sin embargo este problema crece exponencialmente con el número de atributos. Por lo tanto, no podemos explorar exhaustivamente el espacio de solución, aún para un número pequeño de atributos, por lo que se requiere una alternativa a la fuerza bruta para encontrar un agrupamiento G óptimo o cercano al óptimo. Además el manejo de información incompleta complica el problema de aprendizaje. Algoritmos como EM (Expectation-Maximization) (Dempster *et al.*, 1977) son necesarios para estimar los datos faltantes. Lo que provoca un espacio de búsqueda grande y multimodal (Friedman, 1998). Algoritmos determinísticos son propensos a obtener óptimos locales.

Una opción para enfrentar el problema con óptimos locales, es utilizar un método de búsqueda estocástico. Este trabajo explora el uso de cómputo evolutivo, en particular de Algoritmos Genéticos (AGs), para el aprendizaje del CBSD. En el siguiente Capítulo se proporcionan los conceptos básicos que conforman la base del algoritmo propuesto.

Capítulo IV

Enfoque evolutivo

IV.1 Introducción

El cómputo evolutivo es un área de investigación dentro de ciencias de la computación, inspirada en el proceso de evolución natural. Esta área incluye algoritmos genéticos, desarrollados por Holland (1975); estrategias evolutivas, desarrolladas por Rechenberg (1973) y por Schwefel (1993); programación evolutiva, desarrollada por Fogel *et al.* (1966); y programación genética, desarrollada por Koza (1992).

La metáfora fundamental del cómputo evolutivo relaciona el principio de la supervivencia del más apto en la evolución natural a un estilo particular de resolver problemas. Esta relación puede observarse en la Tabla I.

Tabla I. Metáfora básica del cómputo evolutivo enlazando la evolución natural a la solución de problemas

Evolución		Solución del problema
Entorno	$\longleftrightarrow$	Problema
Individuo	$\longleftrightarrow$	Solución candidata
Aptitud	$\longleftrightarrow$	Calidad

IV.2 Algoritmos genéticos

Los algoritmos genéticos (AG) han demostrado ser un enfoque general y flexible que se adapta a cualquier problema de búsqueda y optimización. Una de las principales diferencias con respecto a los métodos clásicos, es que los AGs utilizan una población de soluciones por iteración, en lugar de una sola solución. Para mover la población de

soluciones sobre el espacio de búsqueda, los AGs se basan en el principio de selección natural, donde los más aptos sobreviven y prosperan.

En un organismo biológico, la estructura que codifica la información especificando cómo el organismo será construido es llamado cromosoma. Uno o más cromosomas pueden ser requeridos para especificar el organismo completo. El conjunto completo de cromosomas es llamado un genotipo, y las características físicas resultantes se denominan fenotipo. Cada cromosoma comprende un número de estructuras individuales llamadas genes. Cada gen codifica una característica particular del organismo, y la localización, o locus, del gen dentro de la estructura del cromosoma determina qué características particulares representa el gen. En un locus particular, un gen puede codificar diferentes valores de la característica particular que representa. Los diferentes valores de un gen que pueden ocurrir en un locus particular son llamados alelos. Esta terminología se encuentra resumida en la Tabla II.

Tabla II. Explicación de términos en algoritmos genéticos (Michalewicz, 1996).

Algoritmos genéticos	Explicación
Cromosoma (cadena, individuo)	Solución (codificación)
Genes	Parte de la solución
Locus	Posición del gen
Alelos	Valores del gen
Fenotipo	Solución decodificada
Genotipo	Solución codificada

A pesar de que existen diferentes técnicas de algoritmos evolutivos, todas ellas comparten los siguientes rasgos fundamentales (Michalewicz, 1996):

- Una codificación de soluciones para el problema (genotipo). Un mapeo del espacio de soluciones (fenotipo) al espacio de codificación (genotipo).
- Una función de decodificación del genotipo al fenotipo.

- Una forma para crear una población inicial de soluciones.
- Operadores genéticos que alteren la composición genética de los hijos durante la reproducción.
- Un criterio de evaluación, mediante el cual, diferentes soluciones se comparan de un modo objetivo, o subjetivo según convenga.
- Un mecanismo de selección por el cual ciertas soluciones son destinadas a ser parte de la siguiente generación de individuos y a iniciar un nuevo ciclo en el proceso evolutivo.

IV.3 Estructura general de un algoritmo genético

Un algoritmo genético mantiene una **población** de individuos, es decir F(t), para la generación t. Cada **individuo** representa una solución potencial al problema. Cada individuo es evaluado para dar alguna medida de su **aptitud**. Algunos individuos sufren transformaciones estocásticas por medio de **operaciones genéticas** para formar nuevos individuos.

Hay dos tipos de transformación: **mutación**, la cual crea nuevos individuos haciendo cambios en un solo individuo, y **cruzamiento**, el cual crea nuevos individuos combinando partes de dos o más individuos. Los nuevos individuos, llamados **descendientes** D(t), son entonces evaluados. Una nueva población es formada seleccionando los individuos más aptos de la población de padres y la población de descendientes. Después de varias generaciones, el algoritmo converge al mejor individuo, el cual se espera represente una solución óptima (o cercana a una) para el problema. El esquema general de un algoritmo evolutivo puede verse en la Figura 13.

Debido a que el espacio de búsqueda de una RB es grande, multidimensional y multimodal (Friedman, 1998), el enfoque evolutivo parece ser una técnica apropiada para

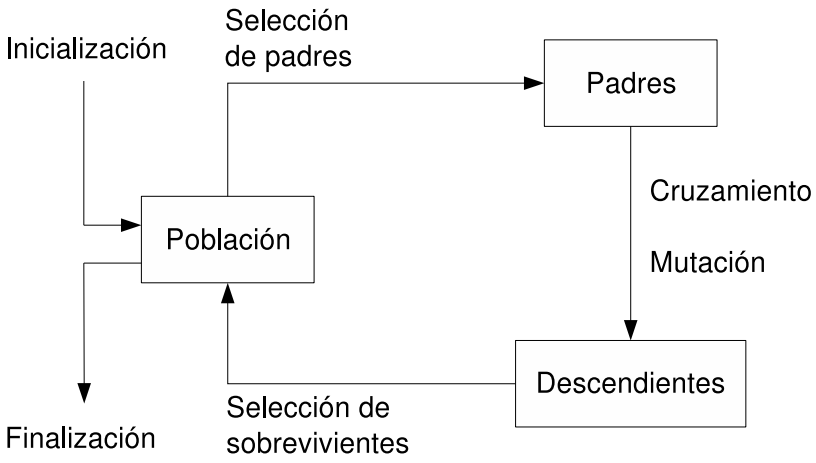

Figura 13. El esquema general de un algoritmo evolutivo como un digrama de flujo, esquema tomado de Eiben y Smith (2003)

resolver el problema de aprendizaje, por lo que podemos considerar la misma situación para el aprendizaje del CBSD, ya que para este problema se realiza la búsqueda de atributos dependientes para fusionarlos en una variable y la eliminación de atributos no relevantes, además de la búsqueda del mejor número de estados para la variable oculta.

En trabajos previos el cómputo evolutivo ha sido utilizado para el aprendizaje de RBs generales. (Larrañaga y Poza, 1996) proponen utilizar algoritmos genéticos para buscar la estructura óptima de una RB. En su búsqueda la estructura de la red es representada por una matriz de adyacencia M de $n \times n$ donde n es el número de variables. El contenido la matriz es binario, la presencia de un uno en la posición M_{ij} indicaba que el nodo j es un padre del nodo i. Para la función de aptitud utilizó una medida Bayesiana.

Myers *et al.* (1999) realizan aprendizaje con datos incompletos para problemas de clasificación. Este enfoque evoluciona tanto los datos como la estructura de la red. El cromosoma para los datos consiste en una cadena de los datos perdidos, donde los genes toman valores muestreados desde el conjunto de valores de la variable correspondiente. La estructura es representada como una lista de adyacencia que contiene los nodos padre

de un nodo en particular.

Wong *et al.* (2002) presentan un enfoque híbrido, donde se realizan análisis de dependencias para reducir el espacio de búsqueda. Con tal reducción el proceso de búsqueda toma menos tiempo para encontrar la solución óptima. La métrica utilizada es el principio de Longitud de Descripción Mínima (Lam y Bacchus, 1994)

Ross y Zuviria (2007) realizan optimización multiobjetivo para el aprendizaje de RBD's, dadas secuencias de datos multivariados se busca sintetizar una red que modele las relaciones causales que explican las secuencias, los criterios de optimización son una medida probabilística de la red, así como una medida de complejidad. El grado de conectividad es controlado con un número máximo de padres por nodo.

IV.4 Representación de individuos

Cómo codificar una solución del problema es un asunto clave cuando se utilizan algoritmos genéticos. Los objetos que forman parte de posibles soluciones dentro del contexto del problema original son referidos como **fenotipos**, mientras que su codificación, esto es los individuos dentro del AG, forman parte del **genotipo**. El primer paso de diseño consiste en determinar la representación genética de una solución candidata al problema. Una solución (un buen fenotipo) es obtenido decodificando el mejor genotipo después de que finaliza el algoritmo.

De acuerdo con el tipo de símbolo utilizado para representar al fenotipo, los métodos de codificación pueden ser clasificados como sigue:

- Codificación binaria. La codificación es llevada a cabo utilizando cadenas binarias (Holland, 1975).

- Codificación de números-reales. La representación binaria tiene problemas cuando existen un número grande de variables o cuando el intervalo de las variables es

muy amplio. En el caso de que se requiera representar números reales se tiene el problema de la precisión deseada. La codificación de números reales es utilizada para problemas de optimización de funciones debido a que la estructura topológica del espacio genotípico de codificación es idéntico al espacio fenotípico (Michalewicz, 1996).

- Codificación de permutación entera. Es utilizada para problemas de optimización combinatoria debido a que la esencia de éstos problemas es la búsqueda de un objeto combinatorio como permutaciones y grafos, o también combinación de datos sujetos a restricciones (Gen y Cheng, 2000).

- Codificación de estructura de datos general. Es utilizada para problemas más complejos, para capturar la naturaleza del problema. Un gen puede ser una estructura de datos compleja, como por ejemplo los pasos a seguir en procesos de diseño de algún sistema en particular.

Estas son codificaciones estándares; sin embargo, pueden ser utilizadas, otros tipos de codificación dependiendo del problema que se desea resolver. Debido a que en este trabajo se desea identificar atributos dependientes e independientes, así como atributos irrelevantes, se requiere una representación que permita identificar a qué grupo pertenece cada atributo. Por esto se utiliza la representación basada en grupos propuesta por Falkenauer (1994) para el problema de agrupamiento de cajas. En la siguiente sección se describe de manera general dicha representación.

IV.4.1 Representación basada en grupos.

Este tipo de representación fue propuesta por Falkenauer (1994) para el problema de agrupamiento de cajas. Donde la aptitud de un individuo depende del agrupamiento de los objetos en las cajas. Esta representación está formada por dos partes: parte objeto

y parte grupo. La parte objeto representa la pertenencia a grupos, es decir qué objetos pertenecen a qué grupos. La parte grupo representa sólo los grupos. Por ejemplo, si la parte objeto es

$$ADBCDB$$

la parte grupo del cromosoma puede ser representado como

$$ADBC$$

Como ya se mencionó, la parte objeto representa la pertenencia, entonces en el ejemplo anterior el primer objeto está en el grupo A, el segundo y quinto en el grupo D, el tercero y sexto en B, y finalmente el cuarto en C. El punto importante de este tipo de representación es que los operadores genéticos trabajan con la parte grupo del cromosoma, la parte objeto solo es necesaria para identificar qué objetos están contenidos en cada caja. De esta forma el cromosoma es de longitud variable.

IV.5 La función de aptitud

El rol de la función de evaluación o aptitud es representar los requerimientos de adaptación, forma la base para la selección, y por lo tanto facilita las mejoras en la población (Eiben y Smith, 2003). Desde la perspectiva de la solución del problema, representa la tarea a resolver o, más específicamente, la función a optimizar en el contexto evolutivo. Técnicamente, es una función o procedimiento que asigna una medida de calidad a genotipos para cuyo cálculo se usa sólo el fenotipo. Es diseñada específicamente para cada problema y tiene la capacidad de asignar a un individuo un valor de aptitud que indica qué tan bueno o malo es como solución del problema.

IV.6 Selección de individuos

El principio detrás de los AGs es esencialmente el de selección natural neo-darwiniana. La selección proporciona la fuerza motriz en un AG. Con demasiada presión de selección, la búsqueda genética terminará prematuramente, con poca presión se volverá más lenta de lo necesario.

En AGs existen dos mecanismos de selección, la selección de padres y la selección de sobrevivientes o de reemplazo. Ambos son responsables de controlar la razón de mejora en la calidad de los individuos.

- **Selección de padres**. Distingue entre individuos en base a su calidad, en particular, para permitir que los mejores individuos se conviertan en padres de la siguiente generación. Un individuo es un padre si ha sido seleccionado para sufrir variaciones a fin de crear descendientes

- **Selección de sobrevivientes**. Es similar a la selección de padres, pero se utiliza en un escenario diferente del ciclo evolutivo. Después de haber creado los descendientes, permite decidir, de entre padres y descendientes, cuáles individuos permanecerán en la siguiente generación.

IV.7 Operadores genéticos

En esencia, los operadores genéticos llevan a cabo una búsqueda aleatoria y no pueden garantizar producir mejoras en los descendientes. Sin embargo, son capaces de hacer una mejor exploración del espacio de búsqueda y pueden escapar de óptimos locales. Generalmente son dos los operadores genéticos que se aplican: cruzamiento y mutación.

IV.7.1 Cruzamiento

El operador de cruzamiento realiza un intercambio de información entre dos o más individuos, produciendo uno o dos descendientes. La idea principal es fusionar información de ambos padres con el fin de obtener descendientes que sean buenas soluciones durante el proceso evolutivo; esto se aprovecha para realizar una búsqueda minuciosa en los espacios entre ellos. El cruzamiento procura agotar la búsqueda en regiones con mayor valor de aptitud, en sucesivas generaciones de selección y cruzamiento.

IV.7.1.1 Cruzamiento para la representación de grupos

La representación propuesta por Falkenauer (1994) está basada en grupos y trabaja con cromosomas de longitud variable, con genes representando los grupos. El procedimiento de cruzamiento es el siguiente (ver Figura 14):

1. Seleccionar aleatoriamente 2 puntos de cruzamiento, delimitando la sección de cruce en cada uno de los padres (Figura 14.a).

2. Insertar el contenido de la sección de cruce del primer padre al primer lugar de cruzamiento del segundo padre (Figura 14.b).

3. Eliminar todos los objetos repetidos en los grupos donde ellos son miembros en el segundo padre, por lo que los viejos miembros de estos grupos dan forma al miembro especificado por los nuevos grupos insertados (Figura 14.c).

4. Si es necesario, adaptar los grupos resultantes, de acuerdo a las restricciones y la función de costo a ser optimizadas (Figura 14.d).

5. Aplicar pasos de 2 a 4 a los dos padres con sus roles invertidos para generar el segundo hijo.

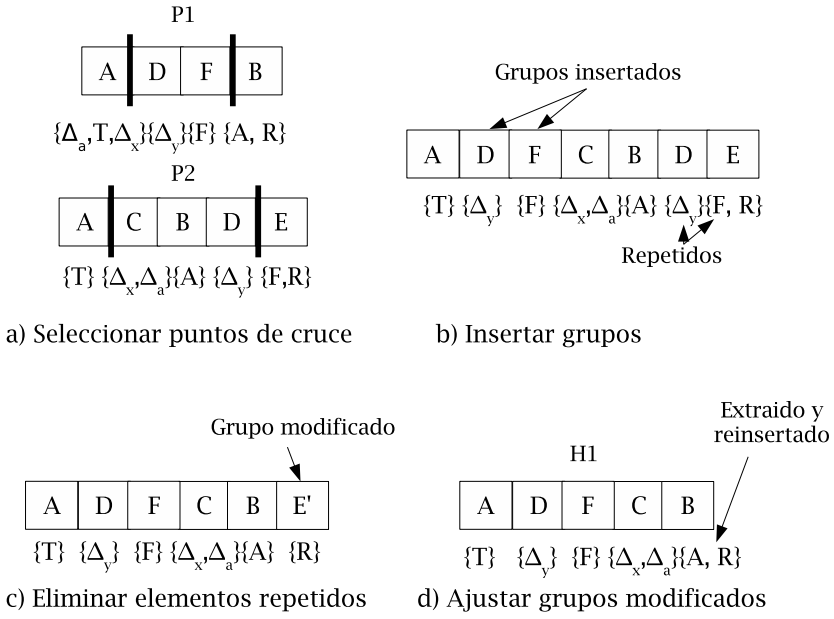

Figura 14. Cruzamiento propuesto por Falkenauer (1994) para la representación basada en grupos. Se seleccionan dos puntos de cruce en ambos padres. El contenido de la sección de cruce se inserta en el inicio de la sección de cruce del segundo padre. Se eliminan los datos repetidos y si es necesario se adaptan los grupos resultantes. El proceso se repite con los roles de los padres invertido

IV.7.2 Mutación

La mutación inserta en un individuo cambios al azar para crear uno nuevo. Ayuda a mantener la diversidad en la población, por lo que se le considera un operador de exploración del espacio de búsqueda.

IV.8 Criterio de paro

Si el problema tiene un nivel de aptitud óptimo conocido, el alcanzar este nivel (tal vez con una precisión dada $\epsilon > 0$) debe ser utilizado como un criterio de paro (Eiben y Smith, 2003). Sin embargo, los AGs no garantizan alcanzar un óptimo, de aquí que esta condición podría no satisfacerse. Esto requiere que el criterio de paro sea extendido con

otros requerimientos, como por ejemplo:

- El máximo tiempo de CPU permitido.
- El número total de evaluaciones de aptitud alcanzadas.
- El número de generaciones en que la mejora de la aptitud permanece bajo un umbral.
- La disminución de la diversidad de la población bajo un umbral dado.

IV.9 Resumen

La metáfora fundamental del cómputo evolutivo relaciona la teoría de la sobrevivencia del más apto a un estilo particular de resolver problemas, el de prueba y error.

Los algoritmos genéticos (AG) han mostrado ser un enfoque general y flexible que se adapta a cualquier problema de búsqueda y optimización. Un algoritmo genético mantiene una población de individuos. Cada individuo representa una solución potencial al problema. Cada individuo es evaluado para dar alguna medida de su aptitud. Algunos individuos sufren transformaciones estocásticas por medio de operaciones genéticas para formar nuevos individuos. Hay dos tipos de transformación: mutación, la cual crea nuevos individuos haciendo cambios en un solo individuo, y cruzamiento, el cual crea nuevos individuos combinando partes de dos o más individuos denominados padres. Los nuevos individuos, llamados descendientes o hijos, son entonces evaluados. Una nueva población se forma seleccionando los individuos más aptos de la población de padres y de la de descendientes. Después de varias generaciones, el algoritmo converge al mejor individuo, el cual se espera represente una solución óptima o subóptima de buena calidad para el problema.

En el siguiente capítulo se describe el algoritmo propuesto y cada uno de sus componentes.

Capítulo V

Aprendizaje evolutivo del clasificador bayesiano simple dinámico

V.1 Introducción

Cuando se tienen datos observados parcialmente o cuando no se conoce el nodo a predecir (nodo clase) nos encontramos con que en muchos casos la determinación del mejor número de estados, así como la asociación de atributos más convenientes se encuentran dados por el experto, con base en su experiencia (Avilés-Arriaga, 2006). En este trabajo se propone un método capaz de obtener lo siguiente:

- La estructura del CBSD.
- Los parámetros asociados a la estructura resultante.
- Un número de estados optimizado para el nodo clase oculto
- Selección de los mejores atributos para cada modelo.

El método propuesto busca identificar variables dependientes o irrelevantes, determinando al mismo tiempo el número de estados de la variable oculta, con el fin de incrementar la exactitud del CBSD y mantener la simplicidad del clasificador. Este proceso es realizado con un enfoque evolutivo.

Para tratar con atributos irrelevantes y atributos dependientes se aplica un método de mejora estructural que elimina o une atributos durante el proceso evolutivo por

medio de los operadores genéticos de cruzamiento y mutación. Para la obtención de los valores no observados de la variable oculta se utiliza el algoritmo EM. Cada uno de los individuos es evaluado con la métrica propuesta en éste trabajo (ecuación (49)). Este proceso se realiza hasta que un número de iteraciones es alcanzado o se cumpla un criterio de paro.

V.2 El algoritmo

El algoritmo principal es descrito en el Algoritmo 1. En el paso 1 se crea en forma aleatoria una población inicial y los parámetros correspondientes son obtenidos. El paso 2 calcula la aptitud de cada uno de los individuos generados basado en un conjunto de datos de prueba parcial. En el paso 3 se inicia un ciclo (ver Figura 15), que finaliza cuando ha transcurrido un número máximo de iteraciones sin cambios en la aptitud del mejor individuo, o bien cuando un número máximo de iteraciones es alcanzado.

Los parámetros de entrada para el algoritmo son:

- Un conjunto de datos de los cuales se debe indicar que porcentaje de éstos serán utilizados para la obtención de parámetros (P_e), el porcentaje para prueba parcial (P_{p1}) y el porcentaje para prueba final (P_{p2}) una vez que se obtenga el mejor clasificador.

- Se debe proporcionar el número de clases que se reconocerán y un factor de ponderación α que permite establecer un compromiso entre la complejidad de la red y su exactitud.

- Un número máximo de estados para el nodo clase oculto de acuerdo al proceso modelado.

- El tamaño de la población, porcentaje de mutación (P_m) y el tamaño del torneo (J) en la selección de padres.

Algoritmo 1 EvoDNBC

Entrada: Datos (D), número de clases (n_clases), número máximo de estados (e_{MAX}), porcentaje de entrenamiento (P_e), porcentaje de datos de prueba parcial (P_{p1}), factor de ponderación (α), máximo número de iteraciones (Max_Iter), porcentaje de mutación (P_m), tamaño de la población ($PobSize$), tamaño del torneo (J), máximo número de iteraciones sin cambios en la aptitud del mejor individuo ($Max\_Iter\_sin\_cambio$).
Salida: Un CBSD y su correspondiente aptitud.

```
1 Inicializar PobSize individuos con n_clases modelos
  aleatorios cada uno.
2 Evaluar la aptitud de los PopSize individuos.
3 Mientras el número de generaciones sin cambio en la aptitud
  del mejor individuo es menor que Max_Iter_sin_cambio y el
  número de generaciones es menor que Max_Iter, Hacer:
4    Para i=1 a el piso de PobSize/2
5      Seleccionar J individuos para participar en un torneo y
       el ganador será el Padre1.
6      Seleccionar J individuos para participar en un torneo y
       el ganador será el Padre2.
7      Cruzar Padre1 y Padre2 para obtener dos nuevos
       individuos.
8    Aplicar mutación a cada nuevo individuo con probabilidad
     P_m.
9    Evaluar la aptitud de la nueva población.
10   Reemplazar la población actual con los mejores PopSize
     individuos entre los padres e hijos de la población.
11 Fin mientras
12 Evaluar el porcentaje de clasificación del mejor individuo
   de la última generación.
```

- Para indicar el término de la búsqueda se requieren dos datos, el máximo número de iteraciones y el máximo número de iteraciones sin cambio en la aptitud del mejor individuo.

Como se puede observar el algoritmo puede dividirse en dos bloques, la inicialización (paso 1) y el ciclo evolutivo (pasos 3-11 ver Figura 15 para un diagrama de bloques).

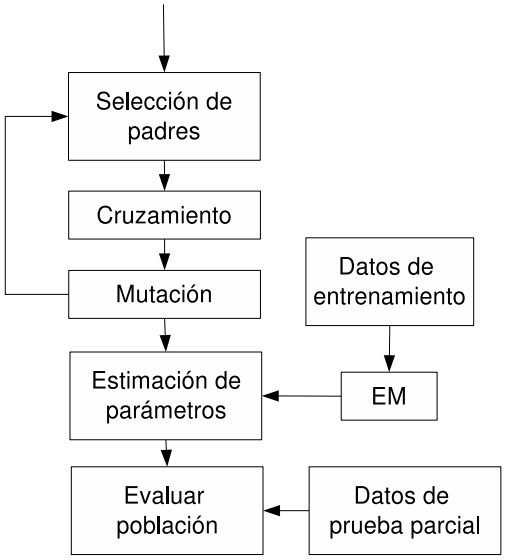

Figura 15. Bloques que conforman el ciclo evolutivo

Los elementos que componen al algoritmo se describen a continuación.

V.2.1 Representación en el problema de aprendizaje del CBSD

La representación basada en grupos (Falkenauer, 1994) descrita anteriormente es utilizada en este trabajo. Ésta representación fue adaptada para el problema del aprendizaje del CBSD. En la representación original un individuo está formado por dos partes, la representación objeto y la representación de grupo.

Para nuestro problema los grupos son los nodos del CBSD y los objetos son los atributos. En la representación propuesta un individuo está formado por l modelos, de

tal forma que el modelo i del individuo j está conformado como se muestra en la Figura 16.

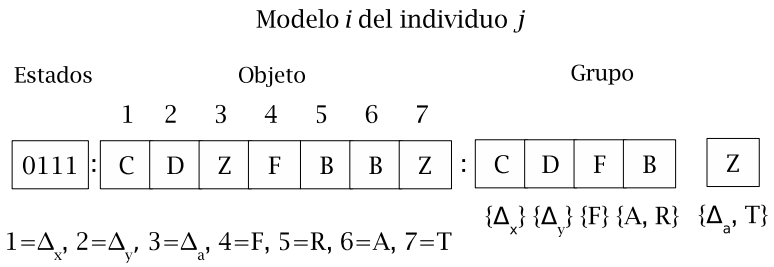

Figura 16. Representación del modelo i del individuo j. La representación está conformada por la parte grupo que representa el número de grupos en el modelo, la parte objeto indica qué atributos pertenecen a cada grupo, un grupo adicional Z es utilizado para depositar atributos eliminados. La representación del número de estados es binaria.

En la parte objeto (en nuestro caso la parte de atributos) cada locus es un identificador para cada atributo y su correspondiente alelo el grupo al que pertenece. La parte grupo contiene el identificador para cada grupo. En nuestro problema un atributo puede no ser asignado a grupo alguno, por lo que se le asigna un identificador especial (Z). El número de estados para el nodo oculto es codificado por medio de una cadena binaria. Por ejemplo, la Figura 16 muestra la representación del modelo i para el individuo j, el nodo oculto tiene siete estados (0111), el atributo Δ_x (locus 1) es asociado al grupo C, Δ_y (locus 2) pertenece al grupo D, el atributo F (locus 4) pertenece al grupo F, los atributos A (locus 6) y R (locus 5) pertenecen al grupo B. La parte grupo indica que se tienen cuatro grupos, los atributos Δ_a (locus 3) y T (locus 7) no son asignados a grupo alguno.

V.2.2 Inicialización de la población

Un individuo está conformado por l modelos, cada uno de estos modelos se inicializa de manera aleatoria. Primero se elige al azar el número de grupos que tendrá el individuo,

como máximo puede haber n grupos, donde n es el número de atributos. Sea k el número de grupos seleccionado, se insertan de manera aleatoria en cada uno de estos grupos los atributos disponibles. En la inicialización todos los atributos son tomados en cuenta. El número de estados de la variable oculta es aleatorio también, pero limitado por un número máximo de estados e_{MAX}.

Una vez que se tiene la representación de cada uno de los individuos, se obtienen sus respectivos parámetros a través del algoritmo Baum-Welch (Rabiner, 1989), donde un conjunto de datos de entrenamiento es utilizado. Posteriormente cada uno de los individuos es evaluado para obtener su valor de aptitud. La descripción de esta etapa se muestra en la Figura 17.

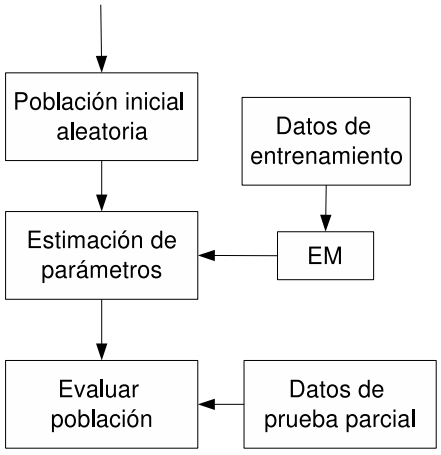

Figura 17. Inicialización de la población

V.2.3 Selección de individuos en el problema de aprendizaje del CBSD

Como ya se mencionó, existen dos tipos de selección durante el proceso evolutivo, la selección de padres que generarán los descendientes y la selección de los individuos que formarán parte de la siguiente generación.

Para seleccionar los padres se utiliza selección por torneo. La seleción por torneo (Eiben y Smith, 2003) es un operador que no requiere ningún conocimiento de la población global, se eligen k individuos (donde k es el tamaño del torneo) que compiten entre sí comparando su valor de aptitud, el individuo seleccionado es aquél que tenga mejor valor de aptitud (menor para los procesos en lo que se desea minimizar o mayor si se desea maximizar), el proceso es repetido hasta que se obtenga el número de padres deseado. En el Algoritmo 2 se muestra el pseudocódigo de la selección por torneo para n padres.

Algoritmo 2 Selección por torneo

```
Entrada:  Población (Pob), tamaño del torneo (k), número de
padres (n).
Salida:  Una población de padres (caldo_cultivo)
Inicio
1 actual_miembro = 1
2 Mientras (actual_miembro ≤ n) hacer
3   Seleccionar de Pob k individuos aleatoriamente
4   Seleccionar el mejor de estos k comparando las aptitudes
5   Identificar este individuo como i
6   caldo_cultivo[actual_miembro] = i
7   actual_miembro = actual_miembro + 1
8 Fin mientras
Fin
```

Mientras el tamaño del torneo sea más grande habrá una mayor posibilidad de que el torneo devuelva padres de aptitud mayor y entre más pequeño se permitirá la sobrevivencia de individuos con aptitud baja.

Para seleccionar a los sobrevivientes, es decir a los individuos que formarán parte de la siguiente generación, se utiliza la selección (μ+λ) (Schwefel, 1993), la cual inicialmente se utilizó en estrategias evolutivas. Con esta estrategia, μ padres y λ descendientes compiten para sobrevivir y los μ mejores entre descendientes y padres son seleccionados como la nueva generación.

V.2.4 Cruzamiento para el problema de aprendizaje del CBSD

El cruzamiento propuesto para el aprendizaje del CBSD está basado en el cruzamiento propuesto por Falkenauer (1994). Se agrega un identificador Z para ubicar a los objetos (atributos) que no pertenecen a grupo alguno y se indican correcciones específicas para cuando se va a ajustar un grupo (paso 4 del cruce propuesto por Falkenauer (1994)). El procedimiento es el siguiente:

1. Seleccionar aleatoriamente dos puntos de cruzamiento, delimitando la sección de cruce en cada uno de los padres (Figura 18.a).

2. Insertar el contenido de la sección de cruce y del primer padre al primer lugar de cruzamiento del segundo padre. Si el primer padre tiene elementos en Z, insertarlos en Z del segundo padre (Figura 18.b).

3. Eliminar todos los objetos repetidos en los grupos donde ellos son miembros en el segundo padre y eliminar aquellos elementos que también estén en Z (Figura 18.c).

4. De los grupos modificados, realizar una de las siguientes acciones 1)Conservar los grupos, 2)Separar los elementos del grupo formando un grupo independiente cada elemento y 3)Deshacer los grupos y re-insertar los elementos de manera aleatoria en los grupos restantes (Figura 18.d).

5. Aplicar pasos de 2 a 4 a los dos padres con sus roles invertidos para generar al segundo hijo.

La Figura 18 muestra un ejemplo del cruce de dos individuos (P1 y P2).

El cruzamiento para el número de estados en el nodo oculto está basado en el cruce de un punto para cadenas binarias (Eiben y Smith, 2003). Recordemos que un clasificador

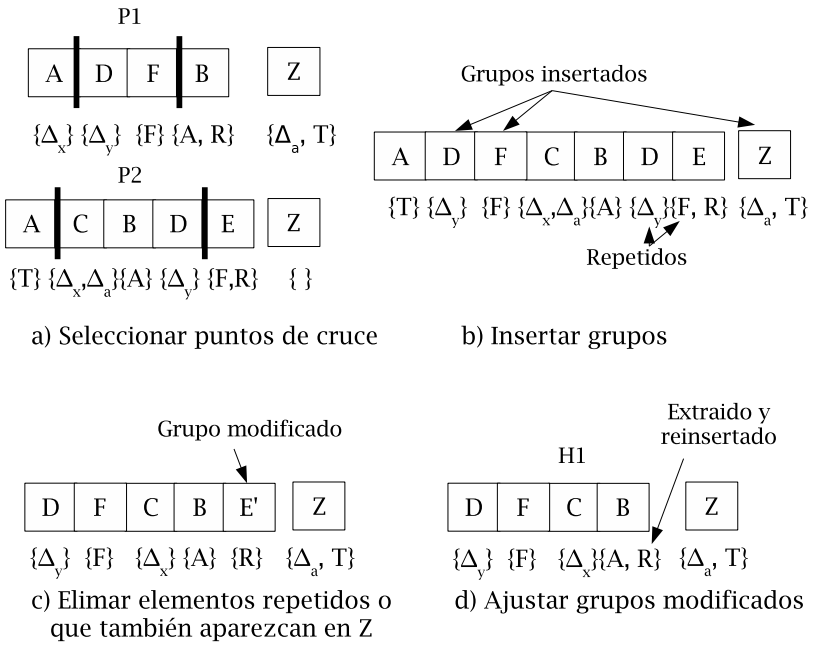

Figura 18. Cruzamiento propuesto para el aprendizaje del CBSD. a) De manera aleatoria se eligen dos puntos de cruce en los dos padres, b) la sección definida por estos dos puntos es insertada del primer padre al segundo. c)y d) Posteriormente se realizan ajustes en el individuo para evitar elementos repetidos. Si el primer padre tiene elementos en Z estos son heredados al segundo padre.

esta condormado por varios modelos, por lo que el proceso de cruce es aplicado a cada modelo seleccionado para cruzamiento.

V.2.5 Mutación

La mutación nos permite manejar el proceso de eliminación de atributos irrelevantes, en cada generación y para cada individuo hay una probabilidad P_m de aplicar mutación. Para este proceso se cuenta con un grupo auxiliar al cual llamaremos grupo Z, para indicar a través de éste qué atributos se encuentran fuera de la solución.

Para este trabajo la mutación se lleva a cabo como lo muestra el Algoritmo 3.

La Figura 19 muestra los cambios que puede sufrir un individuo a través de la mutación. Puede observarse que el grupo Z no participa en la parte grupo. Se tienen dos opciones que son igualmente probables: inserción o eliminación. En la inserción podemos seleccionar insertar un atributo o insertar un grupo. Si se inserta un atributo, éste es tomado de Z y es insertado en cualquiera de los grupos existentes con la misma probabilidad. Si insertamos un grupo, sus elementos son tomados de Z, el tamaño y composición son determinados al azar. En eliminación, se tienen también dos opciones, la de borrar un atributo (seleccionado aleatoriamente desde un grupo) o borrar un grupo. En ambos casos los elementos borrados son insertados en Z.

Se puede observar en la Figura 19 el caso donde una variable es borrada, la variable A del grupo B es borrada. El grupo B es borrado en el ejemplo correspondiente a eliminación de grupo. Para inserción también se tienen dos opciones, podemos ver cómo el atributo T es insertado como parte del grupo A, y el grupo C, conformado por el atributo T, es insertado cómo un nuevo grupo. Observemos que en el caso donde Z esta vacío sólo se permite inserción.

La operación de mutación para el número de estados, es la mutación de un bit elegido al azar, lo cual sucede con una probabilidad P_m.

Algoritmo 3 Pseudocódigo de la mutación para el aprendizaje del CBSD.

```
Entrada:  Individuo (I).
Salida:  Un individuo mutado.
1  Determinar al azar acción a realizar (eliminar o insertar)
2  Si eliminar entonces
3    Determinar al azar si eliminar grupo o atributo
4      Si eliminar atributo
5        Determinar de manera aleatoria de qué grupo
6        Del grupo elegido, determinar al azar que atributo
          eliminar
7        Eliminar el atributo, depositarlo en el grupo Z
8      Si eliminar grupo
9        Determinar al azar qué grupo
10       Depositar los elementos del grupo eliminado en Z
11 Si insertar entonces
12   Determinar al azar qué se va a insertar grupo o atributo
13     Si insertar atributo
14       Determinar de manera aleatoria qué atributo se tomará
          de Z
15       Determinar al azar en qué grupo del individuo se va a
         insertar el atributo
16       Insertar el atributo, borrarlo de Z
17     Si insertar grupo
18       Determinar al azar cuántos atributos de Z se tomarán
         para formar el grupo
19       Tomar al azar los atributos de Z
20       Borrar de Z los atributos elegidos
21       Insertar el grupo en el individuo
```

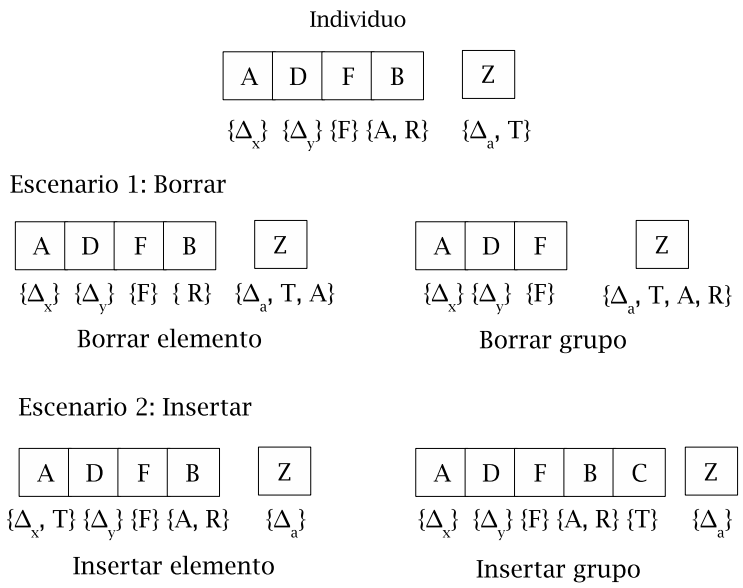

Figura 19. Mutación para el aprendizaje del CBSD. La mutación permite una de dos acciones, borrar o insertar, ya sea un atributo o un grupo. Si la acción es borrar, los atributos son depositados en Z. Si la acción es insertar, los atributos son tomados de Z.

V.2.6 La función de aptitud

Algoritmos de aprendizaje para RBDs son no supervisados, en el sentido de que todos los nodos de la red son tratados de igual forma. Sin embargo, cuando RBDs son utilizadas como clasificadores (que es el caso del CBSD) se debe preferir un enfoque supervisado, en el cual el nodo de clasificación es identificado y el aprendizaje es optimizado para clasificación (Choudhury *et al.*, 2002).

Tomando en cuenta esto, en nuestro problema la función de aptitud está dada por la siguiente expresión:

$$calidad(i) = \alpha PRN(i) + (1 - \alpha)(1 - complejidad(i)) \tag{49}$$

donde $PRN(i)$ es el porcentaje de reconocimiento normalizado y $complejidad(i)$ es el número de parámetros normalizado, obtenidos al evaluar el individuo i. $PRN(i)$ es obtenido una vez entrenados todos los modelos. Para cada muestra de una clase se

realiza lo siguiente:

- La muestra de prueba se presenta a las cada una de las instancias del modelo que constituyen el clasificador, entonces la probabilidad de la muestra dada cada modelo se calcula por medio del algoritmo hacia-adelante.

- Mediante el criterio de máxima verosimilitud, la instancia que maximice la probabilidad de la muestra corresponde a la clase real.

Los resultados obtenidos al repetir el proceso anterior para todas las clases, pueden registrarse en una matriz de confusión. La matriz de confusión es una matriz cuadrada de $n\_clases \times n\_clases$ donde n_clases es el número de clases existentes en el proceso modelado. La diagonal de dicha matriz contiene el conteo de las muestras bien clasificadas, valores fuera de esta diagonal corresponden a errores de clasificación. Por ejemplo, en la Figura 20 se muestra una matriz que tiene dos clases, A y B. Las muestras clasificadas correctamente como A y B, están representadas como a y b respectivamente; c y d corresponden a errores de clasificación, es decir d muestras de la clase B fueron clasificadas como de la clase A, y c muestras de la clase A fueron clasificadas como de la clase B. Como se puede observar, la suma de la diagonal de ésta matriz corresponde al porcentaje de reconocimiento (PRN) o exactitud del modelo, el cual se define de la siguiente manera:

$$PRN(i) = \frac{\#\ de\ muestras\ correctamente\ clasificadas}{\#\ total\ de\ muestras} \tag{50}$$

$complejidad(i)$ es obtenido de la suma de los parámetros de cada modelo en el clasificador, α es un valor que permite indicar el peso que se le da al porcentaje de reconocimiento y a la complejidad respectivamente. El número de parámetros de un modelo es obtenido como sigue:

$$\#parametros = \sum_{i=1}^{n} ||Pa(X_i)|| * (||X_i|| - 1) \tag{51}$$

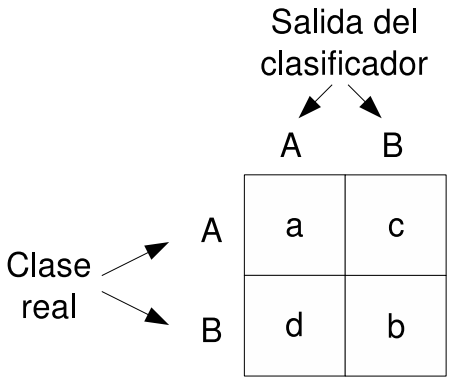

Figura 20. Matriz de confusión con dos clases. Cada renglón corresponde a las muestras de una clase particular presentadas. Las columnas corresponden a la salida de la clasificación.

donde n es el número de nodos, incluyendo el nodo clase, $||Pa(X_i)||$ es el número de parámetros de los padres del nodo X_i, el cual está compuesto por un grupo de variables. $||X_i||$ es el número de parámetros del nodo X_i. Éste valor es definido como sigue:

$$||X_i|| = \prod_{R_j \in X_i} |R_j| \tag{52}$$

donde $|R_j|$ es el número de valores que la variable R_j, un miembro de X_i puede tomar. Note que si el nodo X_i no tiene padres entonces $||Pa(X_i)|| = 1$. En la ecuación (49) α define un compromiso específico entre exactitud y complejidad.

V.3 Resumen

En este capítulo se presentó un algoritmo para el aprendizaje evolutivo del CBSD, el cual busca la mejor estructura del CBSD en base a una medida de calidad. Los elementos que lo conforman son los mismos que un algoritmo genético estándar, y pueden ser resumidos en tres etapas: 1) Generación de la población inicial, 2) Obtención de descendientes y 3) selección de sobrevivientes. El método itera entre las etapas 2 y 3 hasta que un criterio de paro es alcanzado.

El algoritmo propuesto está basado en la representación de grupos (Falkenauer, 1994), utiliza cromosomas de longitud variable permitiendo así la evolución de la estructura. Los operadores genéticos para el problema de aprendizaje son descritos. El operador de cruce permite realizar la búsqueda para detectar la dependencia de atributos, el operador de mutación introduce cambios de borrado e inserción para detectar atributos irrelevantes. La representación del número de estados del nodo clase oculto es binaria, el cruce es de un punto y la mutación cambia de forma aleatoria el valor de un bit. Para seleccionar los padres se utiliza selección por torneo (Eiben y Smith, 2003). Para seleccionar a los sobrevivientes, se utiliza la selección $(\mu + \lambda)$ (Schwefel, 1993).

Puede observarse que la representación y los operadores genéticos propuestos permiten la evolución simultánea de la estructura (agrupamiento de atributos dependientes y eliminación de atributos irrelevantes) y la obtención del número de estados de la variable oculta. Los parámetros de cada individuo en la etapa 1 y 2 son obtenidos por medio del algoritmo EM. La evaluación se realiza con un conjunto de datos de prueba parcial (datos que no son utilizados en la obtención de parámetros) y busca establecer un compromiso entre exactitud y complejidad (número de parámetros). El mejor individuo obtenido al finalizar el ciclo evolutivo es evaluado con un segundo conjunto de datos de prueba parcial para obtener el porcentaje de reconocimiento correspondiente.

En el siguiente capítulo se presentan los resultados obtenidos al evaluar el algoritmo propuesto en el reconocimiento de nueve ademanes de la mano.

Capítulo VI

Experimentos y resultados

VI.1 Introducción

En el capítulo anterior se describió el funcionamiento del método evolutivo propuesto. Este método permite aprender (a partir de datos) la estructura del CBSD, incluyendo la asociación de atributos dependientes, eliminando atributos irrelevantes y obteniendo el mejor número de estados para el nodo clase oculto. Para evaluar el método propuesto se utilizó una base de datos conteniendo observaciones de la ejecución de ademanes de la mano derecha proporcionado por Avilés-Arriaga (2006). Se llevaron a cabo cuatro experimentos con diferentes valores para los parámetros de entrada del algoritmo para el reconocimiento de ademanes ejecutados por un usuario. A continuación se describen los datos utilizados y la configuración de los experimentos realizados.

VI.2 Reconocimiento visual de ademanes

Cuando el ser humano gesticula con sus brazos y manos emite una serie de mensajes que pueden ser interpretados como órdenes o simplemente como refuerzo de instrucciones para el sistema que los esté procesando. La comunicación gestual es un factor importante en el desarrollo de formas de interacción más naturales entre los humanos y las máquinas. Por ejemplo, en ambientes con ruido, proporciona una alternativa a los sistemas de comunicación basados en voz (Wasson *et al.*, 1998).

El reconocimiento visual de ademanes es un área de investigación que ha despertado interés en los últimos años (Stark *et al.*, 1995; Ramamoorthy *et al.*, 2003; Kolsh, 2004).

Es el proceso por el cual ademanes hechos por el usuario se hacen conocer al sistema (Baratoff y Searles, 2006).

Los ademanes, en particular de la mano, son un medio de comunicación, similar al lenguaje hablado (Pavlovic, 1999). La producción y percepción de ademanes pueden ser descritas usando un modelo comúnmente encontrado en el campo del reconocimiento del lenguaje hablado. De acuerdo a este modelo, el ademán es originado como un concepto mental del emisor, posiblemente en conjunción con otras modalidades tales como el habla. Estos son expresados a través del movimiento de brazos y manos. Los observadores perciben los ademanes como un flujo de imágenes visuales, las cuales interpretan usando el conocimiento que tienen acerca de estos ademanes. Actualmente, la mayoría de los sistemas de visión para reconocimiento de ademanes pueden estructurarse como se muestra en la Figura 21, donde se tiene como entrada al sistema una secuencia de video capturada con un dispositivo pasivo, el cual comúnmente es una cámara.

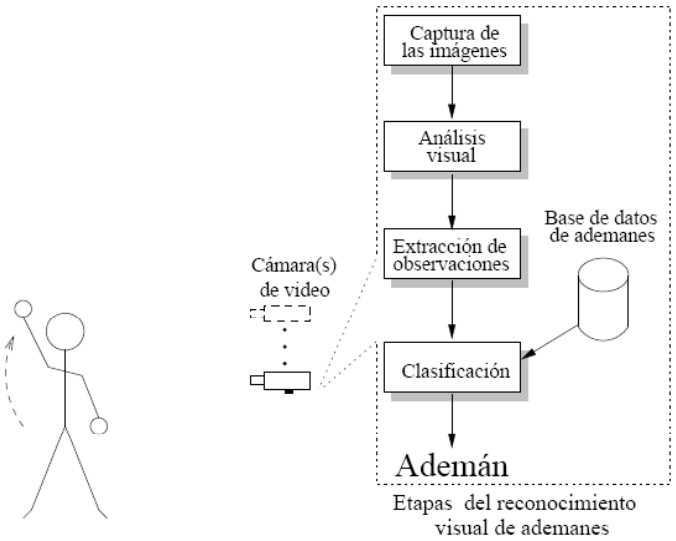

Figura 21. Etapas del reconocimiento visual de ademanes implementadas en un robot móvil, esquema tomado de Avilés-Arriaga (2006)

Inicialmente el reconocimiento visual de ademanes obtiene una secuencia de imágenes a través de una o más cámaras de video. El sistema pasa a una fase de análisis visual en donde se llevan a cabo dos tareas básicas; una de ellas es el seguimiento, el cual permite determinar la posición del miembro humano con el que se lleva a cabo el ademán. La segunda tarea es la segmentación de imágenes, la cual es un proceso que divide una escena en un conjunto de regiones disjuntas basándose en características de similitud. Esta fase va a permitir extraer de la escena el miembro del cuerpo que está llevando a cabo el ademán. Una vez hecho esto se obtienen las medidas de los atributos o características que se consideran relevantes para la descripción del ademán. Finalmente, una vez identificados los elementos encontrados en las imágenes analizadas y puestos en contexto, se activa el módulo del reconocimiento de ademanes, o clasificación. El reconocimiento de ademanes es la fase en la cual los datos analizados de las imágenes visuales de ademanes son reconocidos como un ademán específico; es decir, las estimaciones de los estados del modelo físico son utilizados en alguna forma para inferir cuál es el concepto gestual.

Como puede observarse, cada una de las tareas llevadas a cabo durante el proceso del reconocimiento visual de ademanes por sí solas constituyen un campo amplio de estudio. Esta tesis se enfoca en el último de ellos, la clasificación.

La tarea de un componente clasificador perteneciente a un sistema completo (como el reconocimiento visual de ademanes) es utilizar el conjunto de atributos provistos (por el módulo de extracción de atributos) para asignar el objeto (ademán) a una categoría (clase) (Duda *et al.*, 2001).

VI.3 Los ademanes y los atributos considerados

El método propuesto fue probado en el reconocimiento de nueve ademanes de la mano derecha. Los ademanes considerados se muestran en la Figura (Figura 22). Los nueve

ademanes son :

- Acercarse: el usuario mueve la palma de la mano hacia su torso (Figura 22.a).
- Atención: el usuario levanta su mano encima de su cabeza (Figura 22.b).
- Detener: el usuario debe estirar su brazo hacia el frente y dirige la palma de la mano hacia la videocámara (Figura 22.c).
- Derecha: el usuario mueve la mano hacia su derecha (Figura 22.d).
- Izquierda: el usuario mueve la mano hacia la izquierda (Figura 22.e).
- Girar hacia la izquierda: el usuario con el brazo extendido realiza un movimiento circular en dirección de las manecillas del reloj (Figura 22.f).
- Girar hacia la derecha: el usuario con el brazo extendido realiza un movimiento circular en dirección contraria a las manecillas del reloj (Figura 22.g).
- Saludar: el usuario levanta su mano a la altura del hombro y la balancea de un lado a otro un par de veces (Figura 22.h).
- Apuntar: el usuario estira su brazo y mano sobre su torso como señalando algo frente a él (Figura 22.i).

Todos los ademanes inician y terminan en una posición conocida denominada descanso (Figura 22.j).

Estos nueve ademanes son descritos por siete atributos (Avilés-Arriaga, 2006), tres atributos de movimiento y cuatro de postura. Los atributos son extraídos con un sistema de análisis visual monocular (ver apéndice A para detalles). Los atributos de movimiento son:

- Δ_x. Indica cambios de posición en el eje X.

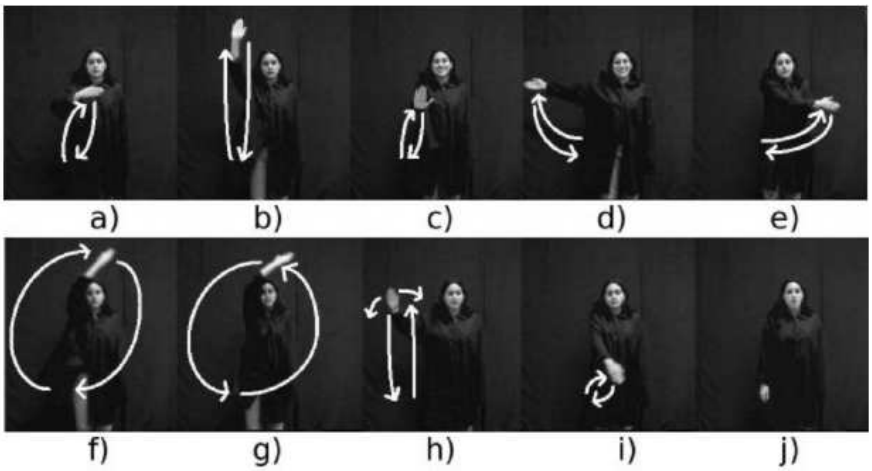

Figura 22. Ademanes considerados. Las flechas indican la trayectoria de cada ademán: a)acercar, b)atención, c)detener, d)derecha, e)izquierda, f)girar a la izquierda, g)girar a la derecha, h)saludar e i)apuntar. La posición de descanso en la que inicia y termina cada ademán se muestra en j) (Imágen tomada de Avilés-Arriaga (2006)).

- Δ_y. Indica cambios de posición en el eje Y.

- Δ_a. Permite registrar el cambio de área de la mano entre dos imágenes consecutivas.

Estos tres atributos pueden tomar uno de tres valores +,-,0, que indican incremento, disminución o sin cambio, dependiendo del atributo. Por ejemplo, $\Delta_a = -$ quiere decir que hubo una disminución en el área de la mano.

Los atributos de postura son:

- *Forma*. Describe la apariencia de la mano, los valores que puede tomar son + si la posición de la mano es vertical, - si la posición de la mano es horizontal o 0 si la mano aparece inclinada hacia la derecha o izquierda.

- *Derecha*. Indica si la mano se encuentra a la derecha de la cabeza (verdadero si se encuentra sobre el torso, falso de lo contrario).

- *Arriba*. Indica si la mano se encuentra arriba de la cabeza.

- *Torso*. Describe si la mano se encuentra sobre el torso del usuario.

Derecha, *Arriba*, *Torso* son atributos binarios, pueden tomar dos valores falso o verdadero. Por ejemplo, $Torso = verdadero$ quiere decir que la mano del usuario se encuentra sobre el torso.

Para realizar el reconocimiento de actividades humanas se consideran diversos aspectos (Davis y Bobick, 1997), como el que una actividad se ejecuta en un lapso variante de tiempo, que no siempre es posible observar la trayectoria completa del ademán por oclusión de la mano y brazo con otras partes del cuerpo, los sistemas de análisis visual no son exactos en la segmentación o seguimiento de la persona y más importante, es muy difícil realizar un ademán dos veces exactamente de la misma manera aún cuando se trate del mismo ejecutante. Por lo que para modelar este tipo de actividades es conveniente utilizar un modelo dinámico.

Los modelos probabilísticos son una forma conveniente para manejar la incertidumbre propia de la ejecución de los ademanes. Particularmente, los modelos ocultos de Markov (Rabiner, 1989) que son un caso especial de las RBDs (Pavlovic, 1999) se han convertido en la técnica estándar en el reconocimiento visual de ademanes (Starner, 1995). En este trabajo se propone aprender CBSDs como una alternativa para modelar procesos dinámicos como el reconocimiento de ademanes.

La estructura del modelo puede ser especificada manualmente utilizando conocimiento acerca del problema. El diseño manual puede insertar una predisposición del modelo, y realizar este proceso será más difícil para redes con muchos atributos. Una alternativa es aprender la red desde datos (Cooper y Herskovits, 1992; Friedman *et al.*, 1998; Heckerman, 1995). Algoritmos de aprendizaje estructural logran esto buscando sobre el espacio de estructuras de redes para encontrar la estructura la cual se ajusta

mejor a los datos.

VI.4 Los experimentos

Se realizaron cuatro experimentos para evaluar la exactitud de clasificación de ademanes en los CBSDs obtenidos. El conjunto de datos está conformado de 50 muestras para cada uno de los nueve ademanes, tomados de un usuario. Se seleccionaron D_e muestras por ademán para construir el conjunto de datos de entrenamiento completo, un conjunto D_{p1} de datos de prueba parcial es necesario para evaluar (calcular el valor de aptitud) cada uno de los individuos en el proceso evolutivo. Finalmente, se evalúa la exactitud de clasificación de los mejores individuos con las D_{p2} muestras de prueba restantes.

La configuración de los experimentos se muestra en la Tabla III

Tabla III. Configuración de experimentos realizados

Exp	D_e	D_{p1}	D_{p2}	α	P_m	$PobSize$	Max_Iter sin cambio	Max_Iter
1	10	10	30	0.8	0.35	12	4	20
2	10	15	25	0.8	0.35	12	4	20
3	10	10	30	0.7	0.35	12	4	20
4	10	15	25	0.7	0.35	12	4	20

Como puede observarse, la diferencia entre estos cuatro experimentos es el tamaño de D_{p1} y el factor de ponderación α. El algoritmo es ejecutado 10 veces para cada experimento. Los clasificadores obtenidos en cada experimento son comparados con el CBSD básico, el cual es un CBSD donde cada uno de los modelos que lo conforman consideran todos los atributos independientes dada la clase y el número de estados para el nodo clase oculto es de dos. Este clasificador se considera básico debido a que sería el clasificador definido por un usuario sin realizar mejora estructural ni realizando búsqueda del mejor número de estados para el nodo clase oculto.

En la Figura 23 pueden observarse los porcentajes de reconocimiento de los clasificadores de los diez mejores individuos producidos por el proceso evolutivo para el experimento 1. En nueve de diez ejecuciones los clasificadores evolucionados obtienen un porcentaje de reconocimiento más alto que el clasificador básico (por arriba del 94%) cuyo porcentaje de reconocimiento es de 93.7%.

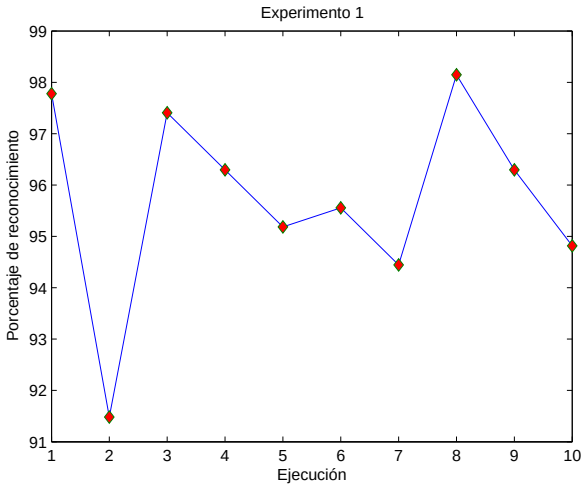

Figura 23. Porcentaje de reconocimiento para los clasificadores evolucionados en diez ejecuciones del experimento 1.

La Figura 24 muestra los porcentajes de reconocimiento para los clasificadores producidos por el proceso evolutivo en el experimento 2. Para este caso el clasificador básico obtiene un porcentaje de reconocimiento de 94.67%. Los parámetros de entrada son iguales que en el experimento anterior, excepto que el tamaño del conjunto de datos de prueba parcial D_{p1} es más grande, lo cual provoca que el conjunto de datos de prueba final D_{p2} se reduzca. Como puede observarse, en todas las ejecuciones el clasificador evolucionado obtiene un porcentaje de reconocimiento mayor al del clasificador básico (superior al 95%).

Los porcentajes de reconocimiento obtenidos en el experimento 3 son todos superiores

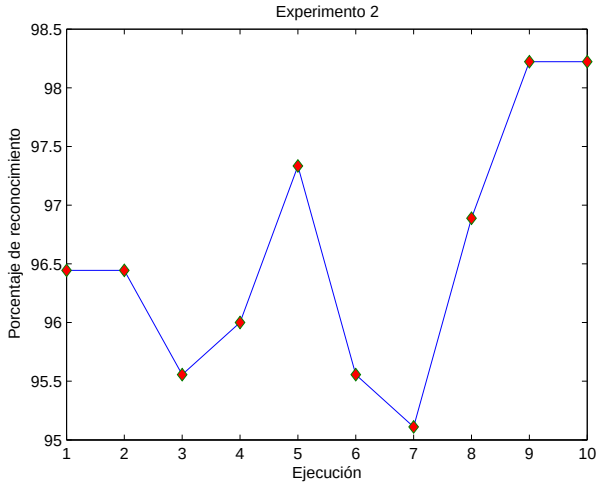

Figura 24. Porcentaje de reconocimiento para los clasificadores evolucionados en diez ejecuciones del experimento 2.

al clasificador básico (ver Figura 25), por arriba del 94%. Los parámetros del algoritmo son los mismos que en el experimento 1 , excepto el valor de α que se fijó en 0.7.

Finalmente, los porcentajes de reconocimiento de los clasificadores de los diez mejores individuos producidos por el proceso evolutivo para el experimento 4 se muestran superiores al clasificador básico, excepto en la ejecución nueve. A pesar de esto, podemos observar que los individuos que sí superan al clasificador básico lo hace en una mayor proporción que en los experimentos anteriores (ver Figura 26).

La Tabla IV muestra la media y desviación estándar de la exactitud y la función de aptitud de los mejores individuos producidos en el proceso de aprendizaje evolutivo.

La Tabla V muestra la media y desviación estándar de los tiempos obtenidos en el proceso de aprendizaje evolutivo en diez ejecuciones de los experimentos descritos en la Tabla III.

En la Figura 27 se muestran los intervalos de confianza obtenidos para cada uno de los cuatro experimentos realizados con un nivel de confianza del 95%.

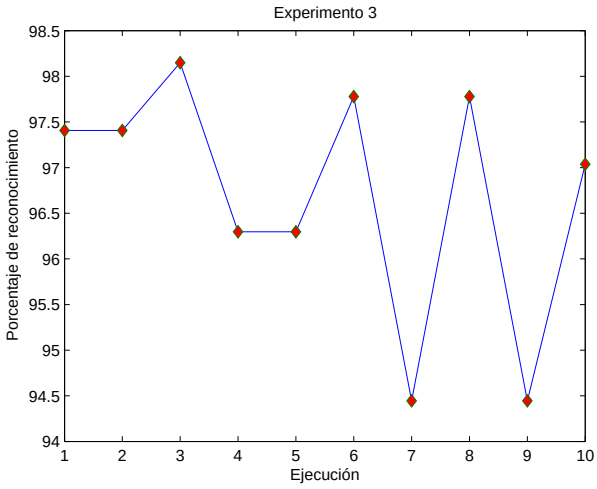

Figura 25. Porcentaje de reconocimiento para los clasificadores evolucionados en diez ejecuciones del experimento 3.

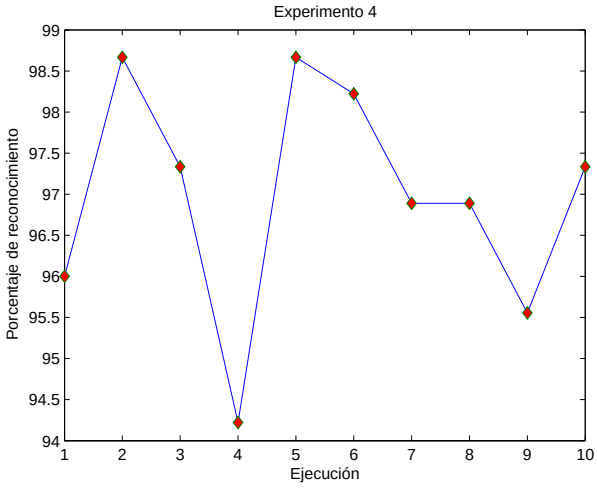

Figura 26. Porcentaje de reconocimiento para los clasificadores evolucionados en diez ejecuciones del experimento 4.

Tabla IV. Promedio y desviación estandar de exactitud y aptitud de los mejores individuos obtenidos en diez ejecuciones de cuatro experimentos

	Exactitud (promedio)	Aptitud (promedio)	Exactitud. (Desv. Std)	Aptitud. (Desv. Std)
Exp1	0.957	0.993	0.020	0.006
Exp2	0.966	0.989	0.011	0.003
Exp3	0.967	0.994	0.013	0.003
Exp4	0.970	0.986	0.014	0.004

Tabla V. Promedio ($\bar{t}$) y desviación estandar $DE(t)$ de los tiempos obtenidos en diez ejecuciones de cuatro experimentos

	$\bar{t}$ en minutos)	$DE(t)$ (en minutos)
Exp1	186.0742	43.5421
Exp2	215.7301	40.4538
Exp3	195.61	46.322
Exp4	206.3951	43.6830

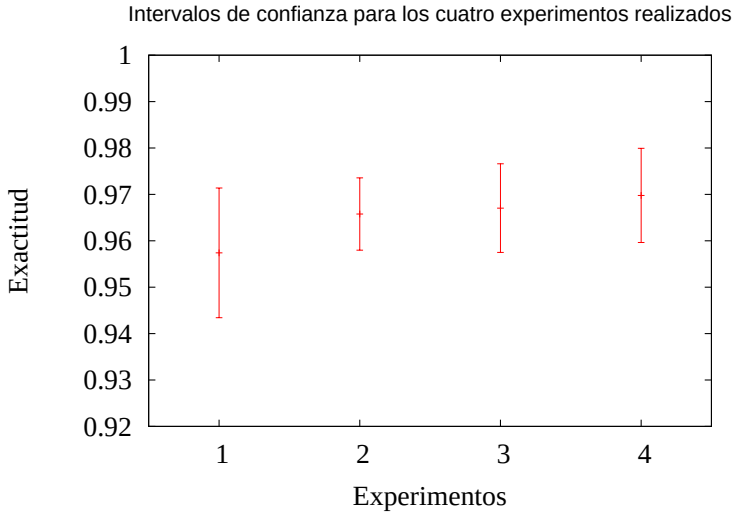

Figura 27. Intervalos de confianza obtenidos para los cuatro experimentos realizados.

La Figura 28 muestra el ejemplo de uno de los clasificadores evolucionados en el experimento 2. Puede observarse cómo cada modelo tiene un número de estados y estructura diferente. Recordemos que cada modelo corresponde a un ademán por reconocer. Por ejemplo, el primer modelo corresponde al ademán *acercar*; está conformado por cinco nodos hijo, donde los atributos T, Δ_x, A y F son independientes dada la clase C, los atributos Δ_y y Δ_a son asociados en un mismo nodo, el nodo clase oculto tiene seis estados y el atributo D fue eliminado. El segundo modelo correspondiente al ademán *atención* tiene cuatro nodos hijo con los atributos Δ_x y Δ_a en el primer nodo hijo, Δ_y, F y A, en el segundo. Los atributos D y T son independientes dado el nodo clase. Ningún atributo fue eliminado y el nodo clase tiene tres estados.

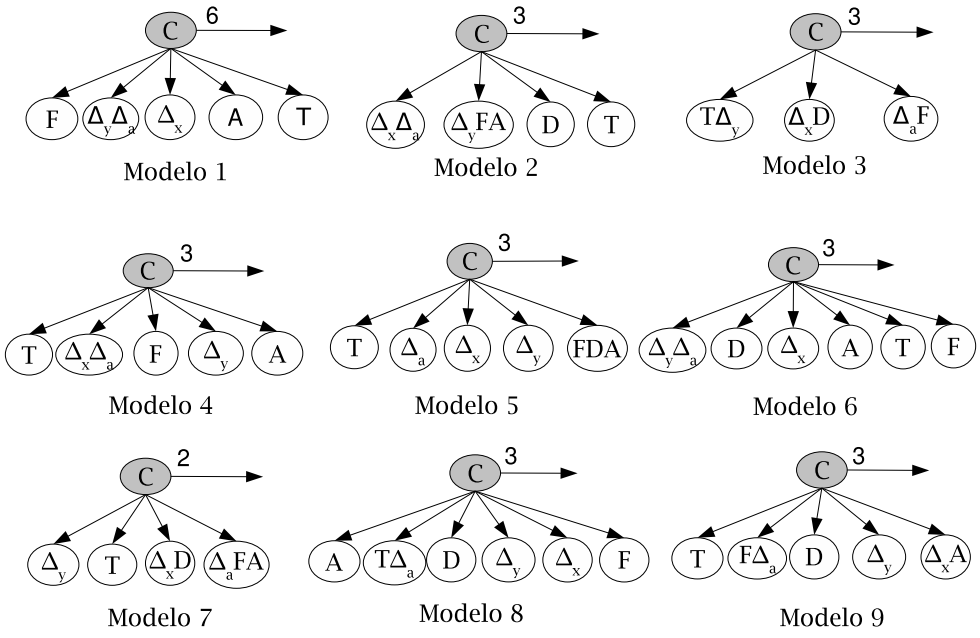

Figura 28. CBSD para los 9 ademanes considerados (*acercar, atención, derecha, izquierda, detener, girar a la derecha, girar a la izquierda, apuntar y saludar*).

Los porcentajes de reconocimiento presentados en la Tabla VI corresponden al clasificador evolucionado presentado en la Figura 28 y al CBSD básico aprendido y evaluado con el mismo conjunto de muestras que el CBSD evolucionado. El clasificador evolucionado supera al clasificador básico en un 2.7%.

Tabla VI. Porcentajes de reconocimiento utilizando el CBSD: el modelo básico vs. el modelo evolucionado.

Ademán	**Exactitud CBSD básico**	**Exactitud CBSD evolucionado**
Acercar	96%	100%
Atención	100%	100%
Derecha	100%	100%
Izquierda	96%	84%
Detener	100%	96%
Girar a la derecha	100%	100%
Girar a la izquierda	100%	100%
Apuntar	88%	96%
Saludar	72%	100%
Promedio	94.67%	97.33%

VI.5 Análisis

El proceso evolutivo introduce la eliminación y combinación de variables al mismo tiempo que se evalúan diferentes números de estado para el nodo clase oculto, hasta que el clasificador más simple con la más alta exactitud es obtenido.

Como se describió antes, un CBSD básico está conformado por modelos donde todos los atributos del conjunto de datos son tomados en cuenta. Cada uno de ellos son un nodo hijo en la estructura del modelo; es decir, se supone independencia entre ellos dado el nodo clase oculto; además, el nodo clase oculto se considera de dos estados. En los resultados de exactitud mostrados en la Tabla VI se puede observar como la asociación de atributos, así como la búsqueda del mejor número de estados para el nodo oculto,

incrementa el porcentaje de reconocimiento de cada modelo y en general del CBSD.

El proceso de aprendizaje evolutivo muestra que un CBSD evolucionado es mejor en el criterio de exactitud promedio. Para validar si esta diferencia es significativa se realizaron 30 procesos de aprendizaje y prueba. Se tomaron diez muestras aleatorias de las cincuenta disponibles para obtención de parámetros y las cuarenta muestras restantes son utilizadas para prueba. Este proceso se realizó tanto para el clasificador de la Figura 28 como para el clasificador básico. La prueba de significancia se hizo con la prueba de suma de rangos de Wilcoxon (Devore, 1995) dado que los porcentajes de reconocimiento no tenían una distribución normal.

El valor obtenido de la prueba fue $p = 1.6031e^{-10}$ el cual es menor que 0.05 y menor que 0.01. Este resultado indica que los porcentajes obtenidos por el CBSD evolucionado y el clasificador básico son significativamente distintos. Por lo que el clasificador evolucionado es significativamente mas exacto.

El algoritmo propuesto devuelve modelos que describen mejor al ademán asociado. Esto es debido a que las relaciones entre atributos y el número de estados del nodo clase oculto están definidos por muestras del ademán correspondiente en el proceso evolutivo. Por ejemplo, el modelo 4 (ademán *izquierda*) en la Figura 28 considera al par de atributos Δ_x y Δ_a como dependientes entre si, los atributos Δ_y, T, F y A son todos independientes dado el nodo clase oculto, el atributo D es irrelevante para este ademán en particular y considera que el número de estados del nodo clase oculto es tres, todo esto a partir del conjunto de datos. En contraste, el modelo básico para este mismo ademán, considera todos los atributos, es decir, el modelo básico utiliza información irrelevante (el atributo D) y no es capaz de representar las dependencias que existen en el conjunto de datos.

El método propuesto es capaz de aprender un configuración específica (asociación de atributos, eliminación de atributos y número específico de estados) para cada modelo

del clasificador.

VI.6 Implementación

Fue utilizado el algoritmo EM (Baum-Welch) con el mismo criterio de convergencia para estimar cada instancia del CBSD. Las probabilidades de transición y observación para todos los modelos en la población fueron inicializados con distribuciones uniformes discretas. Las probabilidades a priori iniciales del nodos clase oculto se estableció como $P(C_1 = c_1) = 1$. La topología aprendida entre los estados del nodo clase oculto es lineal como se muestra en la Figura 29. La probabilidad de cada secuencia de un ademán A, $P(A|.)$, fue calculada utilizando el algoritmo Forward (Rabiner, 1989). Todos los experimentos fueron realizados en una PC con procesador AMD Athlon 1.8Ghz con 3Gb de RAM, el software utilizado fue Matlab 7.0 sobre una plataforma Windows XP.

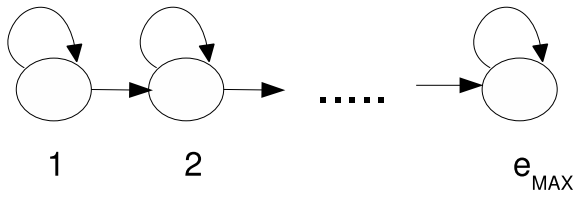

Figura 29. Topologia lineal de e_{MAX} estados para el nodo clase oculto en el aprendizaje del CBSD.

VI.7 Resumen

El proceso evolutivo introduce la eliminación y combinación de variables al mismo tiempo que se evalúan diferentes números de estado para el nodo clase oculto, hasta que el clasificador más simple con la más alta exactitud es obtenido. El algoritmo propuesto fue evaluado con datos conteniendo observaciones de la ejecución de nueve ademanes de la mano derecha proporcionado por Avilés-Arriaga (2006). Se llevaron

a cabo cuatro experimentos con diferentes valores para los parámetros de entrada del algoritmo para el reconocimiento de ademanes ejecutados por un usuario.

El proceso de aprendizaje evolutivo muestra que un CBSD evolucionado es mejor en el criterio de exactitud promedio que un clasificador básico; además, cada uno de los modelos evolucionados describe mejor al ademán asociado. Esto es debido a que las relaciones entre atributos y el número de estados del nodo clase oculto están definidos por muestras del ademán correspondiente en el proceso evolutivo.

El método propuesto es capaz de aprender un configuración específica (asociación de atributos, eliminación de atributos y número específico de estados) para cada modelo del clasificador.

Capítulo VII

Conclusiones y trabajo futuro

VII.1 Resumen

La estructura de una red puede ser especificada manualmente utilizando conocimiento acerca del problema. Pero el diseño manual puede insertar una predisposición del modelo, y será difícil para redes más complicadas con muchos atributos. Una alternativa es aprender la red desde datos. Algoritmos de aprendizaje estructural logran esto buscando sobre el espacio de estructuras de redes para encontrar la estructura la cual mejor soportada por los datos. Esto requiere un un función de puntaje para estructuras candidatas y un procedimiento de búsqueda eficiente.

El CBSD es una extensión del CBS, que se puede describir como una RBD (en particular como un MOM), supone independencia condicional dado el nodo clase oculto, que el proceso descrito es estacionario y Markoviano. Debido a la suposición de independencia condicional, al igual que el CBS, el rendimiento del CBSD disminuye bajo ciertas condiciones, por lo que se requieren métodos de aprendizaje que permitan superar dicha limitante.

Por otro lado, en procesos del mundo real, generalmente se pueden identificar sus salidas. Sin embargo el origen de dichas salidas es desconocido. En una base de datos a esta falta de la información se le conoce como variable oculta, es decir, una variable para la cual no se ha tenido lectura de ninguno de sus valores en cada una de las muestras tomadas de los datos. El CBSD cuenta con una variable de este tipo que le permite modelar la relaciones temporales de procesos dinámicos, por lo que la determinación del mejor número de estados de ésta variable es un problema a resolver.

Siguiendo los enfoques propuestos por Sucar *et al.* (1993, 1994) y Pazzani (1996) podemos identificar dependencias entre atributos, por lo que cada uno de los nodos hijo del CBSD pueden ser vistos como grupos conteniendo uno o más atributos. Sin embargo este problema crece exponencialmente con el número de atributos. Por lo tanto, no podemos explorar exhaustivamente el espacio de solución, aún para un número pequeño de atributos, por lo que se requiere una alternativa a la fuerza bruta para encontrar un agrupamiento G óptimo o cercano al óptimo.

En este trabajo se propone un método basado en AGs capaz de obtener lo siguiente:

- La estructura del CBSD.
- Los parámetros asociados a la estructura resultante.
- Un número de estados optimizado para el nodo clase oculto
- Selección de los mejores atributos para cada modelo.

El algoritmo propuesto está basado en la representación de grupos (Falkenauer, 1994), utiliza cromosomas de longitud variable permitiendo así la evolución de la estructura. El operador de cruce permite realizar la búsqueda para detectar la dependencia de atributos, el operador de mutación introduce cambios de borrado e inserción para detectar atributos irrelevantes. La representación del número de estados del nodo clase oculto es binaria, el cruce es de un punto y la mutación cambia en forma aleatoria el valor de un bit. Para seleccionar los padres se utiliza selección por torneo (Eiben y Smith, 2003). Para seleccionar a los sobrevivientes, se utiliza la selección $(\mu + \lambda)$ (Schwefel, 1993).

La representación y los operadores genéticos propuestos permiten la evolución simultánea de la estructura (agrupamiento de atributos dependientes y eliminación de atributos irrelevantes) y la obtención del número de estados de la variable oculta. Los

parámetros de cada individuo son obtenidos por medio del algoritmo EM. La evaluación se realiza con un conjunto de datos de prueba parcial (datos que no son utilizados en la obtención de parámetros) y busca establecer un compromiso entre exactitud y complejidad (número de parámetros). El mejor individuo obtenido al finalizar el ciclo evolutivo es evaluado con un segundo conjunto de datos de prueba parcial para obtener el porcentaje de reconocimiento correspondiente.

Los resultados experimentales muestran que la red evolucionada tiene una exactitud de clasificación más alta que el clasificador bayesiano simple dinámico básico con una mejora promedio de 2.7% en el reconocimiento de los ademanes.

VII.2 Aportaciones

- **Un método de aprendizaje basado en un enfoque evolutivo para CBSD que incluye mejora estructural.**

  Un enfoque evolutivo ha sido propuesto para resolver el problema de aprendizaje estructural en el diseño de un CBSD. El diseño de la mejor red es modelado como un problema de optimización que mide la exactitud de clasificación ponderado por la complejidad de la red resultante.

- **La representación y los operadores genéticos correspondientes.**

  Se propone un representación basada en grupos, los cromosomas son de longitud variable. Los operadores genéticos trabajan en la parte grupo. El operador de cruce permite realizar la búsqueda del mejor agrupamiento, el operador de mutación permite identificar variables irrelevantes depositándolas en un grupo denominado Z. La representación del número de estados es binaria. El cruce es de un solo punto y la mutación es de un bit.

- **Una función objetivo simple basada en el porcentaje de reconocimiento**

ponderado por la complejidad de la red.

Se propone una métrica sencilla que busca un compromiso entre el porcentaje de reconocimiento y la complejidad de la red. La complejidad de la red esta determinada por el número de parámetros del clasificador.

VII.3 Trabajo futuro

- **Realizar pruebas con muestras de ademanes con variaciones de distancia, rotación y de distintos usuarios.**

  En aplicaciones reales del reconocimiento de ademanes es difícil suponer que un usuario ejecutará siempre los ademanes en una posición frontal hacia la videocámara, a la misma distancia o bien que solo un usuario es el que utilizará el sistema. Para desarrollar aplicaciones que operen en ambientes reales es necesario relajar estas restricciones. Por lo que se requiere analizar el comportamiento de los clasificadores aprendidos con muestras de ademanes con variaciones de distancia, rotación y de distintos usuarios.

- **Extender el método propuesto para incluir el problema de discretización de variables continuas.**

  En un clasificador bayesiano los atributos continuos pueden ser manejados suponiendo una distribución normal o discretizando. La suposición de distribución normal de los datos no siempre se cumple, más aun, en la mayoría de los casos se desconoce. Por lo que realizar un proceso de discretización para manejar estos atributos parece conveniente.

- **Realizar el aprendizaje de cada modelo de manera independiente.**

  Análizar el comportamiento del aprendizaje donde un individuo está conformado

por un solo modelo, para que de esta forma al agregar nuevas clases, solo sea necesario aprender el nuevo modelo y no el conjunto completo de modelos que conforman al clasificador.

- **Reducir el costo computacional calculando los parámetros de modelos similares sólo una vez.**

  En el proceso del aprendizaje, la evolución del algoritmo provoca que la diversidad de los individuos vaya disminuyendo por lo que individuos completos o modelos espécificos aprendidos en la generación anterior pueden ser utilizados para no realizar cálculos innecesarios. Por otro lado, un individuo que sufrió pocos cambios a traves de los operadores de cruce y mutación, solo requiere aprendizaje de parámetros en los agrupamientos modificados.

- **Evaluar el comportamiento de porcentajes de mutación adaptivos en el proceso evolutivo.**

  En los experimentos realizados se observó que existe una relación entre la evolución del número de estados y la búsqueda del mejor agrupamiento. Por ejemplo se observó que la reducción abrupta del número de estados limita la búsqueda en el espacio de agrupaciones posibles. Por lo que para resolver esta problemática se requiere explorar porcentajes adaptivos de mutación y cruce.

- **Realizar pruebas con datos en donde se conoce que el CBSD básico falla.**

  Es importante identificar o generar un conjunto de datos de prueba en los cuales se tenga el conocimiento de las dependencias existentes en los datos, así como de atributos irrelevantes. Esta información permitirá evaluar de mejor manera el comportamiento del algoritmo.

- **Estudiar la efectividad de distintas funciones objetivo.**

  En la literatura de aprendizaje de RBs se han planteado una gran diversidad de métricas para evaluar una red. Es importante evaluar el funcionamiento de cada una de ellas para fines de clasificación en un contexto evolutivo.

VII.4 Conclusiones

Un enfoque evolutivo ha sido propuesto para resolver el problema de aprendizaje estructural en el diseñõ de un CBSD. El diseño de la mejor red es modelado como un problema de optimización que mide la exactitud de clasificación ponderado por la complejidad de la red resultante.

Para diseñar el algoritmo se propone una variante de la representación basada en grupos, la cual favorece la búsqueda de la mejor asociación de atributos. El operador de mutación permite la búsqueda entre las diferentes agrupaciones posibles, mientras que el operador de mutación permite detectar atributos irrelevantes. Para el número de estados se propone una representación binaria.

La red resultante es probada con datos generados desde nueve ademanes. Las evaluaciones experimentales muestran que los modelos obtenidos con el enfoque evolutivo propuesto son mejores de manera significativa. El algoritmo evolutivo genera redes con estructuras sencillas, detecta dependencia entre atributos, así como atributos irrelevantes, además de determinar el mejor número de estados para el nodo clase oculto. Este trabajo representa un pequeño primer paso hacia el diseño evolutivo de CBSDs.

Bibliografía

Avilés-Arriaga, H. H. 2006. "Reconocimiento visual de ademanes aplicado a robots móviles". Tesis de Doctorado, Tecnológico de Monterrey campus Cuernavaca, México. 112 pp.

Avilés-Arriaga, H. H., L. E. Sucar, y C. E. Mendoza 2006. "Visual recognition of similar gestures". En: "ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition", 20-24 agosto, Washington, DC, USA. IEEE Computer Society. 1100-1103 p.

Avilés-Arriaga, H. H., L. E. Sucar, C. E. Mendoza, y B. Vargas 2003. "Visual recognition of gestures using dynamic naive bayesian classifiers". En: "The 12th IEEE International Workshop on Robot and Human Interactive Communication, ROMAN 2003", Washington, DC, USA. IEEE Computer Society. 133- 138 p.

Baratoff, G. y D. Searles 2006. "Gesture recognition". Internet. htt://www.hitl.washington.edu/scivw/EVE/I.D.2.b.GestureRecognition.html, 15 de noviembre.

Cameron, P. J. 1994. "Combinatorics: topics, techniques, algorithms". Cambridge University Press, Primera edición, Cambridge, U.K. 355 pp.

Cheng, J. y R. Greiner 1999. "Comparing bayesian network classifiers". En: "Kathryn Laskey, Henri Prade. Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)", San Francisco, CA. Morgan Kaufmann. 101-108 p.

Chickering, D. M. 1995. "Learning Bayesian networks is NP-Complete". En: Fisher, D. y H. Lenz, editores, "Learning from Data: Artificial Intelligence and Statistics V". Springer-Verlag, Berlin, Alemania, 121–130 p.

Choudhury, T., J. M. Rehg, V. Pavlovic, y A. Pentland 2002. "Boosting and structure learning in dynamic bayesian networks for audio-visual speaker detection". En: "ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3", Washington, DC, USA. IEEE Computer Society. 30789 pp.

Cooper, G. F. y E. Herskovits 1992. "A bayesian method for the induction of probabilistic networks from data". Machine Learning, 9(4):309–347 p.

Cover, T. M. y J. A. Thomas 1991. "Elements of information theory". Wiley-Interscience, Segunda edición, New York, NY, USA. 542 pp.

Dagum, P. y M. Luby 1993. "Approximating probabilistic inference in bayesian belief networks is np-hard". Artificial Intelligence, 60(1):141–153 p.

Davis, J. W. y A. F. Bobick 1997. "The representation and recognition of human movement using temporal templates". En: "CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)", Washington, DC, USA. IEEE Computer Society. 928 pp.

Dempster, A. P., N. M. Laird, y D. B. Rubin 1977. "Maximum likelihood from incomplete data via the em algorithm". Journal of the Royal Statistical Society, 39:1-39 p.

Devore, J. L. 1995. "Probability and statistics for engineering and the sciences". Wadsworth, Inc, Cuarta edición, Belmont, CA, USA. 743 pp.

Diaz-de-Leon, R. y L. E. Sucar 2002. "Recognition of continuous activities". En: Garijo, F. J., J. C. R. Santos, y M. Toro, editores, "IBERAMIA 2002: Proceedings of the 8th Ibero-American Conference on AI", London, UK. Springer-Verlag. 875-881 p.

Duda, R. O., P. E. Hart, y D. G. Strork 2001. "Pattern classification". John Wiley & Sons, Segunda edición, New York, USA. 635 pp.

Eiben, A. E. y J. E. Smith 2003. "Introduction to evolutionary computing". Springer-Verlag, Primera edición, Berlin, Germany. 300 pp.

Falkenauer, E. 1994. "A new representation and operators for genetic algorithms applied to grouping problems". Evolutionary computation. 144 pp.

Fogel, L. J., A. J. Owens, y M. J. Walsh 1966. "Artificial intelligence through simulated evolution". John Wiley, New York, USA. 170 pp.

Friedman, N. 1998. "The bayesian structural EM algorithm". En: Cooper, G. F. y S. Mora, editores, "Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)", San Francisco, CA. Morgan Kaufmann. 129-138 p.

Friedman, N., D. Geiger, y M. Goldszmidt 1997. "Bayesian networks classifiers". Machine learning, 29(2-3):131-163 p.

Friedman, N., K. Murphy, y S. Russell 1998. "Learning the structure of dynamic probabilistic networks". En: Cooper, G. F. y S. Mora, editores, "Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)", San Francisco, CA. Morgan Kaufmann. 139-147 p.

Geiger, D. y D. Heckerman 1996. "Knowledge representation and inference in similarity networks and bayesian multinets". Artificial Intelligence, 82(1-2):45–74 p.

Gen, M. y R. Cheng 2000. "Genetic algorithms and engineering optimization". Wiley Series in Engineering Design and Automation. John Wiley & Sons, Primera edición, New York, USA. 410 pp.

Heckerman, D. 1995. "A tutorial on learning bayesian networks". Reporte Técnico MSR-TR-95-06, Microsoft research report. 57 pp.

Heckerman, D. y E. Horvitz 1998. "Inferring informational goals from free-text queries: A bayesian approach". En: Cooper, G. F. y S. Moral, editores, "Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98)", Madison, WI. Morgan Kaufmann. 230-237 p.

Heckerman, D. y M. P. Wellman 1995. "Bayesian networks". Communications of the ACM, 38(3):27-30 p.

Holland, J. H. 1975. "Adaptation in natural and artificial systems". University of Michigan Press, Primera edición, Ann Arbor. 228 pp.

Jensen, F. V. 1997. "An introduction to bayesian networks". Springer, Primera edición, Berlin, Germany. 178 pp.

Jordan, M. I., Z. Ghahramani, T. S. Jaakkola, y L. K. Saul 1998. "An introduction to variational methods for graphical models". En: Jordan, M. I., editor, "Proceedings of the NATO Advanced Study Institute on Learning in graphical models", Norwell, MA, USA. Kluwer Academic Publishers. 105–161 p.

Kjærulff, U. B. y A. L. Madsen 2005. "Probabilistic networks-an introduction to bayesian networks and influence diagrams". Aalborg University and HUGIN Expert A/S. 133 pp.

Kolsh, M. 2004. "Vision based hand gesture interfaces for wearable computing and virtual environments". Tesis de Doctorado, University of California, Santa Barbara, United States. 184 pp.

Koza, J. R. 1992. "Genetic programming". MIT Press, Cambridge, MA. 840 pp.

Kwoh, C.-K. y D. F. Gillies 1996. "Using hidden nodes in bayesian networks". Artificial Intelligence, 88(1-2):1-38 p.

Lam, W. y F. Bacchus 1994. "Learning bayesian belief networks: An approach based on the MDL principle". Computational Intelligence, 10(4):269-293 p.

Larrañaga, P. y M. Poza 1996. "Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters". IEEE Journal on Pattern Analysis and Machine Intelligence, 18(9):912–926 p.

Levinson, S. E., L. R. Rabiner, y M. M. Sondhi 1983. "An introduction to the application of the theory of probabilistic function of a markov process to automatic speech recognition". Bell sytems and technical journal, 62(4):1035-1074 p.

López, C. 2005. "Clasificadores por redes bayesianas". Tesis de Maestría, Universidad de Puerto Rico, Puerto Rico. 91 pp.

Martínez, M. 2006. "Aprendizaje de clasificadores bayesianos estáticos y dinámicos". Tesis de Doctorado, Tecnológico de Monterrey campus Cuernavaca, México. 91 pp.

Michalewicz, Z. 1996. "Genetic algorithms + data structures = evolution programs". Springer-Verlag, Tercera edición, London, UK. 340 pp.

Michie, D., D. J. Spiegelhalter, C. C. Taylor, y J. Campbell, editores 1994. "Machine learning, neural and statistical classification". Ellis Horwood, Upper Saddle River, NJ, USA. 290 pp.

Minka, T. 2001. "A family of algorithms for approximate bayesian inference". Tesis de Doctorado, Massachusetts Institute of Technology. 75 pp.

Murphy, K. P. 2002. "Dynamic bayesian networks: Representation, inference and learning". Tesis de Doctorado, University of California, Berkeley, California, USA. 212 pp.

Myers, J. W., K. B. Laskey, y K. A. DeJong 1999. "Learning bayesian networks from incomplete data using evolutionary algorithms". En: Banzhaf, W., J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, y R. E. Smith, editores, "Proceedings of the Genetic and Evolutionary Computation Conference", Orlando, Florida, USA. Morgan Kaufmann. 458–465 p.

Neal, R. M. 1993. "Probabilistic inference using Markov chain Monte Carlo methods". Reporte Técnico CRG-TR-93-1, University of Toronto. 140 pp.

Neapolitan, R. E. 1990. "Probabilistic reasoning in expert systems: theory and algorithms". John Wiley & Sons, Primera edición, New York, NY, USA. 433 pp.

Papoullis, A. y S. U. Pillai 2002. "Probability, random variables and stochastic processes". Mc Graw Hill, Cuarta edición, New York, NY, USA. 852 pp.

Pavlovic, V. 1999. "Dynamic bayesian networks for information fusion with application to human-computer interfaces". Tesis de Doctorado, University of Illinois at Urbana-Champaign. 160 pp.

Pazzani, M. 1996. "Searching for dependencies in bayesian classifiers". En: Fisher, D. y H. Lenz, editores, "Learning from Data: Artificial Intelligence and Statistics V", Berlin, Alemania. Springer-Verlag. 239-248 p.

Pearl, J. 1988. "Probabilistic reasoning in intelligent systems: networks of plausible inference". Morgan Kaufmann Publishers Inc., Primera edición, San Francisco, CA, USA. 552 pp.

Rabiner, L. R. 1989. "A tutorial on hidden markov models and selected applications in speech recognition". En: Trew, R. J., editor, "Readings in Speech Recognition, IEEE Proceedings", San Francisco, CA, volume 77. Morgan Kaufmann Publishers. 257-286 p.

Ramamoorthy, A., N. Vaswani, S. Chaudhury, y S. Banerjee 2003. "Recognition of dynamic hand gestures.". Pattern Recognition, 36(9):2069-2081 p.

Rechenberg, I. 1973. "Evolutionstrategie: Optimieriung technisher systeme nach prinzipien der biologisten evolution". Frommann-Holzboog, Stuggart, Germany.

Rish, I. 2001. "An empirical study of the naive bayes classifier". Reporte Técnico RC 22230, IBM research report. 6 pp.

Robinson, R. W. 1977. "Counting unlabeled acyclic digraphs". En: Little, C. H. C., editor, "Combinatorial Mathematics V". Springer Berlin / Heidelberg. 28-43 p.

Ross, B. J. y E. Zuviria 2007. "Evolving dynamic bayesian networks with multi-objective genetic algorithms". Applied Intelligence, 26(1):13–23 p.

Russell, S. J. y P. Norvig 2003. "Artificial intelligence. a modern approach". Pearson Education Inc., Upper Saddle River, New Jersey. 552 pp.

Sahami, M. 1996. "Learning limited dependence bayesian classifiers". En: "Second International Conference on Knowledge Discovery in Databases", Menlo Park, CA. AAAI Press. 335-338 p.

Schwarz, G. 1978. "Estimating the dimension of a model". The Annals of Statistics, 6(2):461–464 p.

Schwefel, H.-P. P. 1993. "Evolution and optimum seeking: The sixth generation". John Wiley & Sons, Inc., New York, NY, USA. 456 pp.

Stark, M., M. Kohler, y P. ZYKLOP 1995. "Video based gesture recognition for human computer interaction". Reporte Técnico 593/1995, Universität Dortmund, GERMANY.

Starner, T. 1995. "Visual recognition of american sign language using hidden markov models". Tesis de Maestría, Massachussets Institute of Technology, Cambridge. 52 pp.

Sucar, L. E., D. F. Gillies, y D. A. Gillies 1993. "Objective probabilities in expert systems". Artificial Intelligence, 61(2):187–208 p.

Sucar, L. E., D. F. Gillies, y D. A. Gillies 1994. "Probabilistic reasonig in higth-level vision". Image and vision computing, 12(1):42-60 p.

Wasson, G., D. Kortenkamp, y E. Huber 1998. "Integrating active perception with an autonomous robot architecture". En: Sycara, K. P. y M. Wooldridge, editores, "Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)", 9-13 de Mayo, New York. ACM Press. 325–331 p.

Wong, M. L., S. Y. Lee, y K. S. Leung 2002. "A hybrid approach to learn bayesian networks using evolutionary programming". En: Fogel, D. B., M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, y M. Shackleton, editores, "Proceedings of

the 2002 Congress on Evolutionary Computation, 2002", 12-17 de Mayo, Washington, DC, USA. IEEE Computer Society. 1314–1319 p.

Yannakakis, M. 1981. "Computing the minimum fill-in is NP-complete". SIAM Journal on Algebraic and Discrete Methods., 2:77-79 p.

Apéndice A

El sistema de reconocimiento visual de ademanes

En el trabajo presentado por Avilés-Arriaga (2006) se implementaron dos sistemas a través de los cuales se obtuvieron las muestras de ademanes, ambos sistemas siguen un proceso de estimación inicial de la posición del rostro y de la mano derecha de una persona para después seguir su movimiento.

El sistema No. 1

- Detección: Para detectar al usuario se implementó un algoritmo de detección de piel del rostro. La regla de Bayes para la clasificación es la siguiente (Jones y Rehg, 1996):

$$P(piel|rgb) = \frac{P(rgb|piel)P(piel)}{P(rgb|piel)P(piel) + P(rgb|\neg piel)P(\neg piel)} \tag{53}$$

  donde $P(piel|rgb)$ es la probabilidad de que un pixel sea piel dado que tiene un color rgb, $P(piel) = \frac{T_s}{T_s+T_n}$, $P(\neg piel) = \frac{T_n}{T_s+T_n}$, T_s y T_n son el número de muestras de piel y no-piel respectivamente tomadas de varias imágenes. De esta manera un pixel se puede clasificar como piel si $P(rgb|piel) \geq P(rgb|\neg piel)$. Los píxeles clasificados como piel son agrupados dentro de regiones homogéneas mediante una implementación del algoritmo de segmentación radial propuesto por el grupo SAVI.

  Para localizar el rostro, el algoritmo de segmentación se aplicó sobre la mitad

superior de la imagen, suponiendo que el rostro del usuario es la región de piel predominante en esta área. Una vez que el rostro ha sido localizado en la imagen, para detectar la mano derecha el algoritmo de segmentación se aplica sobre una región que contiene la mano, determinada mediante proporciones antropométricas y considerando que su brazo se encuentra inicialmente en una posición de descanso.

- Seguimiento: Una vez que la mano ha sido detectada, comienza el seguimiento a través de la secuencia de imágenes restante. En cada imagen se emplea una ventana de atención de 120x120 píxeles definida por la posición de la mano en la imagen anterior. El tamaño de la ventana de búsqueda es definida por una heurística de velocidad máxima producto de la observación de la velocidad natural de movimientos de diferentes usuarios. Con esta estrategia el seguimiento se ejecutó con una velocidad de hasta 30 imágenes por segundo.

- Ambiente de trabajo: El sistema se implementó en una computadora Silicon Graphics O2 con microprocesador R5000, 256Mb de memoria y sistema operativo IRIX versión 6.3. Para captura las imágenes se utilizó una videocámara Silicon Graphics modelo ZEYE. El lenguaje de programación fue ANSI C. La resolución de la imágenes fue de 640x480 píxeles. El sistema inicia con un usuario de pie frente a una videocaámara a una distancia fija entre 1.5m y 5m. La altura de la videocámra al suelo es de 97cm aproximadamente.

El sistema No. 2

- Detección: La deteccción del rostro del usuario se realizó por medio del algoritmo de Viola y Jones (2001) (implementado en OpenCV) que combina los resultados de localización de características del rostro como ojos, ceja y boca usando un esquema basado en el algoritmo AdaBoost (Freund y Sahpire, 1999). Con proporciones antropométricas basadas en el rostro del usuario, se estima una región

donde es posible encontrar la mano derecha del usuario y el torso del usuario. Se extraen aleatoriamente el 50% del los píxeles del rostro del usuario y el 100% de los píxeles del torso para crear modelos particulares de piel y no-piel, respectivamente. Estas muestras, mediante la unión de verosimilitud independiente se fusionan con modelos generales de piel y no-piel construidos fuera de línea. Los modelos generales fueron obtenidos de la fusión de las distribuciones generadas por las muestras T_s y T_n tomadas en el sistema No. 1 y No.2. La fusión fue llevada a cabo por medio de la unión lineal de opinión (Stone, 1961). La clasificación de los píxeles de piel se ejecuta sobre la región estimada de la posición de la mano para segmentarla.

- Seguimiento: Una vez localizada la mano, comienza el seguimiento del movimiento con el algoritmo CAMSHIFT implementado en las bibliotecas OpenCV. El algoritmo ajusta el tamaño de una ventana de segmentación sobre un a distribución de colores predeterminada.

- Ambiente de trabajo: Se implantó en una computadora personal IBM con microprocesador Intel Pentium a 1.6 Ghz, 512 Mb de meomoria RAM, una videocámara Sony modelo EVI-D30 y una tarjeta de video WinTV. El sistema operativo fue Linux Fedora Core 2. Los lenguajes de programación son ANSI C/C++. La resolución de las imágenes fue de 640x480 píxeles. Se supuso que al inicio del sistema una persona parada en una posición de descanso frente a la videocámara a una distancia entre 1.5m y 5m. El sistema fue probado en un ambiente de laboratorio con iluminación artificial constante y controlada. Se acondicionó un cuarto especial cerrado con cortinas para reducir la entrada de luz natural y un fondo azul para facilitar la detección y seguimiento de la persona. Cada participante utilizó una bata de laboratorio azul.