INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS CUERNAVACA

”Aprendizaje de Clasificadores Bayesianos Estáticos y Dinámicos”

Presenta:
Miriam Martínez Arroyo

Sometido al Programa de Graduados en Informática y Computación en cumplimiento parcial con los requerimientos para obtener el grado de:

Doctor en Ciencias Computacionales

Asesor:
Dr. Luis Enrique Sucar Succar

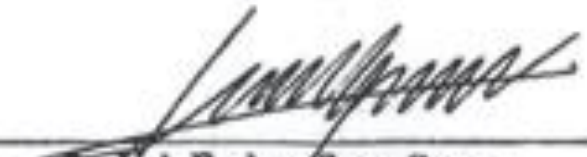
Cuernavaca, Morelos. Junio de 2007

Aprendizaje de Clasificadores Bayesianos Estáticos y Dinámicos

Presentada por:

Miriam Martínez Arroyo

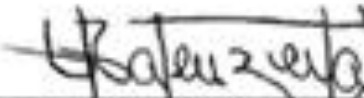
Aprobada por:



Dr. Luis Enrique Sucar Succar
Profesor-Investigador del INAOE, Puebla
Asesor

Dra. Mónica Larre Bolaños Cacho
Directora de la División Académica de Profesional y Posgrado
ITESM Campus Cuernavaca
Sinodal

Dr. Neil Hernández Gress
Director de la División de Posgrados e Investigación
ITESM Campus Estado de México
Sinodal



Dr. Manuel Valenzuela Rendón
Profesor-Investigador
ITESM Campus Monterrey
Sinodal

A Daira y Saúl

”Lo maravilloso de aprender algo,
es que nadie puede arrebatárnoslo.”
B.B. King

Agradecimientos

De manera muy especial le agradezco al Dr. Luis Enrique Sucar Succar por su invaluable apoyo profesional durante la realización de esta tesis, asimismo por los consejos emanados de su gran calidad humana.

A quienes fungieron como sinodales en mi examen profesional, los doctores: Mónica Larre, Neil Hernández y Manuel Valenzuela, por las observaciones realizadas que ayudaron a complementar éste trabajo.

Por las vivencias, angustias y logros compartidos, a mi compañero de estudios J. A. Montero. Porque considero que fue un gran apoyo y motivación para lograr la culminación de esta tesis. A su esposa Mary por los lonches y las oraciones recibidas.

Mas que una agradecimiento, un reconocimiento a mi familia (Pete, Hugo, Lidia, Claudia y Karina), por el apoyo incondicional que siempre nos han brindado a mis hijos y a mí, ¡mil Gracias!

A Doña Clara a quien le ha tocado parte del cuidado de mis peques y porque se puedo seguir contar con ella. A la Mimí quien me apoyo durante los dos primeros años de mis estudios doctorales y quien me regaló su amistad.

A mis amigos Elias, Wences y Vales, quienes de alguna forma y en algún momento contribuyeron en mi estado de ánimo, para sacar adelante este trabajo. A Elisa mil gracias por el apoyo, por las porras recibidas y por tenerme presente en sus oraciones. A Caro con quien compartí "crisis existenciales" !gracias por escucharme! A Claudia, Areli y Any, mis vecinas de Cuernavaca, por los tiempos compartidos y el apoyo recibido.

Al Instituto Tecnológico de Acapulco, por el apoyo brindado para la realización de estos estudios.

Al pueblo de México, que a través del Sistema Nacional de Educación Superior Tecnológica (SNEST), de la Dirección General de Educación Superior Tecnológica (DGEST- antes DGIT) y del Consejo del Sistema Nacional de Educación Tecnológica (CoSNET), me brindó el apoyo económico para la realización de este doctorado.

Por último, pero no menos importante, agradezco infinitamente a mis hijos (Daira y Saúl) por todo el amor que he recibido, por la comprensión que a pesar de su corta edad me han brindado y por el tiempo que les he robado para dedicarselo a este trabajo.

Resumen

Aunque el clasificador bayesiano simple ha sido ampliamente utilizado debido a que es un modelo de clasificación eficiente, fácil de aprender y con gran exactitud en muchos dominios, este presenta dos principales desventajas: la exactitud de la clasificación disminuye cuando los atributos no son independientes, y no puede ocuparse de atributos continuos. Además de que existen otras consideraciones que afectan el proceso de aprendizaje, tales como trabajar con información incompleta o faltante, manejo de grandes cantidades de datos y/o variables, selección de atributos representativos al problema, entre otras. Un clasificador bayesiano simple puede representar dominios estáticos o puede también representar dominios dinámicos, considerar este aspecto complica aún mas el proceso de aprendizaje.

El objetivo, entoces, es proporcionar un método aprendizaje de clasificadores bayesianos simples que evite que la exactitud de la clasificación disminuya cuando los atributos no sean independientes, y se ocupe de atributos continuos no paramétrizados, además de considerar aspectos de selección de atributos relevantes y manejar información oculta, garantizando obtener una buena estructura y conservando la simplicidad del modelo o reduciendo la complejidad del mismo.

Proponemos dos nuevos métodos: aprendizaje de clasificadores bayesianos estáticos (ACBE) y aprendizaje de clasificadores bayesianos dinámicos (ACBD).

El método **ACBE** incluye cuatro etapas, *Inicialización, Discretización, Mejora estructural* y *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 MDL, donde el número de intervalos que minimiza el MDL se obtiene por cada atributo. Para tratar con atributos dependientes y atributos irrelevantes, aplicamos un método que elimina y/o une atributos, basado en medidas de información mutua condicional y evaluando la exactitud de la clasificación despues de cada operación.

El método **ACBD** incluye cinco etapas, *Inicialización, Discretización, Determinación del nodo clase oculto, Mejora estructural* y *Clasificación dinámica*. En método de *Discretización* es el mismo que para el ACBE; la etapa de *Mejora estructural* es similar, solo varia la evaluación de las estructuras resultantes, ya que se consideran como estructuras de árbol y se evaluan través de una *medida de calidad* basada en el principio MDL, la *Determinación del mejor número de estados* para el nodo clase oculto se basa en el algoritmo EM y las estructuras resultantes se evaluan con base a la *medidad de calidad*. Finalmente para la *Clasificación dinámica* se construye la red de transición mediante una técnica general de redes bayesianas dinámicas.

Los métodos se probaron en aplicaciones con datos reales obteniendo muy buenos resultados. El método estático se aplicó en el reconocimiento de piel (con un 98 % de exactitud) y en la detección de cancer cervical (94 % de exactitud), el modelo dinámico se aplicó en el reconocimiento de siete gestos, usando un modelo para cada gesto, los cuales son representados por un clasificador bayesiano dinámico que obtuvo en promedio 98 % de exactitud para los datos de prueba.

Índice general

Índice de figuras x

Índice de tablas xii

Índice de algoritmos xiii

1. Introducción 1

1.1. Motivación 1

1.2. Planteamiento del problema 2

1.3. Objetivos 3

1.4. Metodología 4

1.5. Resultados 6

1.6. Descripción del trabajo 6

2. Clasificadores Bayesianos 8

2.1. Introducción 8

2.2. Clasificador bayesiano 9

2.3. Tipos de clasificadores bayesianos 11

2.3.1. Clasificador bayesiano simple (BN) 11

2.3.2. Clasificador bayesiano simple aumentado a árbol (TAN) 12

2.3.3. Clasificador bayesiano simple aumentado a red (BAN) 12

2.3.4. Red bayesiana general (GNB) 13

2.3.5. Comparación de los clasificadores 13

2.4. Métodos de aprendizaje 14
2.4.1. Selección de atributos 15
2.4.2. Manejo dependencias 15
2.4.3. Manejo de información incompleta 18
2.4.4. Discretización 22
2.4.5. Otras consideraciones 22
2.5. Resumen 23

3. Discretización **24**
3.1. Introducción 24
3.2. Discretización en clasificadores bayesianos 25
3.2.1. Efecto de la discretización 26
3.3. Métodos de discretización 27
3.3.1. Metodos locales y globales 27
3.3.2. Métodos supervisados y no supervisados 27
3.3.3. Métodos univariantes y multivariantes 28
3.3.4. Métodos parámetricos y no parámetricos 28
3.4. Métodos de discretización para el NB 28
3.4.1. Métodos globales y no supervisados 29
3.4.2. Métodos locales y supervisados 29
3.5. Otros métodos de discretización 32
3.5.1. Discretización basada en el MDL 32
3.5.2. Discretización manejando *desviación y varianza* 32
3.5.3. Discretización 1RD 33
3.6. Resumen 34

4. Redes bayesianas dinámicas **35**
4.1. Redes bayesianas 35
4.1.1. Aprendizaje de redes bayesianas 36

4.2. Redes bayesianas dinámicas 37
4.2.1. Aprendizaje de redes bayesianas dinámicas 38
4.2.2. Métodos de aprendizaje basados en ajustes 39
4.2.3. Métodos basados en análisis de dependencias 39
4.3. Modelos Ocultos de Markov (HMMs) 41
4.4. Resumen 41

5. Aprendizaje de clasificadores bayesianos estáticos 42
5.1. Introducción 42
5.2. Metodología 43
5.2.1. Etapa I: Inicialización 43
5.2.2. Etapa II: Discretización 45
5.2.3. Etapa III: Mejora estructural 50
5.2.4. Etapa IV: Evaluación 50
5.3. Resumen 52

6. Aprendizaje de clasificadores bayesianos dinámicos 53
6.1. Introducción 53
6.2. Metodología 55
6.2.1. Etapa I: Inicialización 55
6.2.2. Etapa II: Discretización 58
6.2.3. Etapa III: Determinación del nodo clase oculto 58
6.2.4. Etapa IV: Mejora estructural 58
6.2.5. Etapa V: Evaluación Dinámica 60
6.3. Resumen 60

7. Pruebas y resultados 62
7.1. Clasificadores bayesianos estáticos 62
7.1.1. Metodología 62
7.1.2. Clasificación de piel 63

7.1.3. Clasificación de cáncer cervical 68
7.1.4. Análisis del método estático 73
7.2. Clasificadores bayesianos dinámicos 76
7.2.1. Metodología 76
7.2.2. Reconocimiento de gestos 76
7.2.3. Evaluación 78
7.2.4. Análisis método dinámico 81
7.3. Implementación 81
7.4. Limitaciones 82

8. Conclusiones y Trabajo futuro **83**
8.1. Resumen 83
8.2. Aportaciones 86
8.3. Limitaciones 86
8.4. Trabajo futuro 87

A. Complejidad Temporal **88**

B. Publicaciones originadas **90**

C. Glosario de términos **91**

Referencias **96**

Índice de figuras

2.1. Modelo gráfico de un clasificador bayesiano simple. 10

2.2. Estructura de un clasificador bayesiano simple. 11

2.3. Estructura de un clasificador TAN. 12

2.4. Estructura de un clasificador BAN 12

2.5. Estructura de un clasificador GNB. 13

2.6. (a) Estructura inicial de un clasificador bayesiano simple. (b) Estructura modificada, después de la eliminación del atributo E_2 y la unión de dos atributos: $E_1 y E_3$. 17

4.1. Diferentes estructuras de redes Bayesianas. (a) árboles, (b) Poliárboles y (c) Redes Multiconectadas. 36

4.2. Red bayesiana dinámica (a) Estructura inicial. (b) Red de transición. 38

4.3. Red bayesiana dinámica que evoluciona a través de cinco períodos de tiempo. 38

5.1. Modelo de aprendizaje de clasificadores bayesianos estáticos. Los recuadros marcan las principales etapas del modelo. 44

5.2. Etapa 1: Inicialización del clasificador bayesiano simple estático. 45

5.3. Etapa 2: Discretización del clasificador bayesiano simple estático. 45

5.4. Fase 3: Mejora estructural del clasificador bayesiano estático. 50

6.1. Clasificador bayesiano dinámico. 54

6.2. Modelo de aprendizaje de clasificadores bayesianos dinámicos. 56

6.3. Etapa I: inicialización del clasificador bayesiano dinámico. 57

6.4. Etapa II: Discretización del clasificador bayesiano dinámico. 58

6.5. Etapa III: Determinación del nodo clase oculto del clasificador bayesiano dinámico. 59

6.6. Etapa IV: Mejora estructural, clasificador bayesiano dinámico. 59

7.1. Muestras de la BD de piel 64

7.2. Muestras de la BD de no piel 64

7.3. Experimento 1: Segmentación usando los modelos: (a) RGB normalizado, (b) R-GYI y (c) R-GY, (d) imagen original. 66

7.4. Experimento 2: Segmentación usando los modelos: (a) RGB normalizado, (b) R-GYI y (c) R-GY, (d) imagen original. 66

7.5. Experimento 3: Segmentación usando los modelos: (a) RGB normalizado, (b) R-GYI y (c) R-GY, (d) imagen original. 67

7.6. Ejemplo de una imagen de colposcopia, donde hay 5 regiones (A, B, C, D, E) que son seleccionadas a partir de una secuencia de imagenes. 69

7.7. Ejemplo de la representación de las imágenes de colposcopía mediante el modelo 2. 69

7.8. Modelo de parábola para evidencia en una imagen de colposcopía. 70

7.9. Muestra de una imagen de colposcopia y la interface usada por el experto para etiquetar los casos de entrenamiento. 71

7.10. Ejecución de gestos. 77

7.11. Clasificador dinámico para el gesto *Abrir*. 80

Índice de tablas

2.1. Resultados de la clasificación del ejemplo 2.1. 10

2.2. Resultados en exactitud y desviación estándar de cada clasificador, comparado con el mejor resultado conocido en la literatura [Cheng and Greiner, 1999]. 14

3.1. Resumen de métodos de discretización. 28

7.1. Atributos involucrados en el problema de reconocimiento de piel. 64

7.2. Resultados de la clasificación obtenida a partir de la *mejor* discretización probando con diferentes valores de α. 65

7.3. Proceso de *Mejora estructural* para la obtención del modelo de color en el reconocimiento de Piel. 65

7.4. Clasificador bayesiano usado para determinar la clase piel en diferentes espacios de color. . . 67

7.5. Comparación del nuevo modelo para reconocimiento de piel con otros clasificadores. 67

7.6. Atributos involucrados en el problema de cáncer cervical. 70

7.7. Resultados de la discretización con diferentes α. 72

7.8. Clasificación de cáncer cervico uterino. 73

7.9. Comparación del nuevo modelo para la apliación de Cáncer cervical con otros clasificadores. 73

7.10. Tiempo de aprendizaje para el modelo estático. 74

7.11. Reconocimiento del gesto *abrir*. 78

7.12. Reconocimiento del gesto *mouse*. 79

7.13. Reconocimiento del gesto *teléfono*. 79

7.14. Porcentajes de reconocimiento para todos los gestos. 80

7.15. Tiempo de aprendizaje para el modelo dinámico. 81

Índice de algoritmos

2.1. Algoritmo FSSJ. 17

2.2. Algoritmo BSEJ. 18

2.3. Algoritmo EM. 21

2.4. Algoritmo NBE. 22

3.1. Algoritmo Iterativo. 31

4.1. Algoritmo de aprendizaje de relaciones temporales (ART). 40

5.1. Algoritmo ACBE (Aprendizaje de Clasificadores Bayesianos Éstáticos.) 44

5.2. Algoritmo Inicialización ACBE. 45

5.3. Discretiza ACBE 49

5.4. Poda ACBE 51

6.1. ACBD (Aprendizaje de Clasificadores Bayesianos Dinámicos) 57

6.2. Etapa I: Inicialización del método ACBD. 57

6.3. Etapa III Determinación del nodo clase oculto), método ACBD. 58

6.4. Etapa IV:Mejora estructural del método ACBD. 61

Capítulo 1

Introducción

Un clasificador, en general, suministra una función que mapea (clasifica) un dato (instancia) en una o diferentes clases predefinidas [Cheng and Greiner, 1999]. Existen diversos métodos de clasificación, por ejemplo:

- Bayesianos: NB (*Naive Bayes*), TAN, BAN y GNB [Cheng and Greiner, 1999].
- Árboles de decisión: ID3, C4.5 [Quinlan, 1993].
- Reglas:1RD [Holte, 1993], tablas de decisión.

En especial los clasificadores bayesianos simples son ampliamente utilizados debido a que presentan ciertas ventajas:

- Generalmente son fáciles de construir y de entender
- Las inducciones de estos clasificadores son extremadamente rápidas, requiriendo solo un paso para hacerlo.
- Es muy robusto considerando atributos irrelevantes. Toma evidencia de muchos atributos para realizar la predicción final.

1.1. Motivación

Los clasificadores bayesianos se han aplicado en diversas áreas. Sin embargo, el clasificador bayesiano simple tiene dos principales desventajas:

- La exactitud de la clasificación disminuye cuando los atributos no son independientes, y
- No puede ocuparse de atributos continuos no paramétrizados.

El clasificador bayesiano es un tipo de estructura de red bayesiana, y en este contexto lo podemos ubicar para hablar del aprendizaje automático de redes bayesianas en general, donde el principal problema es utilizar un método que garantice obtener la mejor estructura.

El obtener una red bayesiana a partir de datos, es un proceso de aprendizaje que se divide en dos etapas: el aprendizaje estructural y el aprendizaje paramétrico [Pearl,1988; Hernández O.J. et al, 2004]. La primera de ellas, consiste en obtener la estructura de la red bayesiana, es decir, las relaciones de dependencia e independencia entre las variables involucradas. La segunda etapa, tiene como finalidad obtener las probabilidades a priori y condicionales requeridas a partir de una estructura dada.

Aunque diferentes métodos de aprendizaje han obtenido buenos resultados en la práctica, no se considera a uno como el mejor o el óptimo, ya que no garantizan obtener la estructura optima. Existen otras consideraciones que hacen que el proceso de aprendizaje se dificulte aún mías, tales como manejo de grandes volúmenes de datos, manejo de información incompleta y/o faltante, descubrimiento de variables ocultas, discretización de variables continúas y eliminación de variables irrelevantes al problema, entre otros. Otro aspecto es el de considerar el aprendizaje en ambientes dinámicos para modelar la evolución de las variables a través del tiempo, tomando en cuenta que la mayoría de los fenómenos están en constante evolución.

1.2. Planteamiento del problema

El problema a tratar en esta investigación se encuentra en el dominio de aprendizaje de clasificadores bayesianos simples, enfocado hacia el aprendizaje estructural y parámetrico de clasificadores bayesianos estáticos y dinámicos. En este proceso de aprendizaje existen diversos puntos que considerar, como son:

- **Selección de atributos**: cuando se trata de grandes volúmenes de información, es dificil aplicar algoritmos tradicionales por diversas inconveniencias (implican muchos ciclos, tiempo de procesamiento, mayor costo, etc). Una forma de reducir el problema lo constituye la clasificación o selección de datos. Esta selección permite enfocar la búsqueda en subconjuntos de variables y/o muestras de datos en donde realizar el proceso de aprendizaje. Sin embargo, la clasificación no es un proceso trivial ya que debe identificar un conjunto de categorías o clases que describan al conjunto de datos.
- **Manejo de atributos dependientes**: un aspecto importante en el aprendizaje es encontrar un modelo que represente el dominio del conocimiento, capturando la dependencia entre las variables involucradas en el fenómeno (particularmente cuando se desea predecir el comportamiento de algunas variables desconocidas basados en otras conocidas) buscando simplificar la representación del conocimiento (menos parámetros) y el razonamiento (propagación de las probabilidades).
- **Manejo de atributos continuos**: los datos con los cuales se representa un fenómeno si son continuos se discretizan y aunque existen diferentes técnicas para ello, se debe contar con un método que contribuya a mejorar el proceso de aprendizaje.
- **Manejo de variables ocultas**: existen diversos métodos de aprendizaje que han asumido que los datos son completos, pero existen muchos casos en los cuales la complejidad del problema,

el tamaño del mismo y la disponibilidad del experto dificultan la obtención de los datos ya que puede resultar en un proceso extenso y costoso. La dificultad que se presenta es que cuando no conocemos todos los datos podemos ignorar algunas variables relevantes que pueden influir en la resolución del problema o podemos tener datos parcialmente observados que necesitamos complementar. Además un punto de interés es que cuando hace falta información, no se tiene una métrica definida para evaluar las estructuras obtenidas durante el aprendizaje.

- Otro aspecto es el de considerar el aprendizaje en **ambientes dinámicos** para modelar la evolución de las variables a través del tiempo, tomando en cuenta que la mayoría de los fenómenos están en constante cambio.

Aunque diferentes métodos de aprendizaje sí resuelven algunas de estas dificultades es difícil encontrar un método que considere estos aspectos en conjunto y que proporcione una estructura simple y fácil de aprender. Existen diferentes tipos de clasificadores bayesianos ampliamente probados, como son el TAN, BAN y GNB [Cheng and Greiner, 1999] que obtienen buenos resultados en la exactitud de la clasificación y que consideran aspectos de dependencia entre variables, la inconveniencia que presentan es que estos incrementan la complejidad del aprendizaje ya que aprenden estructuras que involucran más conexiones entre las variables (capítulo 2), que el clasificador bayesiano simple.

El punto entonces, es contar con un método de aprendizaje que sea capaz de capturar la dependencia entre las variables involucradas en el fenómeno, considerando los aspectos antes mencionados y a la vez manteniendo la simplicidad del modelo. En este trabajo se propone una metodología de aprendizaje de clasificadores bayesianos simples considerando lo anterior y cuyos objetivos se detallan a continuación.

1.3. Objetivos

Objetivo general

- Desarrollar dos métodos para el aprendizaje de clasificadores bayesianos simples, uno para aplicaciones estáticas y otro para aplicaciones dinámicas.

Objetivos particulares

Implementar un método para aprendizaje de clasificadores bayesianos simples estáticos, que permita:

- Determinar cuál es la mejor forma de dividir en intervalos la variable continua, de tal manera que maximice el rendimiento del clasificador.
- Mejorar la estructura conservando la simplicidad de la misma mediante las operaciones de eliminación y unión de variables, considerando eliminar variables que no son relevante y la unión o eliminación de variables que son condicionalmente dependientes.

Implementar un método para aprendizaje de clasificadores bayesianos simples dinámicos, que nos permita, además de lo anterior, lo siguiente:

- Determinar el número de estados convenientes para los nodos clases ocultos.
- Que se realice el aprendizaje parámetrico de la estructura completa, considerando el nodo clase oculto, las variables discretas y continuas incluidas originalmente en el modelo y las variables derivadas por unión.

Que en ambos casos se conserve la estructura básica, que nos permita considerar, mediante modelos sencillos, las principales relaciones entre los datos.

1.4. Metodología

El trabajo que se presenta es referente al aprendizaje de clasificadores bayesianos donde se incluye un método de aprendizaje estructural y parámetrico. El enfoque principal para esta investigación es aplicar un método que se ocupe de los problemas de manejo de variables continuas, selección de atributos y detección de dependencias entre variables además de que se ocupe del manejo de variables ocultas, y que permita conservar la simplicidad del modelo.

Proponemos dos métodos: aprendizaje de clasificadores bayesianos estáticos (ACBE) y aprendizaje de clasificadores bayesianos dinámicos (ACBD).

Modelo estático

Este método considera (i) la discretización de variables continuas, (ii) la selección de atributos relevantes y (iii) la eliminación o combinación de atributos dependientesy al mismo tiempo preserva la estructura del clasificador bayesiano simple.

Este método obtiene:

- La estructura del clasificador.
- Un número de intervalos, para los atributos continuos, que contribuye a la mejora estructural.
- Los parámetros asociados (tablas de probabilidad condicional para cada atributo y probabilidades a priori para el nodo clase).

La técnica de discretización que utiliza este método está basada en el mejoramiento de la clasificación. La idea es aplicarla y evaluar el efecto que tiene la discretización en el proceso de aprendizaje, considerando la eliminación de atributos irrelevantes y/o la unión de dos variables en una, buscando una estructura que mejore la exactitud del clasificador.

El algoritmo básico consta de las siguientes etapas:

- Etapa I: Inicialización
- Etapa II: Discretización
- Etapa III: Mejora Estructural
- Etapa IV: Evaluación

Los pasos 2 y 3 son repetidos iterativamente hasta que la estructura-discretización no pueda ser mejorada.

Modelo dinámico

El método dinámico considera (i) la determinación del nodo clase oculto, (ii) la discretización de variables continuas, (iii) la selección de atributos relevantes, (iv) eliminación o combinación de atributos dependientes y (v) el proceso dinámico para prender un clasificador bayesiano simple.

El método obtiene:

- La estructura del clasificador.
- Un número de intervalos, para los atributos continuos, que contribuye a la mejora estructural.
- Un número de estados, para el nodo clase oculto, que contribuye a la mejora estructural.
- Los parámetros asociados (tablas de probabilidad condicional para cada atributo y probabilidades a priori para el nodo clase).

El proceso de discretización de variables continuas es el mismo proceso que en el método de aprendizaje que para el ACBE. La etapa de *Mejora estructural* es similar, cambia únicamente en la forma de evaluar las estructuras resultantes, ya que estas son consideradas como estructuras de árbol y evaluadas a través de una *medida de calidad* basada en el principio MDL (*Minimun Descrption Length*). La Determinación del número de estados para el nodo clase oculto se basa en el algoritmo EM (*Expectation-Maximization*), buscando encontrar una estructura que mejore la exactitud del clasificador. Se generan diferentes clasificadores que son evaluados como estructuras de árboles a través de una *medidad de calidad*. Finalmente para la *Clasificación dinámica* se construye la red de transición mediante una técnica general de RBD.

El algoritmo para este método consta de 5 etapas:

- Etapa I: Inicialización
- Etapa II: Discretización
- Etapa III: Determinación del nodo clase oculto
- Etapa IV: Mejora Estructural
- Etapa V: Evaluación Dinámica

Las etapas II, III y IV son repetidos iterativamente hasta la estructura no pueda ser mejorada.

1.5. Resultados

Los métodos se probaron en diferentes áreas obteniendo muy buenos resultados. El método estático se aplicó en el reconocimiento de piel con un 98 % de exactitud y en la detección de cáncer cervical 94 % de exactitud. El modelo dinámico se aplicó en el reconocimiento de gestos, usando un modelo para cada gesto, los cuales son representados por un clasificador bayesiano dinámico que obtiene una muy buena exactitud para los datos de prueba. Finalmente se evalúan los resultados del clasificador bayesiano dinámico aplicándolo al área de reconocimiento de gestos. El modelo se prueba clasificando 7 gestos (abrir, *mouse*, escribir, borrar, impresora, télefono y hojear), obteniendose un promedio de 98.25 % de reconocimiento de gestos.

1.6. Descripción del trabajo

Este trabajo esta conformado por 8 capítulos que se encuentran distribuidos de la siguiente manera:

- **Capítulo 2: Aprendizaje de clasificadores bayesianos.** Se describe el proceso de aprendizaje del clasificador bayesiano y trabajos relacionados, así como los conceptos básicos que servirán de soporte para los temas tratados en los siguientes capítulos. Se presenta un análisis de trabajos sobre mejora estructural y manejo de información incompleta. Se hace especial énfasis en el método de mejora estructural de [Pazzani, 1996] y el algoritmo EM, los cuales nos sirven de base para el método propuesto.
- **Capítulo 3: Discretización en clasificadores bayesianos.** En este capítulo se presenta una introducción sobre métodos de discretización de atributos continuos en el clasificador bayesiano simple, así como también se muestran trabajos del proceso de discretización antes y durante el aprendizaje y su efecto en este proceso.
- **Capítulo 4: Redes bayesianas dinámicas.** En este capíyulo se tratan algunos problemas de aprendizaje estructural en sistemas dinámicos donde se tienen variables relevantes que son parcialmente observadas o eventos completamente desconocidos y el problema de descubrir variables ocultas. Se describe el algoritmo EM, el cual es utilizado para nuestro trabajo doctoral.
- **Capítulo 5: Clasificador bayesiano simple estático óptimo.** Presentamos nuestro método de aprendizaje para un clasificador bayesiano estático, considerando un método de discretización y de mejora estructural.
- **Capítulo 6: Clasificador bayesiano simple dinámico óptimo.** En este capítulo exponemos un método para aplicarse a dominios dinámicos y con información incompleta (donde el nodo clase se deconoce). Describimos el modelo completo, que involocra, las etapas de: discretización, mejora estructural, determinación de información incompleta, cálculo del mejor número de estados para el nodo clase oculto (EM) y la clasificación dinámica.

- **Capítulo 7: Pruebas y resultados.** Se presentan los resultados de evalúar al clasificador estático aplicándolo a dos áreas: en clasificación de piel en imágenes, y detección de cáncer cervical. Para el primer caso, se evalúa la capacidad de reconocer pixeles de piel y no piel en imágenes. Para el segundo caso, consideramos el diagnóstico de cáncer basado en análisis de imágenes de colposcopia. Finalmente se evalúan los resultados del clasificador bayesiano dinámico aplicándolo al área de Reconocimiento de gestos, el modelo se prueba clasificando 9 gestos (abrir, *mouse*, escribir, etc.).
- **Capítulo 8: Conclusiones y trabajo futuro.** En este capítulo presentamos las conclusiones de esta tesis. Resumimos las principales contribuciones y discutimos algunos trabajos futuros en esta área.

Capítulo 2

Clasificadores Bayesianos

Una opción para el manejo de grandes bases de datos es restringir el espacio de hipótesis. Esto es útil ya que la complejidad de los algoritmos se debe principalmente a que capturan relaciones complejas de datos. Sin embargo, usando modelos más simples de aprendizaje, se pueden obtener también buenos resultados, como por ejemplo, el clasificador bayesiano simple, los árboles de decisión de un nivel y otros. Por otro lado existen muchas tareas de minería de datos y otras aplicaciones, que requieren la agrupación de datos en clases. Por ejemplo, en aplicaciones de préstamos bancarios, éstos pueden ser agrupados en clases de aceptaciones o rechazos. En este capítulo presentamos la definición formal del clasificador bayesiano, mostramos 4 tipos de clasificadores bayesianos y un trabajo comparativo de éstos. Así como algunos métodos de aprendizaje relacionados con los problemas detectados: selección de atributos, manejo de independencias y manejo de información incompleta.

2.1. Introducción

Un clasificador suministra una función que mapea (clasifica) un dato (instancia) en dos o más clases predefinidas [Fayyad et al., 1996]. La inducción automática del clasificador a partir de datos no sólo provee una clasificación que puede ser usada para mapear una nueva instancia en clases, sino que puede también proveer de una caracterización entendible a las clases. Algunos clasificadores son naturalmente más fáciles de interpretar que otros; por ejemplo los árboles de decisión [Quinlan, 1993] son fáciles de visualizar, mientras las redes neuronales son más difíciles. Los clasificadores bayesianos simples presentan algunas ventajas en su utilización [Cheng and Greiner, 1999]:

- Generalmente son fáciles de construir y de entender
- Las inducciones de estos clasificadores son extremadamente rápidas, requiriendo sólo un paso para entrenarlo.
- Son muy robustos considerando atributos irrelevantes, al considerar evidencia de muchos atributos para realizar la predicción final.

Los clasificadores bayesianos se han aplicado en diversas áreas, incluyendo genética [Ferrarri, 2005], detcción de cáncer [Ferrarri, 2006], [Birdwell et al., 2005], categorización de textos [Frasconi and Vullo., 2001], entre otras,obteniendo buenos resultados. En funcionamiento puede, incluso, tener una exactitud comparables a las redes bayesianas en contextos especificos [Lowd and Domingos, 2005]. Sin embargo, el clasificador bayesiano tiene 2 principales desventajas:

- La exactitud de la clasificación disminuye cuando los atributos no son independientes, y
- No puede ocuparse de atributos continuos no parámetricos (gaussianos).

Existen diferentes tipos de clasificadores bayesianos. En párrafos posteriores se presentan cuatro tipos de clasificadores: el clasificador bayesiano simple (BN), el clasificador bayesiano simple aumentado a árbol (TAN), el clasificador bayesiano simple aumentado a redes (BAN) bayesianas y la red bayesiana general (GBN).

2.2. Clasificador bayesiano

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:

$$P(H = h_i|E_1 = e_{1j}, E_2 = e_2, ..., E_n = e_{nj}) \tag{2.1}$$

Esto representa la probabilidad de H=h dado que se conoce que $E_i = e_1, ..., E_n = e_i$.

El clasificador bayesiano simple (*Naive Bayes Classifier -NBC-*) hace dos suposiciones:

- Que los atributos son condicionalmente independientes entre sí dado la clase,
- Que los atributos son discretos.

La probabilidad se puede obtener por el producto de las probabilidades condicionales individuales de cada atributo dado el nodo clase (ecuación 2.2). Esto hace que el número de parámetros se incremente linealmente con el número de atributos, en vez de en forma exponencial.

$$P(H = h_i|E_1, E_2, ..., E_n) = \frac{P(H = h_i)}{P(E_1, E_2, ..., E_n)} \prod_{j=i}^{n} P\left(E_j|H = h_i\right) \tag{2.2}$$

Gráficamente, un NBC se puede representar como una red bayesiana en forma de estrella [Pearl, 1988], con un nodo raíz, H, que corresponde a la variable de la clase, que está conectada con los atributos, $E_1, ..., E_n$. Los atributos son condicionalmente independientes dada la clase, de tal manera que no existen arcos entre ellas. Esta estructura se muestra en la figura 2.1.

Para determinar la clase mas probable se utiliza la regla de clasificación siguiente:

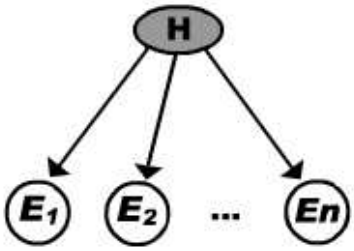


Figura 2.1: Modelo gráfico de un clasificador bayesiano simple.

$$H(e_1, e_2, ..., e_n) = argmax_{h \subset Estados(H)} P(h|e_1, e_2, ..., e_n) \tag{2.3}$$

Ejemplo 2.1: Dado que existe un nodo clase H y dos atributos A y B, todos con valores binarios. Tenemos los siguientes estados:

- C = Juega, NoJuega
- A=si, no
- B=si, no

Entonces si el clasificador bayesiano produce el siguiente resultado (tabla 2.1):

A	B	C
Si	Si	Juega
Si	No	NoJuega
No	Si	Juega
No	No	NoJuega

Tabla 2.1: Resultados de la clasificación del ejemplo 2.1.

Se asume que:

1. P(A = Si |Juega) P(B = Si | Juega)P(Juega) $>$ P(A = Si |NoJuega) P(B = Si | NoJuega) P(NoJuega)
2. P(A = Si |NoJuega) P(B = No|Nojuega)P(Nojuega) $>$ P(A = Si|Juega) P(B = No|Juega) P(Juega)
3. P(A = No|Juega) P(B = Si|Juega)P(Juega) $>$ P(A = No|NoJuega) P(B = Si|NoJuega) P(NoJuega)
4. P(A = No|NoJuega) P(B = No|NoJuega)P(NoJuega) $>$ P(A = No|Juega) P(B= No|Juega) P(Juega)

Lo cual puede ser generalizado para atributos no binarios.

Evaluación del clasificador

El clasificador bayesiano es utilizado para clasificar instancias $e_1, ..., e_n$ con clases verdaderas etiquetadas como $c_1, ..., c_n, e_1, ..., e_n$, y las clases resultantes son etiquetadas como: $\acute{c_1}, ..., \acute{c_n}$. Entonces el error de clasificación esta dado por:

$$|i \in 1, ..., n | C_i \neq \acute{C_i}|/n \tag{2.4}$$

2.3. Tipos de clasificadores bayesianos

Existen diferentes tipos de clasificadores bayesianos. En esta sección se describen cuatro tipos de clasificadores y una comparación de los mismos [Cheng and Greiner, 1999]:

1. Clasificador bayesiano simple (BN)
2. Clasificador bayesiano simple aumentado a árbol (TAN)
3. Clasificador bayesiano simple aumentado a red (BAN)
4. Red bayesiana general (GNB)

2.3.1. Clasificador bayesiano simple (BN)

Un clasificador bayesiano simple [Duda and Hart, 1963] es una estructura que tiene un solo nodo que es padre de todos los otros nodos (figura 2.2) [Cheng and Greiner, 1999], donde E_1, E_2, E_3, E_4 son los nodos hijos y H considerado como nodo padre, es el nodo a clasificar. En esta estructura no son permitidas otras conexiones entre los nodos hijos, por lo que se asume independencia entre éstos dado el nodo clase. En [Langley and Sage, 1994], [Jhon and Kohavi, 1997] y [Kononenko, 1991] podemos encontrar algoritmos de aprendizaje para esta estructura.

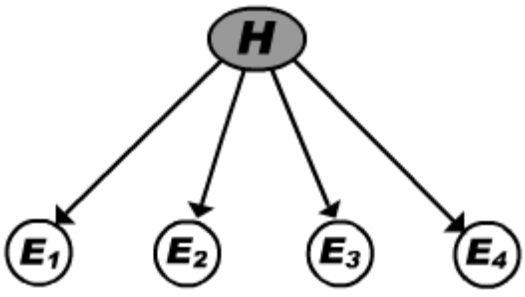


Figura 2.2: Estructura de un clasificador bayesiano simple.

2.3.2. Clasificador bayesiano simple aumentado a árbol (TAN)

El TAN crea la estructura de tal forma que un nodo clase es el padre de todos los otros nodos y después adiciona ligas entre nodos atributos en forma ordenada p.e. (E_1, E_2), (E_2, E_3), (E_3, E_4)..., con base a la información mutua condicional. La dirección de las ligas se determina probando en ambas direcciones, para poder determinar cual estructura maximiza la probabilidad (verosimilitud -"likelihood) de los datos [Friedman et al., 1997]. Una estructura TAN se muestra en la figura 2.3, donde se observa que la diferencia con el clasificador bayesiano simple es que en el TAN las ligas entre los atributos (E_1, E_2, E_3, E_4) forman un árbol.

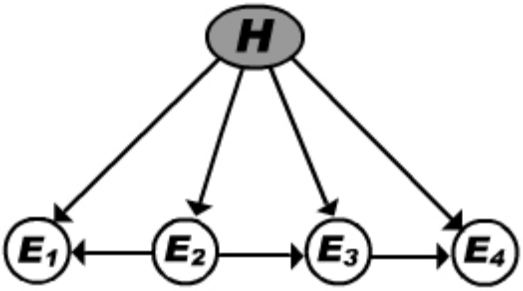


Figura 2.3: Estructura de un clasificador TAN.

2.3.3. Clasificador bayesiano simple aumentado a red (BAN)

El clasificador BAN extiende a TAN permitiendo que los atributos formen un grafo arbitrario, en vez de un árbol [Friedman et al., 1997]. El algoritmo de aprendizaje del BAN es igual al algoritmo del TAN, excepto que BAN usa un algoritmo de aprendizaje no restringido (puede formar una red multiconectada) en vez de uno restringido (que forma un árbol) como lo hace TAN. Una estructura BAN se muestra en la figura 2.4, donde se observa que la diferencia con el TAN consiste básicamente en que los atributos (E_1, E_2, E_3, E_4) forman una estructura de red, aunque el nodo clase sigue conectándose a todos los atributos. Un algoritmo de aprendizaje (CBL1) lo podemos ver en [Cheng et al., 1997].

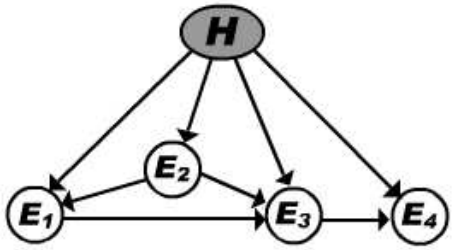


Figura 2.4: Estructura de un clasificador BAN

2.3.4. Red bayesiana general (GNB)

El algoritmo para aprendizaje de una GNB (*General Bayesian Network*) trata al nodo clasificado como un nodo ordinario, lo cual se puede observar en la figura 2.5, donde el GNB está conformado por una estructura de red, donde, el nodo a clasificar no necesariamente esta conectado a todos los atributos.

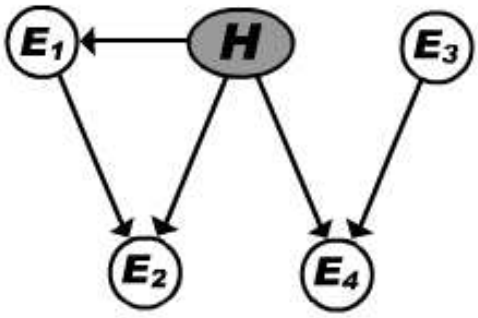


Figura 2.5: Estructura de un clasificador GNB.

2.3.5. Comparación de los clasificadores

[Cheng and Greiner, 1999] para probar el desempeño de los clasificadores, se implementaron los algoritmos de aprendizaje en el programa PowerConstructor 2.0 [Cheng, 1998], donde se ofrecen dos algoritmos de aprendizaje de redes bayesianas: uno para el caso donde el ordenamiento de nodos esta dado (algoritmo CBL1 [Cheng et al., 1997], donde se tomó al nodo clase como el primero en el ordenamiento y a los otros nodos fueron ordenados arbitrariamente. Para las pruebas se utilizaron 8 bases de datos (BD) extraídas del repositorio de UCI (Irvine: University of California), se prefirieron BD que tuvieran pocos o ningún atributo continuo. Cuando se requirió de discretizar atributos continuos, se utilizó el $MLC++$ [Kohavi, 1997]. En los experimentos se usaron 4 algoritmos para 4 clasificadores. Para probar los clasificadores con el conjunto de prueba se usó una versión modificada de JavaBayes v0.341.

Se midió el tiempo de aprendizaje de cada clasificador y se determinó que el clasificador bayesiano simple puede aprender en promedio en un tiempo menor a los demás. Así como también, que el número de atributos involucrados, influye en el aprendizaje de la estructura, ya que los clasificadores con menor número de atributos (BD con menos variables) arrojaron menor tiempo de aprendizaje.

En cuanto al resultado de la clasificación, podemos observarlo en la tabla 2.2, donde se muestran los resultados en exactitud y desviación estándar de cada clasificador. En esta tabla se puede observar que en general GNB, BAN y TAN son mejores que el NB, aunque en una prueba(BD DNA) el NB obtuvo mejor resultado. También podemos observar que el porcentaje de clasificación no es muy superior, siendo en la mayoria de los casos menor de 5 por ciento. BAN fue mejor en 4 de los conjuntos de prueba y que GNB y TAN, fueron mejores en 2 de las pruebas. En particular para el caso de las BD Guardería y Auto, GNB obtuvo resultados inferiores a NB ya que en estos casos los GNB fueron reducidos a NB con ligas faltantes (esta reducción a NB también es llamada NB selectivo), debido a requerimientos del experimento. Las estructuras (GNB, BAN y TAN) que representan mayor

BD	GBN	BAN	TAN	NB	**Mejor resultado públicado**
Adulto	**86.11** ±0,27(8/13)	85.82 ±0,27	86.01 ±0,27	84.18 ±0,29	**85.95**±0,27
Guardería	89.72 ±0,46(6/8)	**93.08** ±0,39	91.71 ±0,42	90.32±0,45	**N/A**
Hongos	99.30±0,16(5/22)	**100**	99.82 ±0,08	95.79 ±0,39	**100**
Ajedrez	**94.65**±0,69(19/36)	94.18 ±0,72	92.50 ±0,81	87.34 ±1,02	**99.53** ±0,21
DNA	79.09±1,18(43/60)	88.28 ±0,93	93.59 ±0,71	**94.27** ±0,68	**96.12**±0,6
Auto	86.11±1,46(5/6)	**94.04**±0,44	**94.10**±0,48	86.58±1,02	**N/A**
Destello	82.27±1,45(1*a*3/10)	82.85±2,00	**83.49**±1,99	80.11±3,14	**83.40**±1,67
Votaciones	95.17±1,89(10 a 11/16)	**95.63**±3,85	94.25±3,63	89.89±5,29	**96.3**±1,3

Tabla 2.2: Resultados en exactitud y desviación estándar de cada clasificador, comparado con el mejor resultado conocido en la literatura [Cheng and Greiner, 1999].

número de relaciones entre sus variables obtuvieron mejores resultados que el clasificador bayesiano simple, sin embargo esta mejora no fue muy superior al NB que obtuvo resultados aceptables.

También se concluyó que cuando se usan BAN o TAN las dependencias débiles entre las variables dado el nodo clase pueden ser fácilmente capturadas. Para el caso del GNB, debido a que el nodo clase es tratado como cualquier otro nodo, las dependencias débiles pueden no ser capturadas. Lo cual sugiere que tratar al nodo clase en forma diferente puede ser de gran utilidad, en algunos dominios, sobre todo en aquellos que impliquen un gran número de variables, para evitar suponer que todas éstas están altamente relacionadas al nodo clase.

La principal ventaja que se tiene al usar el clasificador bayesiano es que la tarea de aprendizaje se simplifica ya que se puede entender e implementar fácilmente, además de que realiza una buena clasificación considerando los atributos relevantes para representar el problema. Sin embargo en las pruebas realizadas ninguno de los clasificadores es aparentemente el mejor para todos las aplicaciones.

El clasificador bayesiano también es aplicado en aprendizaje no supervisado,para determinación de grupos, donde los grupos representan variables que no son observadas directamente (variables ocultas), a menudo el número de grupos son desconocidos y deben ser inferidos. Estos modelos son típicamente aprendidos usando el algoritmo EM [Dempster and Laird, 1977].

2.4. Métodos de aprendizaje

Para la construcción del clasificador bayesiano se han implementado diferentes algoritmos de aprendizaje, a continuación se presentan algunos trabajos clasificados de acuerdo a su aplicación a los difrentes problemas de aprendizaje: *Manejo de dependencias* (grandes bases de datos), *Selección de atributos, Variables ocultas e información incompleta* y *manejo de atributos continuos* (discretización).

2.4.1. Selección de atributos

APRI es un Sistema para Reconocimiento e Identificación Avanzado de Patrones (por sus siglas en inglés: Advanced Pattern Reconognition and Identification System) desarrollado por Ezawa y Nortón [Ezawa and Norton, 1995]. Este algoritmo es aplicado en el descubrimiento del conocimiento de fraudes y deudas incobrables en servicios de telecomunicaciones, básicamente abarca el problema del manejo de grandes volúmenes de datos.

APRI usa una máquina de aprendizaje supervisado con un método que construye modelos de red bayesiana. El método es capaz de predecir considerando algunos eventos con información incompleta y/o datos faltantes. Para probar el método, usan una base de datos real que tiene entre 4 a 6 millones de registros y de 600 a 800 millones de bytes. Estas bases de datos son consideradas grandes para investigar, pero realmente son pequeñas para la industria de las telecomunicaciones ya que manejan montos de información mucho más grandes. Las bases de datos utilizadas contienen detalles de llamadas telefónicas e información de los clientes. El problema principal se refiere a la clasificación incorrecta de los clientes que no pagan. Los datos son descritos por más de treinta variables, algunas discretas y otras continuas. Muchas de las variables discretas tienen grandes valores desordenados y las variables continuas no son normalmente distribuidas, además de que los valores faltantes son también muy comunes. APRI usa como concepto base el de entropía [1] [Kullback and Leiber, 1951], para realizar selecciones de dependencia. Primero selecciona un conjunto de variables y después un conjunto de dependencias entre el conjunto de variables seleccionadas usando una heurística conocida como información mutua mínima. APRI lee la base de datos de entrenamiento no más de 5 veces, en contraste con otros métodos como K2 [Cooper and Herskovits, 1992] en donde la base de datos puede leerse *O(n(u+1))* veces para crear un modelo (donde*n* es el número de nodos y *u* el número máximo de padres por nodo). APRI, construye modelos gráficos de probabilidad usando un proceso de 4 pasos, lo cual requiere de 3 entradas: una base de datos de casos de entrenamiento y dos parámetros T_{pf} y T_{ff} que controlan la densidad del modelo (el número de variables y ligas a incluir), cada uno entre 0 y 1. T_{pf} controla la variable seleccionada (indica si la variable actual, se incorpora a la estructura, el valor depende de la cantidad de variables que se deseen incluir, si el valor es cercano a 0 indica que se desean incluir pocas variables y si es cercano a 1 indica que se desean incluir la mayor parte de las variables). T_{ff} controla las ligas entre variable y variable (indica el porcentaje de ligas que se pueden incluir en la estructura) para el modelo final.

En general el método presenta varios puntos sobresalientes. Uno de los más importantes es que realiza un número constante de lecturas a disco y no un número lineal de veces, como lo hace K2 [Cooper and Herskovits, 1992] (que con 33 variables y 2 padres por nodo, necesita leer la bases de datos 99 veces). En aplicaciones de este tipo con datos de entrenamiento de millones de registros, leer pocas veces es esencial. Otro punto a favor es que realiza la clasificación de variables, sólo considera las más importantes y construye un clasificador como modelo de salida donde sólo se considera una variable a predecir.

2.4.2. Manejo dependencias

[Sucar and Gillies, 1993] proponen la metodología QUALQUANT (Orientación Cualitativa Cuantitativa) la cual es un mecanismo para construir y mejorar la estructura de una red bayesiana. En principio se obtiene un conjunto de reglas subjetivas basado en el conocimiento del experto, a

[1] Relación entre dos diferentes distribuciones definidas sobre el mismo espacio del evento. Ver sección 5.2.2

partir del cual se construye la estructura inicial de la red. Posteriormente, desarrollan un mecanismo basado en técnicas estadísticas para mejorar la estructura inicial de la red. Dicho mecanismo utiliza un conjunto de datos para validar las dependencias representadas por la red, para lo cual se calcula la correlación entre pares de 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. Entonces se tienen dos casos:

- Caso 1: si la correlación es menor a cierto valor predeterminado se puede suponer independencia y no se modifica la estructura.
- Caso 2: si la correlación es mayor a un valor predeterminado, entonces existe dependencia y se modifica la estructura.

En el caso 2, al modificar la estructura se podría introducir una nueva liga entre los nodos pero destruiría la estructura de árbol. Para resolver este problema en otro trabajo [Sucar et al., 1994] presentan 3 alternativas, aplicadas a un método de aprendizaje estructural en multi-árboles, donde la estructura inicial se descompone en diferentes poli-árboles, y cada uno representa el conocimiento de un objeto. La estructura de cada sub-árbol implica que se tiene un nodo padre (objeto a reconocer) y nodos hijos (atributos que lo caracterizan, es decir la estructura de un clasificador bayesiano simple). La estructura de cada sub-árbol implica que todos los atributos hijos son condicionalmente independientes dado el padre. El método comienza con esta suposición, posteriormente mediante pruebas de independencias entre pares de atributos, se puede modificar, considerando las tres alternativas siguientes:

1. Eliminar uno de los dos nodos ya que la información que aportan ambos nodos es casi la misma que la información que aporta un solo nodo, por lo tanto se elimina uno de ellos.
2. Combinar los dos nodos para considerar la información de ambos nodos en uno solo.
3. Crear un nodo auxiliar que sea padre de los nodos en cuestión.

Con el objeto de validar la suposición de independencia se utiliza el coeficiente de correlación a partir de un conjunto de datos disponibles. Las limitaciones que presenta este enfoque son: que no siempre se tiene un experto disponible para obtenr y revisar la estructura de la red y que el método se encuentra limitado a multi-árboles y a dependencia entre pares de variables.

Un trabajo posterior, desarrollado por [Gillies et al., 1996] realiza la creación de nodos ocultos entre variables que en un principio se consideran independientes por tener un mismo padre. Se crean nodos ocultos (en casos donde se encuentra alguna dependencia entre variables) que modelen los efectos de esas dependencias. Para la obtención de la matriz de probabilidades de esos nodos se utiliza un método de gradiente descendiente. Posteriormente se utilizan los métodos de propagación *forward* y *backward* para recalcular las probabilidades del nodo. Cuando existen padres que tienen más de dos hijos se utiliza el método de creación simple *nodo-a-nodo*. Se crean las variables ocultas en forma secuencial comenzando con el dato más dependiente. Aunque este método incorpora nodos ocultos no cuenta un procedimiento para determinar el número de estados requeridos para esos nodos.

Por otro lado, [Pazzani, 1996], presenta un modelo (método basado en ajustes) que busca la optimización global, donde sugiere que los atributos utilizados en una BD común no son condicionalmente independientes de la clase y que la violación de las suposiciones de independencias que afectan la exactitud del clasificador pueden ser detectadas a partir de datos. Por ejemplo en la figura 2.6a se

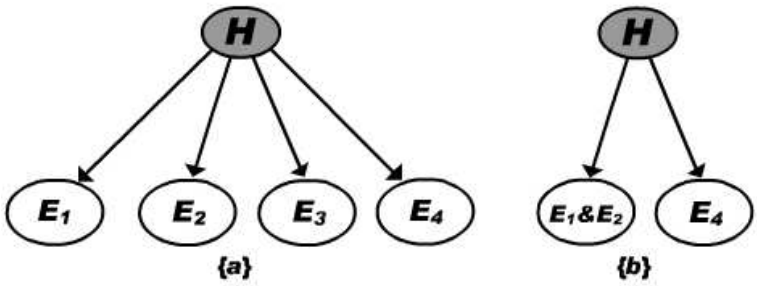


Figura 2.6: (a) Estructura inicial de un clasificador bayesiano simple. (b) Estructura modificada, después de la eliminación del atributo E_2 y la unión de dos atributos: $E_1 y E_3$.

Algoritmo 2.1 Algoritmo FSSJ.

1. Iniciar el conjunto de variables a utilizar a vacío.
   Clasificar todos los ejemplos en la clase más frecuente
2. Repetir en cada paso la mejor operación entre:
   - Considerar cada variable no usada como una nueva variable a incluir en el modelo, condicionalmente independiente de las variables ya incluidas, dada la variable a clasificar.
   - Juntar cada variable no utilizada con un variable ya incluida en el clasificador.
3. Evaluar cada clasificador candidato por medio de LOOCV (Leave One Out Cross Validation). Este procedimiento iterativo se repite N veces para n instancias (casos), dejando cada vez un caso fuera para probar y el resto para entrenar. El promedio de error de las pruebas una vez concluido los n experimentos es la taza de error estimado.
4. Repite 1,2,3 hasta que ninguna operación produzca mejoras.

tiene un dominio de 4 variables E_1, E_2, E_3 y E_4 y una variable a predecir H. Donde se supone que la variable E_2 no es relevante para H, y que las variables E_1 y E_3 son condicionalmente dependientes dado H. Por lo que el modelo quedaría tal como se observa en la estructura de la figura 2.6b.

Para determinar que variable no son relevantes y que variables pueden agruparse, Pazzani propone dos algoritmos (basados en la filosofía estadística de modelización hacia delante y modelización hacia atrás): Algoritmo FSSJ *(Forward Sequential Selection and Joining)* y el Algoritmo BSEJ *(Backward Sequential Elimination and Joining)*. Aunque las dependencias capturadas por el método al probar pares de variables hace más expresivo al modelo que el clasificador bayesiano simple, el algoritmo no considera dependencias entre tripletas de variables o más de ellas, lo que limita la expresividad del modelo generado. El algoritmo tampoco considera variables continuas, por lo que limita su aplicación a ciertos dominios. Los dos métodos principales propuestos por Pazzani se muestran en el algoritmo 2.1 y el algoritmo 2.2.

Un método mas reciente es el propuesto por [zhang et al., 2005], el HNB *(Hidden Naive Bayes)* crea un nodo padre oculto para cada par de atributos, donde se detectan dependencias. Se utiliza la información mutua condicional para estimador las relaciones de dependencia entre pares de varaibles y se asigna un peso de codependencia que determina que relaciones van a ser incluidas en el nuevo modelo, el cual aumenta a una estructura TAN.

Algoritmo 2.2 Algoritmo BSEJ.

1. Iniciar el conjunto de variables a utilizar a vacío. Clasificar todos los ejemplos en la clase más frecuente
2. Repetir en cada paso la mejor operación entre:
   a) Considerar reemplazar cada par de variables usadas por el clasificador como una nueva variable a incluir en el modelo, que conjunta el par de atributos.
   b) Considerar eliminar cada atributo usado por el clasificador.
   c) Evaluar cada clasificador candidato por medio de LOOCV
3. Repite 1,2 hasta que ninguna operación produzca mejoras.

Otra metodología para reducir la estructura del clasificador bayesiano, cuando existen muchos atributos, es relacionar las variables a partir sus valores taxonómicos y agrupandolos, tratando de generar clasificadores que sean sustancialmente más compactos y más exactos. El método propuesto es denominado AVT-NBL (*Attribute Value Taxonomies - Naive Bayes Learning*) y utiliza una función de evaluación, para las estructuras candidatas, basada en el principio MDL [Zhang et al., 2007].

2.4.3. Manejo de información incompleta

En aprendizaje a partir de datos se presentan algunos problemas ya que los fenómenos producidos en el mundo real son raramente observados completamente, lo cual nos lleva a ignorar variables relevantes que no son observadas a simple vista (ocultas) y que pueden influir en la resolución del problema, o a tener datos parcialmente observados que necesitamos complementar. Sin embargo, en la mayoría de las técnicas que se utilizan para esta tarea se ha asumido que los datos son completos; es decir, que los valores de todas las variables que participan en el evento son conocidas para todos los casos en la base de datos.

El manejo de información faltante en los algoritmos de aprendizaje estructural es muy importante ya que los datos omitidos pueden afectar en gran medida a los resultados. Al pasarlos por alto, o suponer que basta con excluirlos de los cálculos (como lo hacen algunos métodos de aprendizaje), se arriesga a obtener resultados no válidos o estructuras erróneas.

El manejo de información incompleta puede tratarse para dos casos:

1. Datos faltantes. Variables con información incompleta o datos parcialmente observados.
2. Información no observada a simple vista. Variables que pueden influir en el proceso de resolución del problema y que se encuentran ocultas.

El primer caso se origina debido a que pueden existir fenómenos que sólo se observan en forma parcial. También existen bases de datos que no han sido apropiadamente llenadas.

El segundo caso es debido a que en la mayoría de los fenómenos reales raramente observamos todas las variables que participan en el evento. Como en el área de medicina, donde por ejemplo, las enfermedades no son siempre conocidas hasta el final de un tratamiento y raramente se tienen resultados para todas las posibles pruebas clínicas. A menudo los modelos causales contienen variables que son sometidas a inferencia pero nunca observadas directamente, como es el caso de los *síndromes* en medicina [Binder et al., 1997]. Un caso común donde se utilizan variables o nodos ocultos es para modelar relaciones de independencia condicional entre variables.

Para ambos casos existen diversos trabajos previos, sin embargo la mayoría de estos se encuentran dirigidos hacia el aprendizaje parámetrico. También existen métodos que tratan los dos casos en conjunto. A continuación se describen algunos métodos relacionados con estos problemas.

- Datos faltantes

La forma más fácil de tratar datos faltantes es utilizar sólo los datos observados completamente, pero no es conveniente sobre todo cuando se trata de grandes cantidades de datos faltantes o cuando se trata de BD pequeñas. Otro método comúnmente utilizado es que la máquina de aprendizaje asigne a cada valor faltante un nuevo valor correspondiente a una categoría *desconocida*. Esto significa que serán agregados más parámetros para ser estimados y por tanto la complejidad de la estructura aumentara, además de que el valor extra no refleja el valor real faltante. Otro punto es que cuando se trata de BD con grandes cantidades de datos faltantes y se requiere realizar clasificación, el modelo puede estar basado en atributos con valores faltantes, lo cual no reflejaría la naturaleza del problema y no es deseable en un clasificador [Ghahramani and Jordan, 1994].

Una técnica muy socorrida también para manejo de datos incompletos es el reemplazar cada valor faltante por un valor (imputación simple) calculado por medio de algún método. Un método puede ser reemplazar el valor faltante por el valor promedio de los valores observados para ese atributo (imputación promedio) o muestrear con base a la distribución incondicional estimada a partir de los valores observados (imputación encubierta) [Little and Rubin, 1997]. Estos métodos podrían también no ser apropiados debido a que los valores pueden no ser los reales. Una mejor técnica la constituye el uso de múltiples métodos de imputación [Rubin, 1987]. Por su parte [Cooper and Herskovits, 1992] proponen un método para aprendizaje de la estructura y los parámetros de una red bayesiana cuando existen datos faltantes.

El algoritmo EM (Expectación-Maximización) es utilizado para tratar con información incompleta cuando se construyen modelos estadísticos [Little and Rubin, 1997] Este algoritmo ha sido utilizado en diversas investigaciones de aprendizaje de RB. Las primeras investigaciones para tratar datos incompletos con el algoritmo EM se dirigían al aprendizaje parámetrico de una estructura de red bayesiana fija [Lauritzen, 1995]. Otras investigaciones más recientemente se enfocan también a aprendizaje estructural. Por otro lado, una carencia detectada en esta tarea es que cuando hace falta información, no se cuenta con una medida para evaluar la estructura de la red. Algunas de las investigaciones que se han realizado para cubrir este punto son las realizadas por [Dempster and Laird, 1977] y [Friedman et al., 1997].

[Singh, 1994] propone un método para aprendizaje estructural y de probabilidades a partir de datos incompletos. El algoritmo propuesto es un método iterativo que usa una combinación de Expectación-Máximización y las técnicas de imputación para refinar iterativamente la estructura de la red y los parámetros de la manera siguiente:

1. Utiliza la estimación actual de la estructura y los datos incompletos para refinar las probabilidades condicionales
2. Coloca nuevos valores en los puntos donde aparecen valores faltantes por imputación encubierta a partir de la nueva estimación de las probabilidades condicionales.
3. Refina la estructura de la red a partir de la nueva estimación de los datos usando algoritmos estándares para aprendizaje de redes bayesianas a partir de datos completos.

[Friedman et al., 1997] realiza un trabajo sobre redes dinámicas con datos faltantes utilizando el algoritmo estructural EM (SEM). El método cuenta además con una forma de evaluación para las estructuras candidatas. Este algoritmo combina modificaciones estructurales y parámetricas con EM. La idea principal del SEM es que cada iteración comienza con la RBD actual. El algoritmo SEM tiene el mismo paso-E que el algoritmo EM y realiza unas modificaciones al paso M. Este trabajo también puede ser considerado para solucionar el segundo problema de nodos ocultos.

- Variables ocultas

Existen diversos trabajos desarrollados para aprendizaje de nodos ocultos, pero la mayoría se encuentran enfocados al área de aprendizaje parámetrico con estructuras ya creadas. Kwoh y Gillies [1996] proponen (en casos donde se buscan independencias entre variables) la creación de nodos ocultos, los cuales modelan las dependencias que podrían resolver el problema. Para la obtención de la matriz de probabilidades de esos nodos, se utiliza un método de gradiente descendiente. El objetivo de la función es minimizar el error al cuadrado entre la medida y el valor calculado del nodo instanciado. Cuando se trata de nodos padres que tienen más de dos hijos, se utiliza el método de creación simple de ”nodo-a-nodo”. En este trabajo quedan pendientes la determinación del número de estados necesarios para los nodos ocultos.

Binder y colaboradores [Binder et al., 1997], investigan el problema de aprendizaje de probabilidades con una estructura conocida y variables ocultas. Ellos presentan un algoritmo de aprendizaje parámetrico basado en gradiente-descendiente. Este trabajo muestra que el gradiente puede ser calculado localmente, usando información disponible como un subproducto de algoritmos de inferencia estándar para RB. Los resultados experimentales demuestran que utilizando conocimiento *a priori* acerca de la estructura, con variables ocultas, se puede mejorar significativamente el aprendizaje de RB. El algoritmo básico APN (*adaptive probabilistic networks*) calcula la distribución conjunta de todas las variables y éste es usado como una técnica de clasificación, que predice los valores de las variables especificadas (ocultas) dado la evidencia de variables especificadas (que influyen directamente) y recalculando las otras. Para medir la calidad de la predicción, utilizan una función que compara la diferencia entre probabilidades *reales* y las probabilidades *calculadas*. Este método se aplica a redes donde las tablas de probabilidad condicional son descritas utilizando un pequeño número de parámetros.

Como se mencionó anteriormente, el trabajo de [Friedman et al., 1997] sirve también para la detección de variables ocultas. El algoritmo EM es ampliamente utilizado tanto para datos faltantes como para variables ocultas [Dempster and Laird, 1977]. El algoritmo EM es un procedimiento iterativo que busca la hipótesis de máxima verosimilitud. El procedimiento es repetidamente ejecutado en dos fases, que se muestran en el 2.3. Este proceso se repite hasta que los valores de los parámetros convergen o hasta que exista una mínima diferencia entre los parámetros actuales y los del ciclo inmediato anterior. Una vez calculados los parámetros asociados a cada nodo se considera que

Algoritmo 2.3 Algoritmo EM.

1. **El paso-E**, que es el paso para la expectación. Donde se utilizan los parámetros actuales para posteriormente completar los datos de los valores no observados con valores esperados por medio de la técnica de relleno (”filling in”). Cuando se trata de datos incompletos los parámetros (denominados *actuales*) se calculan con los datos existentes como si fueran los reales. En el caso de nodos ocultos donde se desconocen los datos completamente (no hay datos), se proporcionan valores iniciales (con la ayuda de un experto o teniendo cierto grado de conocimiento de la aplicación) y se toman estos como si fueran realmente los parámetros actuales.

2. **En el paso-M**, que es el paso para la estimación de máxima verosimilitud, los datos completados son usados como si fueran reales, re-estimando la máxima verosimilitud de los valores de los parámetros, con la finalidad de ajustarlos. Posteriormente, se toman estas tablas y con esas nuevas distribuciones de probabilidad se repiten los dos pasos E y M, comenzando con el calculo de las nuevas frecuencias esperadas para los los valores del atributo H (de la misma manera como se realizo despues de completar aleatoriamente los valores faltantes) y entonces se re-estiman las probabilidades.

la estructura inicial está completa. Este algoritmo se usa comúnmente para ajustar los parámetros de una estructura fija, pero también ha sido aplicado a redes bayesianas dinámicas (algoritmo Estructural de Máxima Expectación -SEM- [friedman et al., 1999], con nodos ocultos (donde no se conoce ningún parámetro).

Algoritmo de aprendizaje iterativo

Lowd presentan un algoritmo para entrenamiento de clasificadores bayesianos que utiliza el EM para determinar nodos ocultos [Lowd, 1996]. El algoritmo NBE (*Naive Bayes Esxtructural*) es un proceso iterativo de dos pasos:

1. Extensión (adicionando componentes mezclados), y

2. Refinamiento (entrenamiento con esos componentes hasta que convergen)

Ambos pasos son importantes para la inicialización del modelo y para adicionar componentes en el aprendizaje parcial del modelo. El primer paso, para adicionar componentes necesita entrenar el modelo múltiples veces con diferentes números de componentes iniciales. Para el paso de refinamiento se utiliza el EM para encontrar los mejores parámetros para esos componentes y eliminar los componentes con menor peso. En cada iteración se dobla el número de componentes adicionado al modelo, cuidando también el tiempo de ejecución del proceso de aprendizaje, en el algoritmo NBE.

Algoritmo 2.4 Algoritmo NBE.

Procedimiento entrenaNBE

- Entrada: Un conjunto de entrenamiento T, un conjunto H, un número inicial de componentes K0 y un umbral de convergencia EM y Add.
- Inicialización de M con un componente

1. k=K0
2. repite
   - Adicionar una nueva mezcla de componentes k a M,
   - inicializar usando k ejemplos aleatorios a partir de T.
   - repite
     - Paso E: Asignar una parte de los ejemplos de T mezclando componentes, usando M.
     - Paso M: Ajustar los parámetros de M para maximizar la verosimilitud de la parte asignada.
     - Si log P(H—M) es mas alto , salvar M en Mmejor
   - hasta que log P(H—M) no pueda ser probado dependiendo del umbral de convergencia EM sobre la última iteración.
   - k= 2 X K
3. hasta que log P(H—M) no pueda ser probado de Add sobre la última iteración
4. Ejecuta el paso E y el paso M dos veces mas en Mmejor, usando ejemplos a partir de H y T.
5. Regresar Mejor

2.4.4. Discretización

Una desventaja que presenta el el clasificador bayesiano es que asume que todos los atributos son discretos o que tiene una distribución normal, lo cual no siempre sucede. Por lo que se hace necesario utilizar un método de discretización que sea adecuado al clasificador. Existen diferentes métodos de discretización que veremos el siguiente capítulo, así como la forma en que esta discretización afecta al proceso de aprendizaje y eficacia del clasificador.

2.4.5. Otras consideraciones

Existen otras consideraciones, además de las mencionadas, que se relacionan con el aprendizaje de clasificadores bayesianos. Entre éstas tenemos, que comúnmente se ha utilizado al clasificador bayesiano para describir modelos individuales, pero considerando que en el mundo real, muchas tareas de clasificación implican diferentes eventos, se pueden implementar varios modelos que representen la secuencia de esos eventos. Kang propone un método llamado RNBL, el cual construye un árbol de clasificadores que representan una secuencia de eventos, que fue probado en la representación de secuencias de proteinas y tareas de clasificacion de textos.

2.5. Resumen

Los clasificadores Bayesianos representan algunas ventajas para aprendizaje, principalmente porque son fáciles de construir y de entender. Existen diferentes tipos de clasificadores bayesianos, en este capítulo se trataron 4; el clasificador bayesiano simple, el TAN, BAN y la GBN. Existen diferentes métodos de aprendizaje para el clasificador bayesiano, sin embargo no se considera a uno como el mejor o el óptimo. Por otro lado podemos ver que comunmente los métodos abordan los problemas de aprendizaje en forma separada, asi tenemos los trabajos de Sucar y Pazzani, que tratan de capturar las relaciones entre atributos, solucionando el problema de considerar a estos como independientes entre si. El método de Ezawa, considera aprender un clasificador con base a atributos sobresalientes, tratando de reducir la estructura eliminando atributos que no se consideran útiles, este método es apropiado para aplicaciones con grandes bases de datos con un gran número de variables, para reducir el problema. Existen tambien métodos que se aplican en casos donde existe información incompleta, la técnica mas comunmente usada es el algoritmo EM, del cual existen diferentes variantes. Por otro lado tenemos el problema de la discretización, tema que se aborda en el siguiente capítulo, donde se da un panorama de trabajos relacionados. Sin embargo no tenemos un método que pueda abordar estos diferente problemas de aprendizaje de manera conjunta. Este trabajo de tesis busca principalmente tratar estos problemas de manera integral considerando además clasificación estática y dinámica.

Capítulo 3

Discretización

En el clasificador bayesiano los atributos continuos son manejados comúnmente asumiendo una distribución normal de los datos. Desafortunadamente esto no siempre es así, la realidad es que las distribución de los datos reales generalmente se desconoce. Otra forma de manejar los atributos continuos es la discretización. En este capítulo se presenta un panorama de la discretización de atributos continuos en el clasificador bayesiano.

3.1. Introducción

Una razón para aplicar un buen método de discretización en el proceso de aprendizaje es que mejora la clasificación. Otra razón para la discretización de variables es el incremento de la velocidad en los algoritmos de inducción [Catlett, 1991].

Diferentes métodos de discretización son aplicados en diversas áreas de aprendizaje, como C4.5 en árboles de decisión [Quinlan, 1993], K-medias en HMM, cluster, Histogramas, entre otros. Sin embargo éstos no son del todo adecuados para el clasificador bayesiano. Existen otros métodos especialmente creados para aplicarse al clasificador bayesiano, donde se ha observado que es posible reducir el error de la clasificación aplicando un método de discretización [Pazzani, 1996]. Un estudio de diferentes métodos de discretización y de cómo influye en la reducción del error en la clasificación se pueden encontrar en [Yang and Weeb, 2005].

A continuación se describe el proceso de discretización en el clasificador bayesiano, como influye en la reducción del error de clasificación y los factores que afectan a este proceso. Se presenta también una definición del proceso de discretización [Yang and Weeb, 2006], así como una clasificación básica de los métodos de discretización. Se describen diversos métodos de discretización comunmente aplicados a los clasificadores bayesianos, así como trabajos relacionados que muestran el impacto de la discretización en el aprendizaje de clasificadores bayesianos [Fayyad et al., 1996], [Pazzani, 1996], [Yang and Weeb, 2002], [Valdés et al., 2003] y otros.

3.2. Discretización en clasificadores bayesianos

Como vimos en el capítulo anterior, para determinar la clase más probable en un clasificador bayesiano simple necesitamos calcular:

$$P(H = h_i|E_1,E_2,...,E_n) = \frac{P(H = h_i)}{P(E_1,E_2,...,E_n)} \prod_{j=i}^{n} P(E_j|H = h_i) \tag{3.1}$$

Donde la clase H es discreta y los atributos E_i pueden se discretos o continuos. Debido a que los datos cuantitativos tienen características diferentes de los datos cualitativos [Bluman, 1992], [Samuels and Witmer, 1999], el cálculo de las probabilidades de la ecuación 3.1, es diferente para cada tipo de atributo. La clase usualmente toma un pequeño número de valores. La probabilidad de *P(H=h)* puede ser estimada a partir de la frecuencia de las instancias con *H=h*. La probabilidad *P(E=ei| H=h)*, donde E_i es cualitativa, puede ser estimada a partir de la frecuencia con *H=h* y la frecuencia de instancias con $E = e_i$, *H=h*. Para estimar *P(H=h)* se puede usar el estimador *Laplaciano* [1]:

$$\frac{n_c + k}{N + n \times k} \tag{3.2}$$

donde:

- n_c es el número de instancias que satisfacen *H=h*.
- N es el número de instancias de entrenamiento.
- n es el número de clases.
- k es igual 1.

Para estimar *P(E=ei| H=h)*, se usa estimado M [Cestnick, 1990]:

$$\frac{n_{ci} + m \times P}{n_c + m} \tag{3.3}$$

donde:

- n_{ci} es el número de instancias que satisfacen $E = e_i \wedge H = h$.
- n_c es el número de instancias que satisfacen *H=h*.
- P es $P(E = e_i)$.
- m es una constante.

Cuando se trata de atributos continuos, E_i tiene una gran cantidad de valores o incluso un número infinito de valores [Bluman, 1992], [Samuels and Witmer, 1999].

[1] Vease derivación en [Cestnick, 1990]

Lo que significa que la probabilidad de un valor particular e_i, dado la clase *H*, *P(E=ei| H=h)* puede ser infinitamente pequeño y generalmente hay muy pocas instancias para un solo valor del conjunto entrenamiento. Por lo que no es recomendable que la *P(E=ei| H=h)* sea derivada a partir de la frecuencia observada, en contraste a cuando se trata de atributos discretos, donde *P(E=ei| H=h)* es completamente determinada por una función de densidad de probabilidad *f* [Langley and Sage, 1994]. Cuando se trata de atributos continuos, para construir f generalmente se asume que los valores de dada la clase forman una distribución normal (Gaussiana) [Dougherty et al., 1995].

La discretización es una alternativa para procesar datos cuantitativos (el clasificador bayesiano maneja datos continuos asumiendo una distribución normal), convirtiéndolos a cualitativos para de esta manera estimar la densidad de probabilidad a partir de frecuencias. La discretización de un atributo cuantitativo consiste en corresponder cada valor $e*_i$ de $E*_i$ a un intervalo $[a_i, b_i]$ de $E*_i$, donde cada valor cuantitativo original e_i $[a_i, b_i]$. Un inconveniente es que dependiendo del número de instancias y de posibles valores cuantitativos (rangos) de que se trate, al escalar los datos cuantitativos puede haber pérdida de información.

3.2.1. Efecto de la discretización

Existen diversos trabajos que muestran que un buen proceso de discretización, aumenta la exactitud del clasificador bayesiano simple. En el trabajo realizado por [Dougherty et al., 1995] se muestra un análisis de tres métodos de discretización: intervalos de igual anchura, 1RD y minimización de la entropía, aplicándolos como un paso de pre-procesamiento para el algoritmo C4.5 y el clasificador bayesiano simple. En este trabajo Dougherty muestra empíricamente que el error de la clasificación puede ser reducido si se aplica un método de discretización en vez de suponer una distribución normal. Dougherty para estos experimentos uso 16 BD (con al menos un atributo continuo y cuando más 3000 instancias) extraídas del repositorio de Irvine [Murphy and Aha., 1994]. El clasificador bayesiano simple, aplicando los métodos de discretización, obtuvo en promedio 83.97 % de exactitud, contra el 76.57 % que fue el obtenido por el clasificador bayesiano simple asumiendo distribución Gaussiana. Concluyendo que la disparidad de los resultados en la exactitud del clasificador se debe a la deficiencia de la distribución Gaussiana ya que se considera inapropiada para algunos dominios.

Otra investigación referente a la efectividad de la discretización en el clasificador bayesiano, es la realizada por Ying Yang y Webb [Yang and Weeb, 2002]. Realizan un análisis de los factores que pueden afectar al error de clasificación, cuando los datos son procesados por discretización. Estos, sugieren que la exactitud de estimar la $P(H = h|E_i = e_i)$ en datos sin discretizar, para estimar la $P(H = h|E*_i = e*_i)$ en datos ya discretizados, puede ser la clave para la efectividad del método de clasificación. Ellos consideran 2 factores relevantes a considerar: los puntos de corte (límites) y el error de tolerancia en la estimación de probabilidad. La forma como la discretización trata con esos factores afecta a la *desviación* y la *varianza* al generar el clasificador. Otro punto a considerar es la probabilidad *a priori* de la clase la cual también afecta la clasificación. Valdés [Valdés et al., 2003] por su parte sugiere un método que realiza una discretización global considerando la relación entre atributos además de considerar la relación de los atributos con la clase. Aunque éste método ha sido probado en otros contextos (ID3, C4.5 y C4.5 rules), podemos observar que ésta es una de las principales desventajas del clasificador bayesiano simple, ya que en principio por definición, asume que todos los atributos son independientes entre sí, y no considera que no siempre esta suposición es verdadera. Otros puntos a considerar en el proceso de discretización, son: el número de atributos que conforman el problema, el número de instancias que conforman el conjunto de entrenamiento y el número de clases a clasificar.

3.3. Métodos de discretización

En general, una función de discretización consiste en agrupar los valores continuos de un atributo en rangos, porciones, intervalos o grupos de valores para usarlos como valores nominales. Sin embargo, esta tarea no es trivial, ya que ¿cómo decidir cuántos intervalos tendrá la variable en cuestión? ¿Qué pasa cuando existen variables cuyo flujo no es normal, sino que presentan diferentes variaciones en sus valores? A continuación se verán algunos métodos de discretización que están enfocados para contestar estas preguntas.

Con respecto a los métodos de discretización en general, comenzaremos con mostrar los principales grupos en los cuales se dividen. Se pueden clasificar en base a 4 dimensiones [Dougherty et al., 1995]:

- métodos globales y locales,
- métodos supervisados y no supervisados,
- métodos univariantes y multivariantes,
- métodos paramétricos y no paramétricos.

3.3.1. Metodos locales y globales

Los métodos locales [Quinlan, 1993], son aquellos que usan conjuntos de intervalos diferentes para cada atributo, consideran discretizar a cada atributo en forma independiente. Producen particiones que son aplicadas para localizar regiones de la instancia espacial (objetos similares que tienen una alta probabilidad de pertenecer a la misma clase). Los métodos globales [Chmielewski, 2005] son los que trabajan con respecto al conjunto completo de datos de entrenamiento, se consideran todos los atributos, donde cada atributo es particionando en regiones iguales.

3.3.2. Métodos supervisados y no supervisados

Los métodos de discretización supervisados, utilizan información de la clase para seleccionar el punto de corte para un nuevo intervalo y usan clases etiquetadas. La discretización supervisada puede ser caracterizada como basada en error, basada en entropía o basada en estadísticos dependiendo de la manera (medida usada) para seleccionar los intervalos, como [Kerber, 1992], [Holte, 1993], [Fayyad et al., 1996], otros. Los métodos no supervisados no utilizan información de la clase, son semejantes en cuanto a la igualdad en la anchura de los intervalos recipientes y en que no hacen uso de etiquetas en el proceso de discretización. Dividen en un cierto número de intervalos y no consideran necesario el conocer la distribución de los datos.

En Tabla 3.1 [Dougherty et al., 1995] se puede observar un resumen de algunos métodos de discretización, identificandolos como globales o locales y supervisados o no supervisados. Se describe un ejemplo de cada grupo en los siguientes párrafos.

Métodos	Globales		Locales	
	Métodos	Autores	Métodos	Autores
Supervisados	1RD	Holte (1993)	C5.5	Quinlan (1993)
	ChiMerge	Kerber (1992)	Vector de cuantización	Kohonen (1989)
	D-2	Catlett (1991b)	Máxima entropía jerárquica	Chiu et al. (1990)
	Particionamiento Recursivo de mínima entropía (Método basado en entropía)	Fayyad-Irani (1993) Ting (1992)	Multi-intervalo (Método basado en entropía)	Fayyad-Irani (1993)
	Cuantificador Adaptivo	Chang et al. (1991)		
	MCC	Van de Merckt (1993)		
	Valor máximo	Weiss et al. (1990)		
No supervisados	Intervalo de igual anchura	Catlett (1991)	Método basado en agrupamiento *Clustering*	Chimielwski-Grzymala-Busse (1994)
	Intervalos de igual frecuencia		Discretización iterativa	Pazzani (1995)
	MCC	Van de Merk (1993)		

Tabla 3.1: Resumen de métodos de discretización.

3.3.3. Métodos univariantes y multivariantes

Los métodos que discretizan cada atributo por separado son univariantes. Los métodos que discretizan considerando las relaciones entre el conjunto de atributos son multivariantes [Bay, 2000].

3.3.4. Métodos parámetricos y no parámetricos

La discretización parámetrica requiere que el usuario le proporcione una entrada, como número máximo de intervalos. La discretización no parámetrica no necesita que un usuario le proporcione ninguna entrada y sólo usa información a partir de los datos.

3.4. Métodos de discretización para el NB

En la literatura encontramos diferentes métodos de discretización, la mayoría aplicados a otros contextos diferentes al clasificador bayesiano (Naive Bayes). Sin embargo, también encontramos un número considerable de ***métodos de discretización aplicados al clasificador bayesiano***. En esta sección se describen algunos de ellos, que consideramos representativos de acuerdo a su forma de realizar el proceso. A continución se describen algunos métodos de discretización, según la clasificación de [Dougherty et al., 1995].

3.4.1. Métodos globales y no supervisados

Intervalos con igual anchura

La discretización con intervalos de igual anchura (*EWD*) es el método más simple de discretización de datos. Esto involucra clasificar los valores observados de una característica y dividir el rango de valores observados de la variable en *k* partes de tamaño igual, donde *k* es un parámetro dado por el usuario [Catlett, 1991] [Dougherty et al., 1995]. Entonces tenemos:

$$\delta = \frac{X_{max} - X_{min}}{k} \tag{3.4}$$

X_{max} y X_{min} son los límites y *k* el número de intervalos. El método es aplicado para cada valor continuo independientemente. Éste no utiliza información de instancias de clases por lo que se le considera un método no supervisado.

Aunque el método es muy sencillo existen algunos problemas al realizar particiones de esta forma. En principio, no queda claro como se selecciona el mejor valor para *k*, tampoco queda claro que se pueda usar el mismo valor de *k* para cada atributo y por último el método no considera valores críticos en la variable, pues sólo divide la variable en partes iguales.

Intervalos basados en frecuencias

La discretización con intervalos de igual frecuencia (*EFD* [Catlett, 1991] y [Dougherty et al., 1995]) divide los valores continuos en *k* (parámetro definido por el usuario) intervalos procurando que cada intervalo tenga aproximadamente el mismo número de instancias de entrenamiento. Cada intervalo contiene *n* instancias de entrenamiento con valores adyacentes (posiblemente identicos). Al realizar las particiones, puede ser que instancias similares sean colocadas en el mismo intervalo por lo que no siempre es posible generar *k* intervalos con igual frecuencias.

3.4.2. Métodos locales y supervisados

Particionamiento recursivo de entropía mínima

El algoritmo de particionamiento recursivo de mínima entropía [Fayyad et al., 1996] está clasificado como un método de discretización supervisado y local (Fayyad lo aplica localmente en cada nodo durante la generación de un árbol), pero *también puede ser tratado como un método global* [Ting, 1994]. Este método busca un buen conjunto de puntos para discretización de atributos numéricos. El método está basado en entropía de la información y usa MDL [Rissanen, 1978] como criterio de paro para determinar cuándo debe parar de subdividir intervalos. Este algoritmo utiliza el concepto de entropía para seleccionar el punto límite de los intervalos.

Para un atributo (*A*), se busca el punto límite de la partición (*Tmin*) el cual minimice la función de entropía sobre todas los posibles límites de las particiones y el cual es seleccionado como un límite de discretización arbitrario. Este método puede ser aplicado recursivamente para particiones inducidas por Tmin hasta que alguna condición de paro sea activada, creándose múltiples intervalos del atributo *A*. Para determinar el criterio de paro (CP) se usa el principio de descripción de longitud mínima (MDL). El criterio es el siguiente: la partición producida para un punto de corte T de un atributo A de un conjunto S de N ejemplos es aceptada si:

$$Gain(A,T;S) > \frac{log_2(N-1)}{N} + \frac{\Delta(A,T;S)}{N} \tag{3.5}$$

y es rechazada de otra manera. Donde:

$$\Delta(A,T;S) = log_2(3^k - 2) - [k * Ent(S) - k_1 * Ent(S_1) - k_2 Ent(S_2)] \tag{3.6}$$

y

- *Gain(A,T;S)* es la ganancia de información de un punto de corte *T*.
- *Ent(S)* es la entropía de la clase de un subconjunto *S*
- *K* es el número de clases en el subconjunto S_i.

Debido a que las particiones son evaluadas independientemente, algunas áreas en el espacio continuo podrían ser particionadas muy finamente (intervalos muy pequeños) mientras otras (los cuales tienen entropía relativamente baja) podrían ser particionadas toscamente (intervalos de gran tamaño o con elementos disjuntos).

Discretización iterativa

[Pazzani, 1996] propone un método iterativo que crea un conjunto de intervalos iniciales ya sea definidos por el usuario o automáticamente, para después aplicar el algoritmo iterativo. El problema de buscar un buen conjunto de puntos límites para discretizar valores para atributos numéricos puede ser visto como un problema de búsqueda. En particular, un método podría generar todos los posibles puntos limites y estimar el error (o costo del error de clasificación) del método de aprendizaje con esos puntos limites usando validación cruzada dejando uno fuera (LOOCV). Desafortunadamente, generar y probar todos los posibles puntos límites es impráctico. En el peor caso, hay al menos *O(2AN)* posibles puntos límites, donde *A* es el número de atributos numéricos y *N* es el número de instancias. LOOCV es usado para estimar el error debido a que puede ser eficientemente implementado, con el clasificador bayesiano.

Algoritmo 3.1 Algoritmo Iterativo.

1. Estimar el error (o costo de error en la clasificación) de cada ajuste usando LOOCV del conjunto de puntos limites y reordenar los ejemplos mal clasificados hasta que sean correctamente clasificados
2. Reordenar los atributos aleatoriamente
3. Para cada atributo en el conjunto de atributos
   - A Aplicar todos los operadores en todas las formas posibles para el punto limite actual del atributo
   - B Estimar el error (o costo del error en la clasificación) de cada ajuste usando LOOCV
   - C Si el error de algún ajuste es menor que el error del punto limite actual, entonces hacer el ajuste con el menor error
4. Si el punto límite no fue ajustado en el paso 3 entonces retornar el punto limite actual, de lo contrario ir al paso 1 (tomar nuevos ejemplos)

Con este método, un clasificador bayesiano puede ser construido completamente a partir del conjunto de entrenamiento y cuando va dejando un ejemplo fuera, la contribución de este ejemplo en la probabilidad estimada, es eliminada antes de clasificar el ejemplo. Para optimizar la velocidad de la validación cruzada de LOOCV, los ejemplos son ordenados en el paso 2, como ejemplos mal clasificados, y estos son usados primero por el clasificador con la partición actual. Cuando se calcula el error de clasificación con un nuevo punto límite, se evalúa si se obtuvo el error mínimo con la partición actual, para detener la prueba de LOOCV y no tener que seguir calculando para todos los ejemplos. Este método iterativo, de discretización supervisado aplicado al clasificador bayesiano, tiene un procedimiento de inicio, que parte de un conjunto *semilla* de puntos límites (p.e. iniciando la discretización de un atributo en 5 particiones) y dos operadores que ajustan esos puntos límites, los cuales son:

1. Mezcla de dos intervalos contiguos
2. División de un intervalo en dos intervalos considerando introducir un nuevo punto límite que es el punto medio del intervalo.

El proceso de ajuste utiliza una estrategia de búsqueda iterativa que se describe en el algoritmo 3.1.

En conclusión, vemos que este método es fácil de implementar y en general es rápido (en los experimentos, no requirió más de 5 minutos de CPU, usando una SPARC 20). Otro punto a su favor es que considera ejemplos mal clasificados para incluirlos en el proceso y en general en las pruebas obtiene buenos resultados. Sin embargo, no garantiza encontrar el conjunto óptimo ya que es sensible al cambio de semilla inicial y esto puede arrojar diferentes resultados. También deja pendiente la manera de seleccionar la semilla inicial, ya que sólo menciona hacerlo en formas aleatoria o dejar que el usuario la determine, lo cual no siempre es posible de obtener. Otro punto es que no se toma en cuenta la forma de ordenamiento de las variables ya que lo hace en forma aleatoria, lo que de hacerse podría obtener mejores resultados.

3.5. Otros métodos de discretización

Existe un gran número de métodos de discretización para máquinas de aprendizaje. La mayoría de estos son aplicados a otros contextos diferentes al clasificador bayesiano, como son árboles de decisión, reglas de decisión, etc. A continuación se presentan algunos de estos, que podrían ser aplicados al clasificador bayesiano y/o a redes bayesianas.

3.5.1. Discretización basada en el MDL

[Friedman et al., 1997] introducen un método de discretización de atributos continuos basado en el principio de descripción mínima (MDL). La discretización se lleva a cabo mientras se realiza el proceso de aprendizaje de *redes bayesianas*. El concepto base es el uso del principio MDL, esta medida balancea la complejidad del aprendizaje de la discretización y el aprendizaje de la estructura de la red. Ésto garantiza que la discretización de cada variable tenga los intervalos justos que capturen su interacción con las variables adyacentes en la red. El método es una generalización del propuesto por Fayad e Irani [Fayyad et al., 1996], el cual es un método de aprendizaje supervisado para discretizar variables buscando maximizar la información mutua con respecto a la variable clase. La diferencia es que la discretización de cada variable maximizando la información mutua se realiza con respecto a todas las variables relacionadas en la estructura. Esto es, considerando todos los nodos vecinos de cada variable a discretizar.

En comparación con el método original, aplicado para aprendizaje de redes bayesianas de [Lam- Bacchus, 1994], la diferencia principal consiste en que, para la discretización, se tiene un término adicional *DLA(A)* (Longitud de descripción para discretizar a *A* en una estructura *D*). Este término refleja el monto de involucrar un número *x* de intervalos en la discretización reflejado en la información mutua. El problema entonces es buscar una discretización que mínimice no una estructura sino una relación *DLlocal* (la variable a discretizar y los nodos que la afecten -manto de Markov-).

El involucrar la discretización en el proceso de aprendizaje representa diversas ventajas. Primero, obtiene los intervalos justos para cada variable reflejando su interacción con las variables adyacentes. Segundo, las variables que no tienen interacción con otras variables pueden discretizarse en forma independiente. Los principales problemas que se presentan son: la búsqueda puede resultar muy extensa dependiendo del número de variables a discretizar, de la complejidad de la estructura, cuando existe un gran número de relaciones entre las variables a discretizar (el manto de Markov se extiende) y cuando existe un gran número de valores en las variables a discretizar (BD extensas), el cálculo de las probabilidades crece en forma exponencial.

3.5.2. Discretización manejando *desviación y varianza*

[Yang and Weeb, 2002] realiza un análisis de diferentes métodos de discretización y propone tres nuevas técnicas de discretización para el *clasificador bayesiano*. Estas técnicas manejan la desviación y la varianza (*bias and variance*): para ajustar la frecuencia de los intervalos y el número de intervalos, puesto que argumentan que son factores que influyen en el error de clasificación. Estos métodos son: discretización proporcional, discretización de frecuencia fija y discretización no disjunta. A continuación se da una breve descripción de estos métodos.

Discretización proporcional (PD)

Este método parte de la consideración que realizan [Moore and S., 1994], la cual argumenta que un buen método de discretización debe tener baja desviación y baja varianza, lo cual utilizan para discretizar por igualdad de pesos. Como resultado de un análisis, determinan que discretizaciones con intervalos que tienen mayores frecuencias, tienden obtener baja varianza y alta desviación, al contrario de discretizaciones con gran número de intervalos que tienden a obtener baja desviación y alta varianza. Por lo que el método consiste en buscar un equilibrio entre el número de intervalos y la frecuencia de los intervalos, es decir, busca un conjunto de intervalos de igual frecuencia e igual número de particiones, donde se tiene un número *k* de intervalos para las variables y en cada intervalo un mismo número de instancias. A esta discretización se le llama discretización proporcional (PD). Para determinar el número de intervalos y de frecuencias, se usa un incremento en el número de instancias en el conjunto de entrenamiento.

Discretización de frecuencia fija (FFD)

EFD fija el número de instancias en un intervalo, para discretizar a un atributo continuo en *k* intervalos, donde cada intervalo tiene el mismo número de instancias de entrenamiento. FFD fija la frecuencia del intervalo. FFD forma conjuntos de intervalos de igual frecuencia *m*. Los valores son ordenados enforma ascendente y colocados con frecuencia aproximadamente igual en cada intervalo. Se pruebean diferentes números de frecuencia y de intervalos.

Discretización no disjunta (NDD)

Al formar intervalos, los valores de una instancia pueden ser colocados en uno de dos tipos de intervalos: favorable o desfavorable. Suponiendo que una instancia (compuesta por diferentes atributos) tiene como clase mas probable un valor *c*, los valores discretizados de esos atributos, si son colocados en un intervalo favorable, significa que cada uno de los valores de esa instancia deberan corresponder mayormente con la clase mas probable *c*, si por el contrario son colocados en un intervalo desfavorable la mayoría de los valores de los atributos de esa instancia corresponderán a la clase menos probable. La discretización no disjunta se guía por esta evaluación, de tal manera que puede ser que un valor continuo pueda ser discretizado y colocado en más de un intervalo, formando intervalos traslapados, para después ser cambiado a un intervalo más favorable.

3.5.3. Discretización 1RD

Este algoritmo de discretización llamado 1RD (*one rule discretizer*) [Holte and Porter, 1989] clasifica los valores observados de un atributo continuo y divide el dominio del atributo en partes, dividiendo en un número finito de intervalos, tratando de hacer cada intervalo *puro* (cada uno conteniendo sólo instancias de una clase particular). Sin embargo ésto podría originar que se continuar subdividiendo un intervalo hasta que todos lo intervalos sean *puros*, lo cual podría resultar en que cada intervalo tuviera sólo una instancia de cada valor real observado. Para evitar esto, el algoritmo forma particiones de al menos un tamaño mínimo, excepto la partición del extremo derecho que podría ser de un tamaño menor. El principal problema que se observa en este método es la definición del tamaño mínimo de las particiones. El tamaño propuesto por Holte podría no ser apropiado en muchas aplicaciones ya que se basa únicamente en pruebas que él consideró, lo cual es un universo muy reducido.

Discretización basada en error

Este método discretiza un atributo continuo con respecto al error en el conjunto de entrenamiento. Se utiliza un parámetro k, para definir el máximo número de intervalos y se obtiene un conjunto óptimo de intervalos, si después de discretizar las instancias, éstas son bien clasificadas [Maass, 1994], [Jhon and Kohavi, 1997].

3.6. Resumen

Existen diversas aplicaciones que manejan atributos continuos. En el clasificador bayesiano los atributos continuos son tratados comúnmente asumiendo una distribución normal de los datos. Desafortunadamente esto no siempre es así, la realidad es que las distribución de los datos reales generalmente se desconoce. La discretización es una alternativa para procesar datos continuos, convirtiéndolos a discretos. Un inconveniente es que dependiendo del número de instancias y de posibles valores continuos de que se trate, al escalar los datos continuos puede haber pérdida de información.

La efectividad de un método de discretización en el clasificador bayesiano se prueba principalmente con respecto a la reducción del error de clasificación. El efecto que tiene la discretización en la exactitud del clasificador se ha analizado a través de diferentes trabajos, [Dougherty et al., 1995] muestra mediante un análisis empírico que el error de la clasificación puede ser reducido si se aplica un método de discretización en vez de suponer una distribución normal. Por su parte [Yang and Weeb, 2002] establecen las condiciones particulares sobre las cuales se puede trabajar la discretización en un clasificador bayesiano.

Observamos que diversos factores afectan el proceso de discretización, entre estos encontramos la determinación de los puntos de corte (límites), el error de tolerancia en la estimación de probabilidad, la relación entre atributos, el número de atributos que conforman el problema, el número de instancias que conforman el conjunto de entrenamiento y el número de clases a clasificar.

El principal problema que se presenta al tratar de realizar un método de discretización general es que está estrechamente relacionado con la aplicación y con el comportamiento de los datos. Y aún más, dentro de una misma aplicación se pueden tener variables con valores donde la distribución de los mismos no presenta el mismo flujo de datos, por lo que también puede depender de los valores de cada variable.

En base al análisis de diferentes métodos de discretización encontramos que estos son aplicados, en su mayoría, en forma separada del aprendizaje de la estructura y que el principal problema de encontrar la mejor discretización, es precisamente el espacio de búsqueda de los mejores puntos de corte. En este trabajo presentamos un nuevo método de discretización basado en el principio MDL, el cual considera el ir discretizando los atributos continuos individualmente y aprendiendo la estructura al mismo tiempo. Este método se describe en el capítulo 5.

Capítulo 4

Redes bayesianas dinámicas

Un clasificador bayesiano puede representar dominios estáticos o puede también representar dominios dinámicos. El clasificador bayesiano es un tipo de estructura de red bayesiana (RB), y en este contexto lo podemos ubicar para hablar del aprendizaje automático de redes bayesianas en general, y en particular de redes bayesianas dinámicas (RBD). En este capítulo se describen conceptos relacionados con las redes bayesianas dinámicas, iniciando con una definición de RB, posteriormente se describen los conceptos de RB estáticas y dinámicas, así como tambien se muestran algunos métodos relacionados.

4.1. Redes bayesianas

Las redes bayesianas (RB) o probabilísticas consideran que para describir el mundo real no es necesario utilizar una tabla de probabilidades en las que se listen todas las combinaciones concebibles de sucesos. La mayoría de los sucesos son condicionalmente independientes de la mayoría de los demás. Por lo que no deben considerarse todas sus interacciones. En lugar de ésto se puede usar una representación más local en donde se describan grupos de sucesos que interactúen [Pearl, 1988].

Por ejemplo, el que una persona se enferme de tifoidea se considera que es independiente de que tenga un accidente de auto o de que el nivel de ozono aumente o disminuya; sin embargo, si se quiere predecir la probabilidad de que una persona se enferme de tifoidea entonces se tendrían que consideran los factores que pueden influir para que se presente o no la enfermedad.

Formalmente una red bayesiana (RB) se define como un Grafo Acíclico Dirigido (GAD) en la que se tiene [Pearl, 1988], [Neapolitan, 1990]:

- Un conjunto de nodos que representan variables aleatorias y un conjunto de arcos dirigidos entre pares de nodos.
- Una variable es independiente de sus no descendientes dados sus padres.

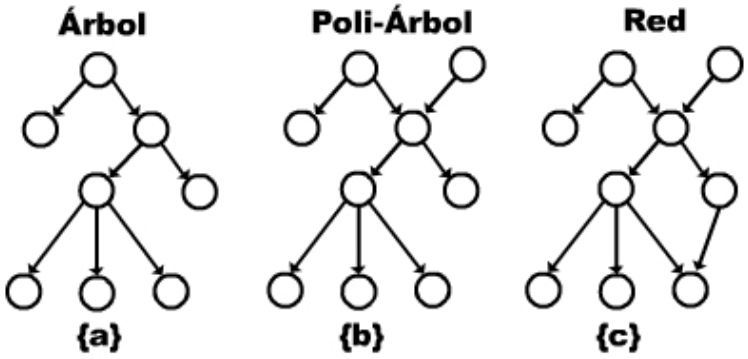


Figura 4.1: Diferentes estructuras de redes Bayesianas. (a) árboles, (b) Poliárboles y (c) Redes Multiconectadas.

La variable al final de un arco es dependiente de la variable que se encuentra al principio del mismo, y se les denomina, nodo hijo y nodo padre, respectivamente. Una red Bayesiana tiene al menos un nodo raíz (sin padre alguno) y un nodo terminal (sin hijo alguno). Una red bayesiana puede adquirir, dependiendo del dominio que se quiera modelar, tres tipos de estructuras. La estructura más simple son los árboles, donde los nodos tienen un solo padre, excepto el nodo raíz, figura 4.1(a). Un poli-árbol se puede ver como varios árboles conectados, donde hay nodos que tienen dos o más nodos padres, figura 4.1.(b). La estructura más compleja es la red múlticonectada, que puede modelar mejor a diversos fenómenos, ésta estructura puede tener más de una trayectoria entre parejas de nodos, figura 4.1.(c) [Neapolitan, 1990].

4.1.1. Aprendizaje de redes bayesianas

Existen diferentes técnicas de aprendizaje automático para obtener la estructura y parámetros de una red. El aprendizaje en redes bayesianas se divide en dos partes: aprendizaje paramétrico y aprendizaje estructural. El aprendizaje paramétrico consiste en que a partir de datos y una estructura conocida de una RB se obtengan las probabilidades a priori y condicionales requeridas. El aprendizaje estructural consiste en obtener la estructura de una RB, describiendo las relaciones de dependencia e independencia entre las variables involucradas. Los métodos de aprendizaje estructural, se pueden dividir en dos tipos según sea la forma en que considera a una RB:

- Métodos basados en ajustes (búsqueda y evaluación)
- Métodos basados en análisis de dependencias

El primer tipo considera que una RB es una estructura que codifica la distribución conjunta de los atributos, lo cual sugiere que la mejor estructura de red es aquella que mejor se ajusta a los datos. Por tanto existen algoritmos de aprendizaje basados en medidas, los cuales integran un método de búsqueda para proponer diferentes estructuras y una forma de evaluación para determinar cuál es la mejor, buscando una estructura que optimice la medida de ajuste de la red. Las medidas más utilizadas para este propósito son la medida Bayesiana[1] [Cooper and Herskovits, 1992] y la función de entropía MDL [Lam and Bacchus, 1994].

[1] calcula la proporción de la probabilidad posterior para pares de estructuras

El segundo tipo se basa en que la estructura de una RB codifica un grupo de relaciones de independencia condicional entre los nodos, de acuerdo al concepto de separación-d [Pearl, 1988]. El aprendizaje de la estructura se realiza por identificación de relaciones de independencia condicional (CI) entre los nodos. Usando alguna prueba estadística (como Chi-cuadrada e información mutua), se pueden buscar las relaciones de independencia condicional entre los atributos y usar esa relación como restricción para construir una RB. Estos algoritmos son referidos como algoritmos basados en restricciones ó algoritmos en base a CI [Spirtes et al., 1993], [Cheng and Greiner, 1999].

Otras dos consideraciones que se deben hacer para aplicar un método de aprendizaje son:

- Tipo de estructura de que se trate: clasificador bayesiano, árboles, poli-árboles o redes multi-conectadas.
- Tipo de redes bayesiana: estáticas o dinámicas.

4.2. Redes bayesianas dinámicas

Las estructuras anteriores representan un dominio *estático*, describiendo una distribución de probabilidad sobre un conjunto fijo de variables. Sin embargo, existen fenómenos que no se pueden describir con estas estructuras, puesto que cambian con el tiempo. Las redes bayesianas dinámicas (RBD) vienen a resolver este problema, puesto que modelan la evolución estocástica de un conjunto de variables $X = X_1, ..., X_n$ a través del tiempo. Al igual que las RB estáticas, la RBD son definidas por una estructura gráfica y un conjunto de parámetros, en el cual también se especifica una distribución conjunta sobre las variables aleatorias.

En general, en RBD, se considera lo siguiente:

- Suposición 1. Los procesos son markovianos: es decir, las variables del periodo de tiempo actual son condicionalmente independientes de las variables de periodos de tiempo pasados, dadas las variables del tiempo inmediatamente anterior.
- Suposición 2. Los procesos son estacionarios: es decir, que las probabilidades de transición (entre un periodo de tiempo y otro) son independientes del tiempo.

Las RBD se constituyen de dos partes [Friedman et al., 1997]:

- i) Una estructura inicial *Bo* que especifica la distribución sobre el estado inicial *X[0]* y
- ii) Una red de transición $B\rightarrow$ sobre las variables $X[0] \cup X[1]$ que es tomada para especificar la probabilidad de transición $P(X[t+1]|[t])$ para todo *t*.

La probabilidad esta dada por:

$$P_B(X[1]|X[0]) = \prod_{i=1}^{n} P_{B\rightarrow}(X_i[1]|Pa(X_i[1])) \tag{4.1}$$

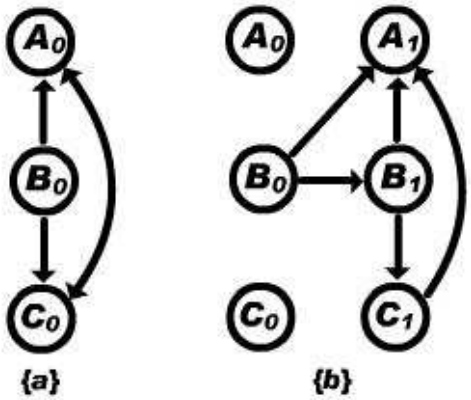


Figura 4.2: Red bayesiana dinámica (a) Estructura inicial. (b) Red de transición.

La figura 4.2 muestra un ejemplo de una RBD [Stephenson, 2000]. En la figura 4.2a se muestra una posible estructura inicial *Bo* con tres variables estáticas (*A,B,C*). Esta representa las probabilidades a priori para todas las variables de la red en un periodo de tiempo *t=0*. En la figura 4.2b se muestra la posible red de red de transición B→, está estructura representa para todos los periodos de tiempo *t=1,t=2,..., t=n* las probabilidades condicionales de cada variable dadas las otras variables.

En una red bayesiana dinámica el conjunto de probabilidades y variables definidas son las mismas para cada periodo de tiempo, con excepción de la red *a priori* en el tiempo inicial *t=0*, la cual tiene sus propias distribuciones de probabilidad. Estos es, una RBD se puede construir a partir de la red inicial (la estructura es repetida en cada periodo de tiempo) y la red de transición (también para cada periodo de tiempo). En la figura 4.3, se muestra una red dinámica para 5 periodos de tiempo *t=0,...,t=4* construida a partir de la red inicial en un periodo de tiempo *t=0* y la red de transición.

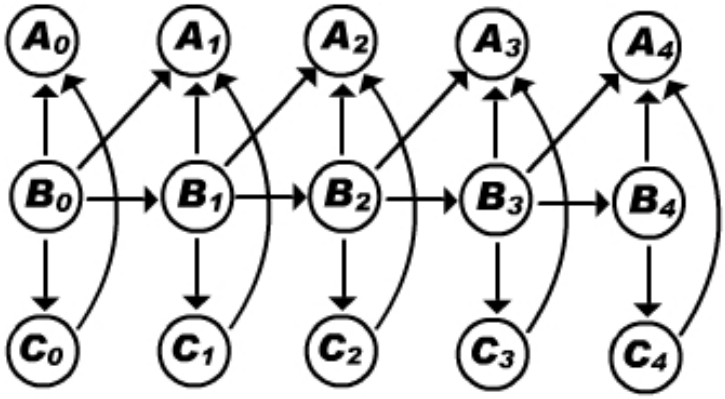


Figura 4.3: Red bayesiana dinámica que evoluciona a través de cinco períodos de tiempo.

4.2.1. Aprendizaje de redes bayesianas dinámicas

El aprendizaje de redes bayesianas dinámicas es más complejo. Se debe construir tanto la red inicial Bo (que especifica una distribución sobre el estado inicial del proceso) como el modelo de transición *B→* definido sobre *B(t,t+1),* que especifica las probabilidades de transición entre los estados del proceso temporal.

Al igual que en RB estáticas o tradicionales, el aprendizaje de RBD, se divide en dos partes: estructural, aprendizaje de la estructura inicial y la red de transición; y paramétrico, aprendizaje de las probabilidades asociadas con las estructura. Se pueden presentar dos casos: cuando los datos son completos o todas las variables presentes en la estructura son observadas y cuando existen algunas variables ocultas, o son variables no observadas. Cuando los datos son completos, la meta principal del aprendizaje paramétrico, es encontrar la MLE (Estimación de Máxima Verosimilitud) de los parámetros de cada CPD (tabla de Distribución de Probabilidad Condicional). Cuando algunos de los nodos son ocultos, se puede usar el algoritmo EM para encontrar (localmente) un óptimo MLE de los parámetros de cada CPD [Murphy and Aha., 1994].

A continuación se presentan dos métodos de aprendizaje, uno perteneciente al tipo de métodos basado en ajustes y otro método basado en análisis de dependencias.

4.2.2. Métodos de aprendizaje basados en ajustes

[Friedman et al., 1997] realiza un trabajo sobre redes dinámicas con datos faltantes utilizando el algoritmo EM estructural (SEM) [Friedman et al., 1997]. El método cuenta además con una forma de evaluación para las estructuras candidatas. Este algoritmo combina modificaciones estructurales y parámetricas con EM. El algoritmo SEM tiene el mismo paso-E que el algoritmo EM y realiza unas modificaciones al paso M.

Friedman en el algoritmo SEM, considera el mismo paso-E, completando los datos faltantes por medio de los valores esperados basados en la estructura y los parámetros actuales. Pero adicionalmente, en el paso-M, calcula, de acuerdo a la estructura actual, la medida de una estructura candidata. Es decir el algoritmo SEM completa los datos faltantes usando la red actual y funciona como una estructura con datos completos en la búsqueda de la mejor estructura. Después de un número de pasos, el algoritmo detiene la búsqueda de estructura, usando la red actual para completar los datos; y el proceso se repite.

El algoritmo propuesto por Friedman presenta la ventaja de que cubre diversos aspectos como son técnicas de búsqueda para aprendizaje en presencia de variables ocultas, manejo de información incompleta e inferencia para redes complejas. El algoritmo es capaz de detectar correlaciones que involucran interacciones temporales; sin embargo, este no es capaz de realizar correctamente el descubrimiento cuando involucra un gran número de períodos de tiempo. No soporta variables que involucren diferentes velocidades (escalas de tiempo), supone que todas las variables dinámicas evolucionan en un periodo de tiempo similar. El algoritmo tampoco consideran cuestiones sobre el número de iteraciones y acceso a la BD (no consideran la dimensión de ésta), tampoco definen con qué porcentaje de valores faltantes el método obtiene resultados aceptables, así mismo no se define un criterio de paro para el algoritmo.

4.2.3. Métodos basados en análisis de dependencias

[Campos and Puerta., 2000], propone utilizar cualquier algoritmo de aprendizaje de RB para crear la estructura *Bo*, y posteriormente si el proceso es estacionario y markoviano se puede repetir la misma estructura para cada periodo de tiempo *t* y aprender el modelo de transición utilizando el algoritmo ART.

Algoritmo 4.1 Algoritmo de aprendizaje de relaciones temporales (ART).

Entrada:
Bo y B(t+1) ordenados topológicamente empezando por los nodos raíces.
Salida:
$B \rightarrow (t+1)$

1. B→(t+1)=Û
2. Para *j*=1 hasta *n* hacer:
3. Seleccionar el nodo x_j(t+1) de B(t+1)
   a) Para i = 1 hasta n hacer
   b) Seleccionar el nodo x_i(t) de B(t)
   c) Si $\neg I(\mathrm{x}_i(t), x_j(\mathrm{t+1}) | \mathrm{MM}^{ij})$ entonces
   d) B→(t+1)= B→(t+1) $\cup(\mathrm{x}_i(t), x_j(t+1))$
   e) Fin para
4. Fin para i=n

Este algoritmo realiza pruebas de independencia condicional entre el conjunto de variables del periodo de tiempo actual *V(t)* y el conjunto de variables del periodo de tiempo siguiente *V(t+1)*, entre cada par de variables, una por cada periodo de tiempo. Las pruebas se realizan en orden topológico ancestral determinado por sus estructuras *Bo)* y *B(t+1)*, dado un subconjunto de variables *V(t, t+1)*, de tal manera que dependiendo del resultado de las pruebas, se compruebe si existe o no una relación temporal. El conjunto V(t, t+1) se denomina *manto de Markov* del nodo i-ésimo $x_i(t) \in V(t)$, según un orden ancestral para *Bo*, asociado al j-ésimo x_j *(t+1)*, según el orden ancestral para *B(t+1)*, de la RBD. A ese conjunto se le denomina MM^{ij}. En en algoritmo 4.1, se describe los pasos de este proceso.

En general, el algoritmo funciona de la siguiente manera. El conjunto de relaciones temporales estará inicialmente vacío. Se comenzarán a realizar pruebas de independencia condicional dado el subconjunto MM^{ij} por los nodos raíces de las estructuras *B(t)* y *B(t+1)*. Una vez aprendidos los arcos temporales que le llegan al nodo raíz de *B(t+1)* desde cualquier nodo de *Bo* entonces se pasa a comprobar el siguiente nodo de *B(t+1)* en el orden topológico.

En este método el número de pruebas de independencia condicional para cada pareja de variables, siendo una por cada periodo de tiempo, es de n^2, donde n es el número de variables para *Bo)*. Por lo cual, si se tienen estructuras muy complejas con muchas variables, el número de pruebas se incrementa. Por otro lado se tiene el problema de que la sabana de Markov incluya muchas variables por lo que el número de pruebas a realizar se incrementaría.

4.3. Modelos Ocultos de Markov (HMMs)

Las RBD resultan ser una generalización de los modelos ocultos de Markov (HMM) y de sistemas dinámicos lineales como los filtros Kalman [Kalman, 1960] [2], representaciones menos estructuradas que una red bayesiana dinámica [Murphy 01]. Los modelos ocultos de Markov son una técnica estadística de amplio uso en el reconocimiento de patrones secuenciales de datos. Estos modelos pueden ser aplicados a una gran variedad de problemas, debido a que existen diversas modificaciones a la estructura básica de los mismos. Actualmenete gozan de gran popularidad, entre otras razones, por que pueden caracterizar de manera precisa datos en presencia de ruido y ligeras variaciones.

Un HMM es una máquina de estados finitos caracterizada por dos procesos estocásticos, uno de estos determina la transición entre los estados y es no observable, el otro proceso, genera observaciones de salida para cada estado. Los estados no son determinados de manera directa a partir de las observaciones, por lo que son considerados como ocultos. La característica principal de un HMM, consiste en su habilidad para encontrar la secuencia más probable de estados que pudo haber generado una secuencia dada de observaciones.

4.4. Resumen

Las redes bayesianas (RB) son un modelo gráfico que representa relaciones causales entre variables y puede representar dominios estáticos o dominios dinámicos. Existen diferentes tipos de redes bayesianas: árboles, poli-árbole y redes multiconectadas. Los clasificadores bayesianos son otro tipo de red bayesiana. Las RB dinámicas al igual que las RB estáticas, son definidas por una estructura gráfica y un conjunto de parámetros. La diferencia consiste en que para modelar una RBD se necesita además de la estructura inicial (una estructura que se repite en cada periodo de tiempo, equivalente a una red estática) una red de transición entre cada período de tiempo.

El aprendizaje de redes bayesianas se divide en dos partes, aprendizaje paramétrico (cálculo de las probabilidades asociadas a cada variable) y aprendizaje estructural (creación de la estructura o modelo). En el aprendizaje estructural los métodos pueden dividirse básicamente en dos tipos: métodos basados en ajustes (que involucran búsqueda y evaluación de la mejor estructura) y métodos basados en análisis de dependencias (identificación de relaciones de independencia condicional $-CI-$). Se presentaron en esta sección dos métodos (uno de cada tipo) aplicados a procesos dinámicos. También se presentó una descripción de los HMM ya que éstos son una forma particular de RBD.

El aprender RBD en general, sobre todo considerando nodos ocultos, es un proceso muy complejo. En esta tesis nos enfocamos a un tipo particular de RBD, el clasificador bayesiano dinámico que se describe más adelante.

[2] El filtro de Kalman es una técnica recursiva para determinar los parámetros correctos de un sistema que evoluciona con el tiempo. Dados unos estimadores iniciales y los parámetros propios del sistema dinámico, el filtro va prediciendo y auto ajustándose con cada nueva medida

Capítulo 5

Aprendizaje de clasificadores bayesianos estáticos

En el capítulo 2 se trató el aprendizaje de clasificadores bayesianos simples y, aunque vimos que existen diferentes métodos de aprendizaje, no se considera a uno como el mejor, además de que existen diversas consideraciones que afectan al proceso de aprendizaje. En el capítulo 3 vimos, particularmente, como afecta la discretización en dicho proceso. El método de aprendizaje de clasificadores bayesianos simples que proponemos incluye la búsqueda de una estructura que mejore la exactitud del clasificador conservando la simplicidad de la misma haciendo uso de métodos de discretización y mejora estructural. En este capítulo se describe el proceso de aprendizaje de nuestro modelo estático.

5.1. Introducción

El clasificador bayesiano simple es un modelo de clasificación eficiente, fácil de aprender y con gran exactitud en muchos dominios. Sin embargo, tiene dos principales desventajas:

- (i) La exactitud de la clasificación disminuye cuando los atributos no son independientes, y
- (ii) No puede ocuparse de atributos continuos no parametrizados.

En este trabajo proponemos un método que aprende un clasificador bayesiano simple que considera la discretización de atributos continuos y la búsqueda de atributos que no son independientes. El método incluye dos fases, el de discretización y el de la mejora estructural, que se repiten alternativamente hasta que la exactitud de la clasificación no puede ser mejorada. La discretización se basa en el principio descripción de longitud mínima (MDL), donde el número de intervalos que minimiza el MDL se obtiene por cada atributo. Para tratar con atributos dependientes y atributos irrelevantes, aplicamos un método de mejora estructural, que elimina y/o une atributos, basado en medidas de información mutua y condicional. En el método de aprendizaje propuesto, los principales puntos que se consideran son los siguientes:

- Permite considerar, mediante modelos sencillos (clasificador bayesiano simple), las principales relaciones entre los datos.
- Cuando se trata de variables continuas, con base a la discretización de éstas, permite determinar cuál es la mejor forma de dividir en intervalos esta variable continua, de tal manera que maximiza el rendimiento del clasificador bayesiano simple.
- Permite mejorar la estructura conservando la simplicidad de la misma mediante las operaciones de eliminación y unión de variables, considerando eliminar variables que no son relevante y la unión de variables que son condicionalmente dependientes.
- Realiza el aprendizaje parámetrico de la estructura completa, considerando el nodo clase, las variables discretas y continuas incluidas originalmente en el modelo y las variables derivadas por unión.

5.2. Metodología

El método de aprendizaje considera (i) la discretización de variables continuas, (ii) la selección de atributos relevantes y (iii) la eliminación o combinación de atributos dependientes para construir un clasificador bayesiano simple.

El método obtiene:

- La estructura del clasificador.
- El *mejor* número de intervalos para los atributos continuos.
- Los parámetros asociados (tablas de probabilidad condicional para cada atributo y probabilidades a priori para el nodo clase).

En el método propuesto se utiliza un método de discretización con base al mejoramiento de la clasificación. La idea es aplicarlo y evaluar el efecto que tiene la discretización en la búsqueda de la mejor estructura. Para la creación del clasificador bayesiano simple se usa el algoritmo incluido en [Lacave and Díez, 2005] y para obtención de la mejor estructura se incluyen algunas variaciones al método de [Pazzani, 1996]. Este método ha obtenido buenos resultados en la clasificación al aplicar las operaciones de eliminación y unión de variables. El algoritmo básico (algoritmo 5.1) del método de aprendizaje de clasificadores bayesianos estáticos consta de 4 etapas que podemos observar en la figura 5.1. Los detalles de cada etapa del algortimo se describen a continuación.

5.2.1. Etapa I: Inicialización

Esta etapa consiste básicamente de tres pasos (figura 5.2), requeridos para la creación de la estructura estática completa del clasificador bayesiano simple que será la base para el método de aprendizaje. Este paso se realiza solamente una vez a través del algoritmo 5.2, que considera una partición inicial (dos intervalos de igual tamaño) para todos los atributos continuos, para la creación del clasificador inicial y sus parámetros, los cuales se aprenden de los datos de entrenamiento.

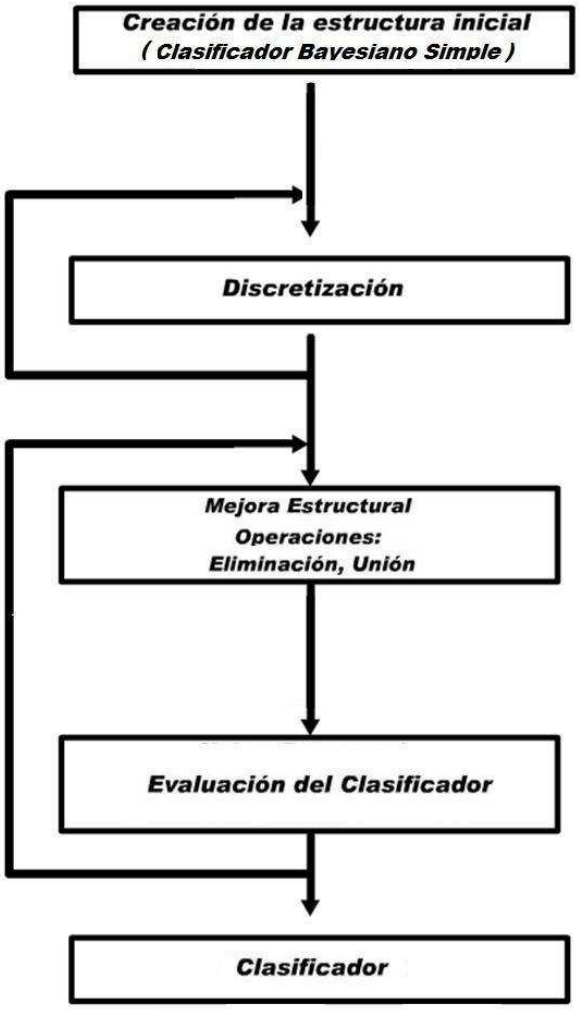


Figura 5.1: Modelo de aprendizaje de clasificadores bayesianos estáticos. Los recuadros marcan las principales etapas del modelo.

Algoritmo 5.1 Algoritmo ACBE (Aprendizaje de Clasificadores Bayesianos Éstáticos.)

1. **Etapa I: Inicialización**
   a. Define la estructura y la discretización inicial.
   b. Estimación de parámetros para la estructura inicial.

2. **Etapa II: Discretización**
   a. Mejora de la discretización con base al MDL.
   b. Evaluación de la estructura con base al MDL.

3. **Etapa III: Mejora Estructural**
   a. Mejora de la estructura basada en la eliminación y combinación de variables
   b. Evaluación de la estructura con base a la clasificación

4. **Etapa IV: Evaluación**
   a. Evaluar el clasificador con datos de prueba.

Los pasos 2 y 3 se repiten mientras la estructura-discretización no pueda ser mejorada.

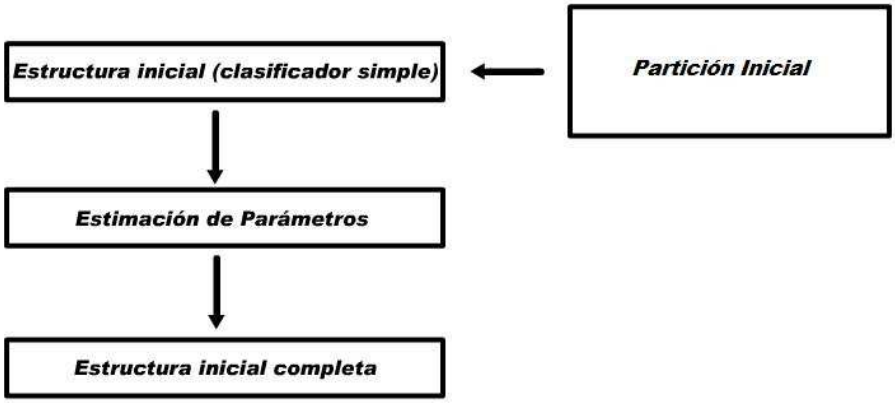


Figura 5.2: Etapa 1: Inicialización del clasificador bayesiano simple estático.

Algoritmo 5.2 Algoritmo Inicialización ACBE.

1. Para las variables continuas incluidas en el modelo se realiza una discretización inicial (dos intervalos de igual tamaño).
2. Creación de la estructura inicial que consiste en un clasificador bayesiano simple.
3. Estimación de parámetros: una vez calculados los parámetros asociados a cada nodo se considera que la estructura inicial está completa.

5.2.2. Etapa II: Discretización

Partiendo de la estructura actual, en esta etapa la discretización de cada atributo continuo es evaluada en función del mejoramiento de la estructura a través de una medidad de calidad. A partir de una discretización inicial uniforme generamos particiones adicionales basadas en el principio MDL. Cada atributo es procesado independientemente, en cada iteración se toma un atributo dividiendo el intervalo, se calcula el MDL considerando la nueva partición de la rama y se compara con el MDL obtenido considerando la partición anterior, el proceso se repite en forma iterativa hasta que éste no puede ser mejorado o no mejora en un porcentaje determinado. Para entender el método veremos a continuación como funciona el principio MDL. La figura 5.3 muestra esta etapa.

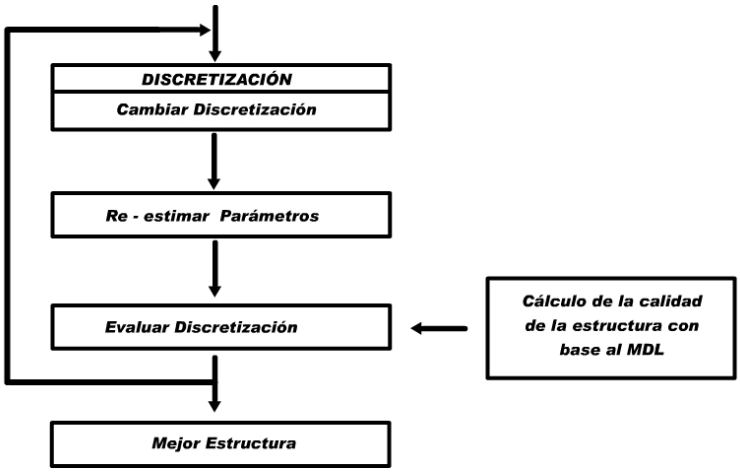


Figura 5.3: Etapa 2: Discretización del clasificador bayesiano simple estático.

Principio MDL

El principio MDL (por sus siglas en inglés: *Minimum Description Length*) [Lam and Bacchus, 1994] es un método que usa el principio de descripción de longitud mínima de Rissanen [Rissanen, 1978].

El principio MDL está basado en la idea de que el mejor modelo de una colección de elementos de datos es el modelo que minimiza la suma de la longitud de la codificación del modelo y la longitud de la codificación de los datos dado el modelo. Es decir que este método tiende a preferir estructuras menos complejas en lugar de estructuras densamente conectadas, estableciendo un compromiso entre exactitud y complejidad. Para aplicar el principio MDL a las redes bayesianas, [Lam and Bacchus, 1994] especifican como evaluar las dos representaciones, la de red misma y la de los datos. Considerando que para representar una red bayesiana en particular se necesita:

- Una lista de los padres de cada nodo.
- Un conjunto de probabilidades condicionales para cada nodo dados sus padres, requeridas para paramétrizar la red.

Suponiendo que se tienen *n-nodos* en el dominio de un problema, para un nodo con *K-padres* se requerirían *K-*log_2*(n)* bits para listar a sus padres y para representar las probabilidades condicionales, asociadas con la estructura, la longitud de la representación será el producto del número de bits requerido para almacenar el valor numérico de cada probabilidad condicional por el número total de probabilidades condicionales que son requeridas. En una red bayesiana, una probabilidad condcional es requerida para cada diferente instanciación de los nodos padres y del mismo nodo. Por ejemplo, si un nodo que puede tomar 5 valores distintos tiene 4 padres donde cada uno de los cuales puede tomar 3 distintos valores, se necesitarían 3^4**(5-1)* probabilidades condicionales.

Por lo que, bajo las consideraciones anteriores, la longitud de descripción total para una red está dada por:

$$\sum_{i=1}^{n} \left[K_i log_2(n) + d(S_i - 1) \prod_{j \in F_i} S_i \right] \tag{5.1}$$

Donde:

- n= número de nodos.
- K_i= número de nodos padres para cada nodo i.
- S_i= número de valores que puede tomar cada variable.
- F_i= el conjunto de padres de la variable i.
- d = número de bits requeridos para almacenar un valor numérico.

Donde para un dominio de un problema en particular n y d son constantes.

La longitud de la representación del modelo se considera un factor para determinar que tan conveniente resulta una estructura de RB, pero no es el único factor determinante en la especificación de la red, también se debe considerar la longitud de la representación de los datos a partir del modelo.

Al considerar la longitud de la representación de los datos, en una estructura de red bayesiana, se involucra el concepto de distribución conjunta ya que la especificación de la estructura de datos es una tarea de aprender la distribución conjunta de una colección de variables aleatorias X=X_1,...,X_n. Donde cada variable X_i tiene asociada una colección de valores X=x_{i1},...,x_n que pueden tomar, el número de valores dependerá en general de i. Cada opción distinta de valores para todas las variables en X define un evento atómico en la distribución conjunta básica y es asignada a una probabilidad particular en esta distribución.

Para una colección de N puntos de datos donde se requiere representarlos como una cadena binaria, existen varias maneras en las cuales se puede hacer, una opción es usar el MDL, donde únicamente se busca la representación de códicos de caracteres. Con códigos de caracteres cada evento es asignado a una cadena binaria única. Cada punto es convertido a su código de caracteres y los *n-puntos* son representados por cadenas for,adas por la concatenación de esos códigos de caracteres unidos.

El problema que se presenta ahora es ¿cómo hacer la comparación de esta longitud codificada con la longitud original, si no tenemos acceso a las probabilidades reales?. Lam y Bacchus utilizan el concepto de entropía cruzada, que se define como:

Entropía cruzada: Dado dos distrubuciones, P y Q, definidas sobre un mismo espacio de un evento, e_1,...,e_i. Se tienen dos probabilidades asignadas p_i por P y q_i por Q. La medida de *entropía cruzada* de [Kullback and Leiber, 1951] es una propuesta de la relación entre dos diferentes distribuciones definidas sobre el mismo espacio del evento. En particular, la *entropía cruzada* entre P y Q, *C(P,Q)*, esta dada por la ecuación 5.2.

$$C(P,Q) = \sum_{i=1}^{t} p_i(log_2(p_i) - log_2(q_i)) \tag{5.2}$$

La cual es utilizada para evaluar los posibles modelos de red, de acuerdo al teorema 5.1.

Teorema 5.1. *La longitud codificada de los datos se incrementa monotónicamente en función de la entropía cruzada entre la distribución definida por el modelo y la distribución real (demostración en [Lam and Bacchus, 1994]).*

Este teorema muestra que en lugar de usar la longitud de los datos codificados para evaluar los posibles modelos, se podría usar la evaluación de entropía cruzada, realizándose de una manera computacionalmente factible. El Teorema 5.1. también muestra que en cierto sentido el principio MDL puede ser visto como una generalización de trabajos previos de [Chow and Liu, 1968]. Por lo que Lam y Bacchus proponen una extensión de este método, para casos generales definiendo un nuevo peso para cada nodo X_i, con respecto a un conjunto arbitrario de padres que esta dado por:

$$W(X_i, FX_i) = \sum_{X_i, FX_i} P(X_i, FX_i) log_2 \frac{(X_i, FX_i)}{P(X_i)P(FX_i)} \tag{5.3}$$

donde se asumen todos los posibles valores que X_i y sus padres F_x pueden tomar.

Lam y Bacchus relacionan la medida de entropía cruzada con el teorema de Chow y Liu por medio del siguiente teorema:

Teorema 5.2. *C(P,Q) decrementa monotónicamente en función de:*

$$\sum_{i=1,Fx_i\neq 0}^{n} W(x_i, Fx_i) \tag{5.4}$$

De aquí que esto será minimizado sí y solo si la suma es maximizada.

Lam y Bacchus realizan entonces el cálculo de información mutua para hacer factible la medición MDL en forma local de la RB. Es decir, la búsqueda es guiada por el decremento de MDL local calculado al modificar localmente la estructura (agregar, eliminar o invertir un arco) y no es necesario calcular el MDL total de la estructura. El algoritmo también cuenta con un procedimiento de orientación automática de enlaces, cuando no se conoce el ordenamiento de las variables. El algoritmo de orientación de enlaces está basado en la idea de seleccionar la orientación que más decremente el MDL local de los nodos afectados por el enlace a agregar.

En este trabajo aplicamos el principio MDL a una estructura conocida: a un clasificador bayesiano simple, por lo que la búsqueda de la mejor estructura se reduce. Entonces, el principio MDL lo aplicamos a esta estructura fija y nos enfocamos a evaluar cada nodo continuo del clasificador con base al número de intervalos; es decir, lo aplicamos al problema de discretización.

Método Propuesto

Como vimos, la medida MDL hace un compromiso entre la exactitud y la complejidad de la estructura. Nosostros usamos ese mismo criterio para discretizar; es decir, evaluamos la estructura cada vez que realizamos una partición en cada atributo continuo. La medida que utilizamos para evaluar este equilibrio entre exactitud y complejidad, es similar a la que esta propuesta dentro de [Martinez and Sucar, 1998], la cual estima la exactitud midiendo la información mutua entre los atributos y la clase; y la complejidad contando el número de parámetros. Una constante α, en $[0,1]$, se utiliza para balancear el peso de cada aspecto, exactitud contra complejidad. Así, la medida de calidad está dada por:

$$Calidad_{Red} = (1-\alpha) * \left(1 - \frac{Longitud_{Red}}{Longitud_{RedMax}}\right) + (\alpha) * \left(\frac{Peso_{Red}}{Peso_{RedMax}}\right) \tag{5.5}$$

donde:

- $Calidad_{Red}$: Medida de equilibrio entre la complejidad y la exactitud de la estructura (ec. 5.5)
- $Longitud_{Red}$: Mide el tamaño de la estructura actual con base a la (ec. 5.1), donde se considera el número de variables en la estructura y el número de parámetros por cada variable.
- $LongitudMax_{Red}$: Define un tamaño máximo que puede tomar la estructura inicial con base a la (ec. 5.1), donde se considera un número máximo de variables en la estructura y un número máximo de parámetros por cada nodo.
- $Peso_{Red}$: Mide el peso de la estructura actual con base a la (ec. 5.3), donde se considera la IM entre las variables y el nodo clase.
- $PesoMax_{Red}$: Define un peso máximo que puede tener la estructura inicial con base a la (ec. 5.3), donde se considera un número máximo de variables en la estructura y un número máximo de parámetros por cada nodo, siendo el mismo número considerado para calcular la longitud máxima.

Algoritmo 5.3 Discretiza ACBE

Entrada: NB
Salida: NB mejor discretización

1. Calcula la **Calidad** de la estructura.
2. Considerar todas las variables continuas.
3. Considerar discretización inicial con 2 intervalos.
4. Inicia Ciclo
   a. Tomar la primer variable continua.
   b. Evaluar la calidad de la rama: $Calidad_Actual$.
   c. Inicia Ciclo
      - Particionar en forma temporal cada intervalo.
      - Calcular la calidad considerando cada partición
      - Si es mejor calidad
        - a) Agregar partición en forma definitiva
        - b) $Calidad_Actual = Mejor_Calidad$.
      - Si no es mejor calidad
        - No se considera la partición

   d. Finaliza Ciclo (hasta que no mejore la calidad).
5. Finaliza Ciclo (hasta que no haya variables).

Un valor α=0.5 da la misma importancia a complejidad y exactitud, mientras que α cercano a 0 considera darle mayor importancia a la complejidad y α cercano a uno mayor importancia a la exactitud. La *longitud máxima* es estimada considerando el número máximo de intervalos por atributo (dependiendo de la distribución de los datos). El peso máximo, corresponde a la sumatoria de información mutua de la estructura completa, que se calcula a partir de la *longitud máxima*. Esto es con base a un número máximo de valores, N_{Max}, que es dado por el usario. Al final de esta etapa se obtiene la mejor discretización para la estructura actual.

La ec. 5.6 se aplica a cada rama de la estructura que con atributos discretos y está dada por:

$$Calidad_{Rama} = (1-\alpha) * \left(1 - \frac{Longitud_{Rama}}{Longitud_{RamaMax}}\right) + (\alpha) * \left(\frac{Peso_{Rama}}{Peso_{RamaMax}}\right) \tag{5.6}$$

El método tiene, entonces, 2 parámetros seleccionados por el usuario:

- α cuyo valor significa el peso (importancia) que se le da a la exactitud y a la complejidad.
- N_{Max} que es valor máximo de posibles valores para una variable a discretizar, considerado para determinar la $Longitud_{RamaMax}$.

. Estos valores dependen de la aplicación particular, y como veremos en la sección de resultados, no son críticos para la operación adecuada del método. El procedimiento de discretización se detalla en el algoritmo 5.3.

5.2.3. Etapa III: Mejora estructural

Dado la actual discretización (estructura que obtiene el mejor MDL de la etapa anterior). En esta fase la estructura es mejorada eliminando atributos superfluos y eliminando o combinando atributos dependientes. Esta fase considera el algoritmo *Poda* (5.4), para realizar la mejora estructural. La descripción de la etapa la podemos observar en la figura 5.4.

El algoritmo *Poda*, inicialmente considera todas las variables para calcular la IM (información mutua) entre cada variable y el nodo clase, eliminando aquellas variables que no son relevantes para el clasificador. En seguida se realiza el cálculo de la IM entre pares de variables dado el nodo clase. De acuerdo a ese valor los pares de variables son ordenados en forma descendente. Orden considerado para realizar el proceso de *eliminación* y *unión* de variables. Para cada par de variables se prueban las dos operaciones, y se realiza la operación que mejore en mayor porcentaje la exactitud del clasificador o en el peor de los casos no lo afecte negativamente, si se obtiene un porcentaje similar al probar las dos operaciones, entonces se prefiere la operación de *eliminación*, debido a que simplifica la estructura. Para evaluar las operaciones se compara el porcentaje de clasificación del modelo obtenido después de cada operación y el modelo anterior (inicialmente se considera el original, posteriormente se considera el resultante de la operación anterior). Este proceso se repite hasta que no haya más atributos superfluos o dependientes, mientras se mejore el resultado de la exactitud del clasificador. Cuando existen aplicaciones que involucran una gran cantidad de atributos, y el número de pares de variables a considerar es muy grande, se considera utilizar un umbral (definido por el usuario) para evaluar únicamente *N* relaciones.

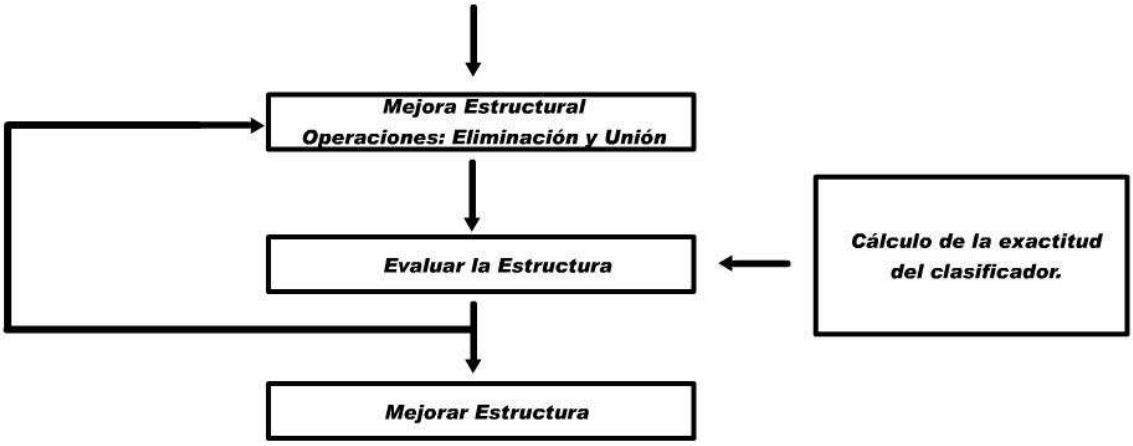


Figura 5.4: Fase 3: Mejora estructural del clasificador bayesiano estático.

5.2.4. Etapa IV: Evaluación

Esta etapa consiste en la evaluación de la exactitud del clasificador bayesiano con los datos de prueba (diferentes del que está usado para el entrenamiento). Si el resultado no es satisfactorio se repiten las etapas III y IV.

En este trabajo se aplica la metodología en dos áreas: clasificación de piel en imágenes, y detección de cáncer cervical. Para el primer caso, se evalúa la capacidad de reconocer pixeles de piel y no piel en imágenes, usando 3 modelos de color. Para el segundo caso, consideramos el diagnóstico de cáncer basado en análisis de imágenes de colposcopia, usando 3 modelos matemáticos. Los experimentos y resultados se presentan en el capítulo 7.

Algoritmo 5.4 Poda ACBE

Entrada: NB mejor discretización
Salida: NB mejor estructura
Inicio:

1. Considerar todas las variables
   a. Calculo de la IM (información mutua) entre la variable y el nodo clase
   b. Ordenamiento de la IM (Descendente)
   c. Considerar "*N*" ramas abajo de un umbral (dado por el experto)
   d. Eliminar atributos que no proporcionan infornación
2. Considerar atributos restantes
   a. Calculo de la IM (información mutua) entre pares de variables dado el nodo clase
   b. Ordenamiento de la IM (Descendente)
3. Clasificación en el modelo original
   a. Obtención del porcentaje de clasificación
4. Inicia ciclo
   a. Tomar la primer rama (IM mayor)
   b. Realizar las operaciones en forma temporal sobre la estructura permanente
      - i) Operación Elimina var1: Si existe la var 1 en el modelo entonces Eliminación de la var1, De lo contrario no se realiza nada
      - ii) Operación Elimina var2: Si existe la var 2 en el modelo entonces Eliminación de la var2, De lo contrario no se realiza nada
      - iii) Operación une var1 y var2: Si existen las 2 variables (var1 y var 2) entonces Unión de la var1 y var2, De lo contrario no se realiza nada

   c. Se evalúa la clasificación para cada estructura temporal resultante
      - i. Si hay una mejor clasificación entonces: Se compara con la obtenida anteriormente del modelo permanente
      - Si es mayor entonces
      - Realiza la operación permanentemente
      - Si no mejora la clasificación
      - No se hace nada y se continua con el mismo modelo permanente
      - ii. Si existe mas de una mejor clasificación (iguales) entonces: Se compara con la obtenida anteriormente del modelo permanente
      - Si es mayor entonces
      - Realiza la primera operación permanentemente
      - Si no mejora la clasificación
      - No se hace nada y se continua con el mismo modelo permanente
        d. Se obtiene el nuevo modelo permanente
      - Hasta que no haya ramas

   El paso 4 se realiza hasta que no haya pares de atributos dependientes dado el nodo clase (de acuerdo a un umbral establecido).

5.3. Resumen

Se ha presentado una nueva metodología para aprendizaje de clasificadores bayesianos simples. El método incluye dos fases, la discretización y la mejora estructural, que se repiten alternativamente hasta que la exactitud del clasificador no puede ser mejorada. La fase de la discretización se basa en el principio de descripción de longitud mínima (MDL), donde el número de intervalos que reduce al mínimo el MDL se obtiene por medio de una medida de calidad de la estructura. La fase de la mejora estructural elimina y/o une pares de atributos, esta mejora se basa en la medida de información mutua condicional. La etapa de la mejora estructural utiliza reglas de decisión simples, considerando sólo ciertos pares de atributos. Sin embargo, aunque el proceso de búsqueda esta guiado considerando aquellas ramas que son más fuertes, el proceso de ejecutar las operaciones depende en gran medida del orden en que se toman las variables. Las principales aportaciones de ésta metodología es la integración de los procesos de *mejora estructura* y *discretización*. Así como la variante del principio MDL para discretización. En el siguiente capítulo esta técnica es ampliada para aprender clasificadores bayesianos dinámicos y considerar datos incompletos.

Capítulo 6

Aprendizaje de clasificadores bayesianos dinámicos

En el capítulo 5 se trató el aprendizaje de clasificadores estáticos. En este capítulo ampliamos el método para aplicarse a dominios dinámicos y con información incompleta (donde el nodo clase se desconoce). Describimos el modelo completo, que involocra, además de las etapas de discretización y mejora estructural, las etapas para la determinación de información incompleta y cálculo del mejor número de estados para el nodo clase oculto.

6.1. Introducción

Un clasificador bayesianos dinámico está compuesto, al igual que los estáticos, de una estructura gráfica y un conjunto de parámetros(figura 6.1). En este clasificador también se especifica una distribución conjunta sobre las variables aleatorias. Se realizan las mismas consideraciones que para una RBD:

- Suposición 1. Los procesos son markovianos: es decir, las variables del periodo de tiempo actual son condicionalmente independientes de las variables de periodos de tiempo pasados, dadas las variables del tiempo inmediatamente anterior.
- Suposición 2. Los procesos son estacionarios: es decir, que las probabilidades de transición (entre un periodo de tiempo y otro) son independientes del tiempo.

Los clasificadores bayesianos dinámicos, al igual que las RBD, se constituyen de dos partes:

- i) Un clasificador inicial *Bo* que especifica la distribución sobre el estado inicial *X[0]* y
- ii) Una red de transición $B\rightarrow$ sobre las variables $X[0] \cup X[1]$ que es tomada para especificar la probabilidad de transición $P(X[t+1]\ left|[t])$ para todo *t*.

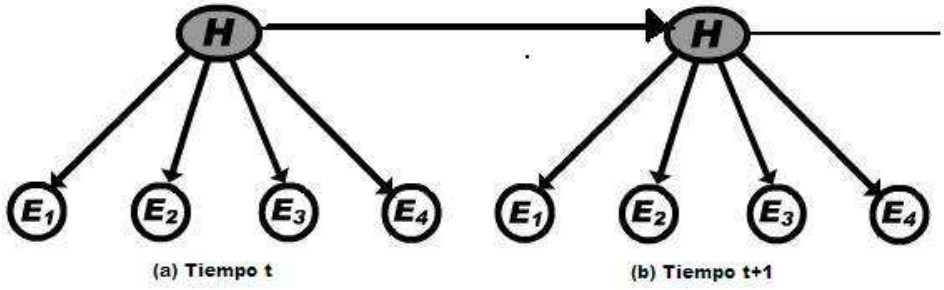


Figura 6.1: Clasificador bayesiano dinámico.

Cuando se trata de clasificadores bayesianos simples dinámicos, la red de transición está compuesta únicamente por una trayectoria que liga la estructura *t* con la estructura *t+1*, como se muestra en la figura 6.1.

Los clasificadores bayesianos han sido aplicados a diversos dominios que involucran tiempo, como en el reconocimiento de actividades humanas (gestos), procesamiento de voz, diagnóstico médico, bionformática y otras aplicaciones; donde es conveniente manejar secuencias de tiempo ya que involucran una actividad duradera donde se desconoce la información que pueda tener a través del tiempo. El problema que se presenta es que al involucrar el tiempo la complejidad del procesamiento crece, aunado a dominios que manejan gran cantidad de variables y grandes bases de datos, el aprendizaje se dificulta aún más que en los clasificadores bayesianos estáticos. Los clasificadores bayesianos dinámicos pueden llevar a cabo la clasificación por medio de los algoritmos habituales de inferencia que se aplican a clasificadores bayesianos estáticos. Sin embargo, cuando se trata de observaciones parcialmente observadas, principalmente cuando no se conoce el nodo a predecir (nodo clase) nos encontramos con que en muchos casos la determinación de los mejores estados se encuentran dados por el experto con base a su experiencia.

Los clasificadores bayesianos dinámicos son una generalización de los modelos ocultos de Markov (HMM), donde se desconoce el nodo de observación. El nodo de estado representa la evolución temporal y es oculto debido a que se desconocen los posibles estados para ese nodo. Los HMM se componen de dos procesos estocásticos, uno de éstos determina la transición entre los estados y es no observable, el otro proceso, genera observaciones de salida para cada estado.

Al igual que cuando los datos son completos, cuando se trata de atributos ocultos, los parámetros asociados al nodo deben ser estimados. Un método para esto es el algoritmo EM [Dempster et al., 1997; Lauritzen, 1995] para datos faltantes. El algoritmo EM (visto en el capítulo 3) es un procedimiento iterativo que busca la hipótesis de máxima verosimilitud. El procedimiento es repetidamente ejecutado en dos fases: Paso E y paso M. Una vez calculados los parámetros asociados a cada nodo se considera que la estructura está completa.

Este trabajo parte principalmente de estos problemas comentados anteriormente (manejo de: datos continuos, información incompleta, dependencia entre variables y procesos dinámicos), por lo cual proponemos un método para obtener un clasificador bayesiano simple dinámico que, incluye además de las etapas del clasificador estático, el manejo y determinación del nodo clase oculto mediante el Algoritmo EM. Éste es un proceso cíclico que evalúa estructuras a través de una medida de calidad (con base al MDL) para diferentes números de estados, obteniendo el *mejor* número de estados para el nodo clase. De esta forma se obtiene la *mejor* estructura para realizar la clasificación dinámica. Esto permite:

- Determinar el número de estados convenientes para los nodos clases ocultos.
- Que se realice el aprendizaje parámetrico de la estructura completa, considerando el nodo clase oculto, las variables discretas y continuas incluidas originalmente en el modelo y las variables derivadas por unión.

En ambos casos se busca además conservar la estructura básica que permite considerar mediante modelos sencillos las principales relaciones entre los datos.

6.2. Metodología

El método propuesto considera (i) la determinación del nodo clase oculto, (ii) la discretización de variables continuas, (iii) la selección de atributos relevantes, (iv) la eliminación o combinación de atributos dependientes y (v) la evaluación dinámica, para aprender un clasificador bayesiano simple dinámico.

El método obtiene:

- La estructura del clasificador.
- Un número de intervalos, para los atributos continuos, que contribuye a la mejora estructural.
- Un número de estados, para el nodo clase oculto, que contribuye a la mejora estructural.
- La selección de los *mejores* atributos para el modelo: eliminando atributos irrelevantes y eliminando/uniendo atributos dependientes.
- Los parámetros asociados (tablas de probabilidad condicional para cada atributo y probabilidades a priori para el nodo clase).

El proceso de discretización de variables continuas es el mismo proceso que en el proceso de aprendizaje de NB estáticos óptimos. Para la creación del clasificador bayesiano simple se usa el algoritmo incluido en [Lacave and Díez, 2005] y para obtención de la mejor estructura un algoritmo similar al del clasificador bayesiano.

El algoritmo básico muestra los pasos para el proceso de aprendizaje de clasificadores bayesianos dinámicos (ACBD), que consta de 5 etapas (algoritmo 6.1) cada una de las cuales se describen a continuación.

6.2.1. Etapa I: Inicialización

Esta etapa consiste básicamente de cuatro pasos, requeridos para la creación de la estructura estática del clasificador bayesiano simple que será la base para el método de aprendizaje. Éste paso se realiza solamente una vez y se consideran todos los atributos (estructura completa) y una partición inicial para las atributos continuos (dos intervalos de igual tamaño) y dos estados para el nodo clase (considerando que el número m´nimo de posibles valores que pueden tener). De acuerdo con esta estructura inicial, los parámetros se aprenden de los datos de entrenamiento.

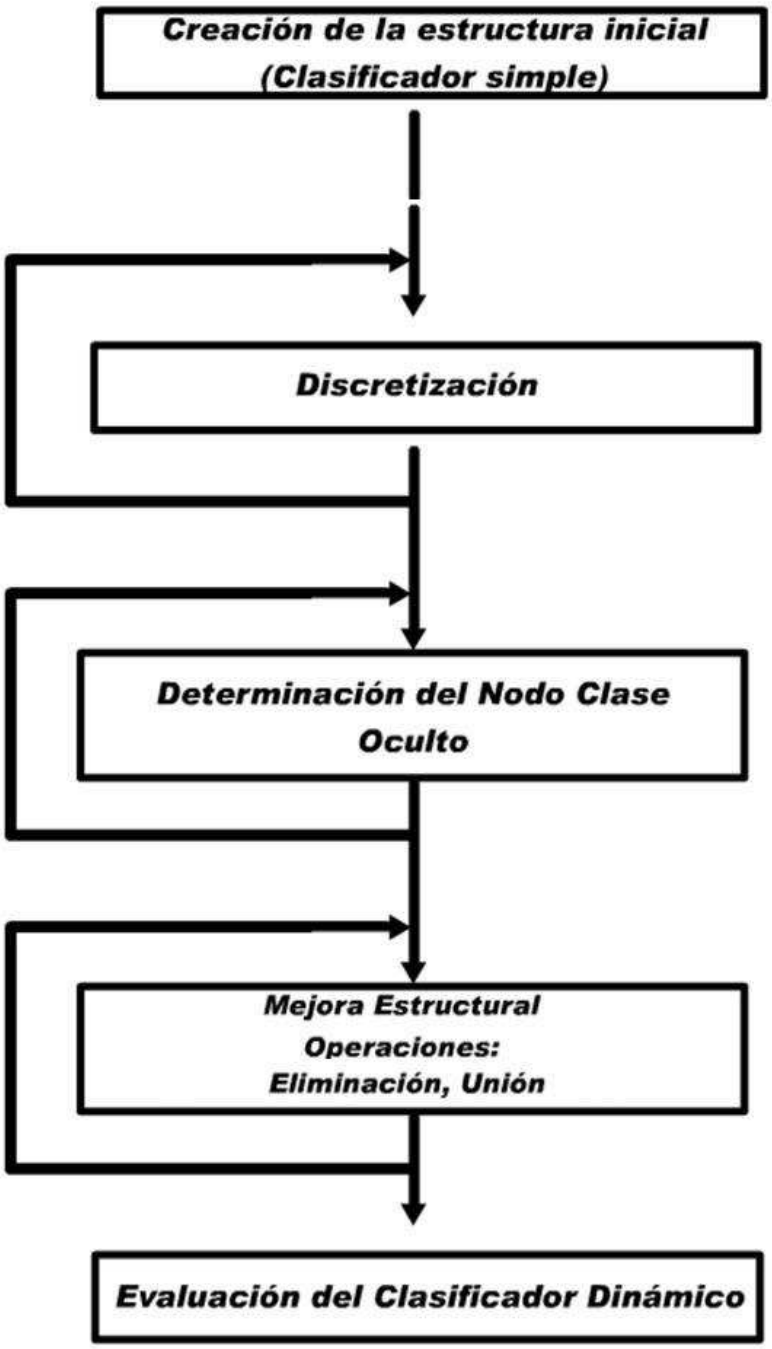


Figura 6.2: Modelo de aprendizaje de clasificadores bayesianos dinámicos.

Algoritmo 6.1 ACBD (Aprendizaje de Clasificadores Bayesianos Dinámicos)

1. **Etapa I: Inicialización**
   a. Define la estructura y la discretización inicial.
   b. Estimación de parámetros para la estructura inicial
2. **Etapa II: Discretización**
   a. Mejora de la discretización con base al MDL
   b. Evaluación de la estructura
3. **Etapa III: Determinación del nodo clase oculto**
   a. Variación del número de estados (2...N)
   b. Evaluación de la estructura con base al MDL
4. **Etapa IV: Mejora Estructural**
   a. Mejora de la estructura basada en la eliminación y combinación de variables
   b. Evaluación de la estructura con base al MDL
5. **Etapa V: Evaluación Dinámica**
   a. Clasificación con datos de prueba.

Los pasos 2, 3 y 4 son repetidos iterativamente hasta que la estructura o discretización no pueda ser mejorada.

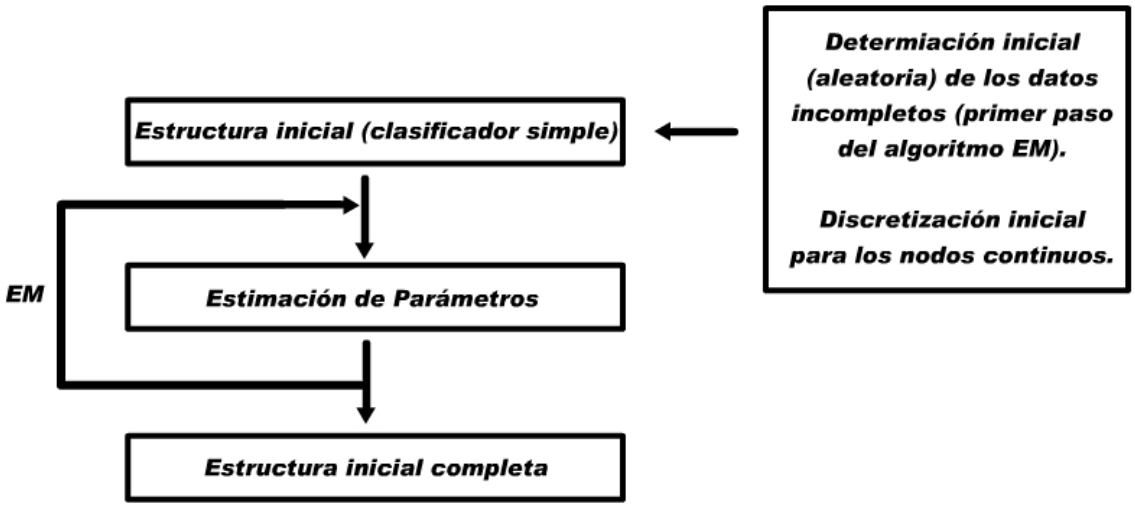


Figura 6.3: Etapa I: inicialización del clasificador bayesiano dinámico.

Algoritmo 6.2 Etapa I: Inicialización del método ACBD.

1. Discretización inicial: semilla inicial (dos intervalos de igual tamaño).
2. Determinación de datos faltantes (nodo clase oculto):
   a) Número inicial de estados (2 estados).
   b) Determinación de datos faltantes: Algoritmo EM.
3. Creación de la estructura inicial
4. Estimación de parámetros. Una vez calculados los parámetros asociados a cada nodo se considera que la estructura inicial está completa.

Algoritmo 6.3 Etapa III Determinación del nodo clase oculto), método ACBD.

1. Para un valor de α
   Repite con N:2 hasta N
   Se considera que el nodo clase puede tener de 2 hasta N estados
   - Se reestiman parámetros en cada ajuste del número de estados, aplicando el algoritmo EM.
   - Se evalua la estructura con base al MDL
2. Seleccionar el número de estados con mejor MDL.

6.2.2. Etapa II: Discretización

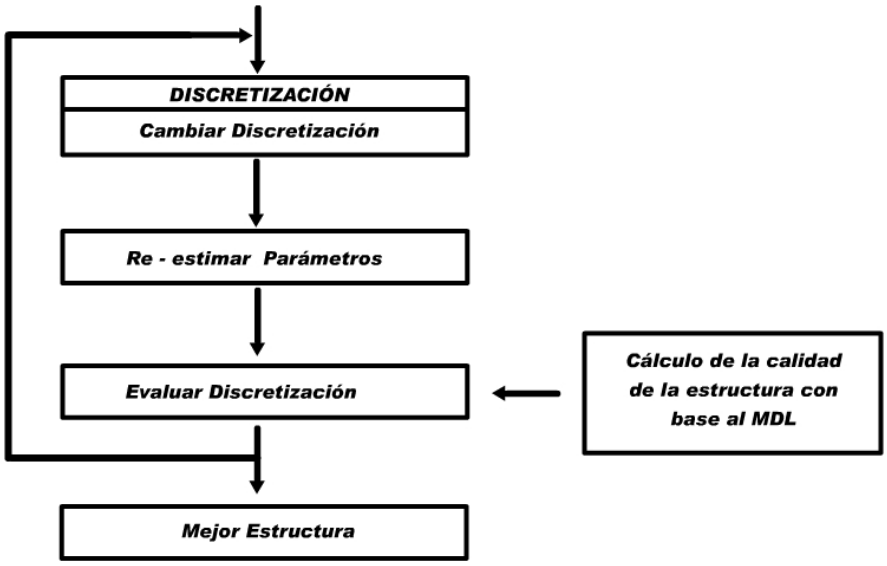


Figura 6.4: Etapa II: Discretización del clasificador bayesiano dinámico.

Esta etapa es similar a la etapa III, del algoritmo 5.3 (Discretiza ACBE).

6.2.3. Etapa III: Determinación del nodo clase oculto

Esta etapa consta de los pasos descritos en el algoritmo 6.3. Este es un proceso iterativo que busca el mejor número de estados para el nodo clase con base a una medida de calidad. Inicia con un número inicial de estados: de 2 hasta N posibles estados y cada estructura generada es evaluada con base al MDL.

6.2.4. Etapa IV: Mejora estructural

Dado la actual discretización (la que obtiene el mejor MDL de la etapa anterior), en esta fase la estructura es mejorada para eliminar atributos superfluos y eliminar o combinar atributos dependientes. Esta fase considera el algoritmo 6.4. El paso 3 se realiza hasta que no haya pares de atributos dependientes dado el nodo clase arriba del umbral establecido. Este proceso se repite hasta que no hay más atributos superfluos o dependientes.

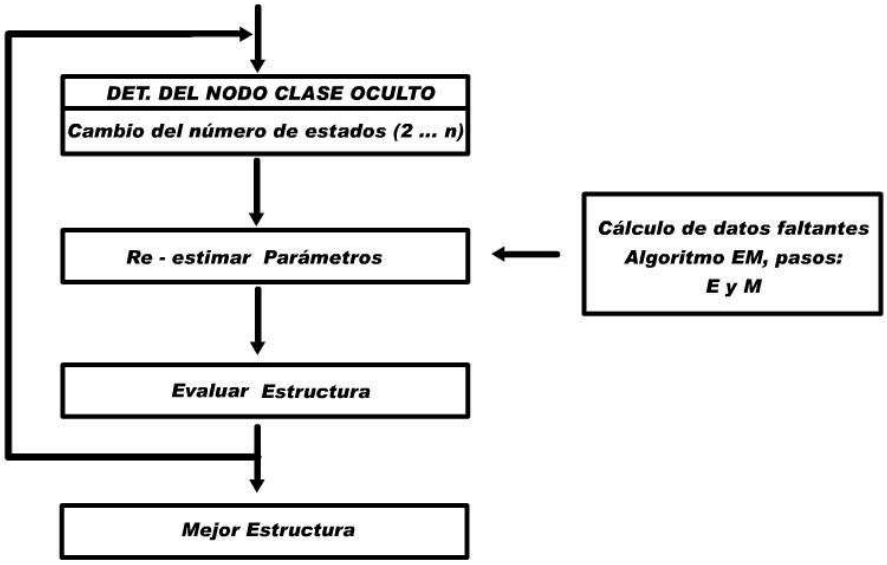


Figura 6.5: Etapa III: Determinación del nodo clase oculto del clasificador bayesiano dinámico.

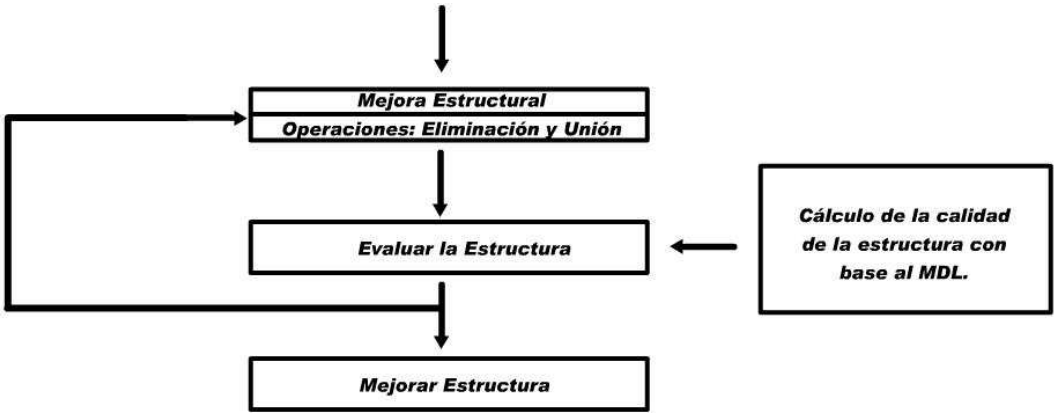


Figura 6.6: Etapa IV: Mejora estructural, clasificador bayesiano dinámico.

6.2.5. Etapa V: Evaluación Dinámica

Esta etapa consiste en la evaluación de la exactitud del clasificador bayesiano dinámico, con datos de prueba (diferente del que está usado para el entrenamiento).

Los clasificadores bayesianos dinámicos, se constituyen de dos partes, el clasificador inicial *Bo* (que se ha optenido a partir de las etapas anteriores) que especifica la distribución sobre el estado inicial y la red de transición $B\rightarrow$ sobre las variables que es tomada para especificar la probabilidad de transición entre los periodos de tiempo. Esta red de transición puede ser construida a partir de algún método en general de aprendizaje de RBD.

6.3. Resumen

Este trabajo parte principalmente de los problemas detectados en el aprendizaje de clasificadores bayesianos, por lo cual proponemos una nueva méetodología para obtener un clasificador bayesiano simple dinámico que, incluye además de las etapas del clasificador estático, el manejo y determinación del nodo clase oculto mediante el Algoritmo EM. Este es un proceso cíclico que evalúa estructuras a través de una medida de calidad (con base al MDL) para diferentes números de estados, obteniendo el mejor número de estados para el nodo clase. De esta forma se obtiene la mejor estructura para realizar la clasificación dinámica. En este capítulo presentamos un método de aprendizaje de clasificadores bayesianos simples dinámicos, que responde a las necesidades de aprendizaje de estructuras dinámicas que involucran nodos clase ocultos, además de cubrir las necesidades de manejo de datos continuos y de mejora estructural.

El método es un proceso iterativo que busca la mejor estuctura con base a una medida de calidad (MDL) y consta de 5 etapas: *Inicialización, Determinación del nodo clase oculto, Discretización, Mejora estructural y Evaluación dinámica*, las etapas están dadas según el orden de ejecución. Se parte de una estructura inicial (con un número inicial de inervalos y estados para el nodo clase oculto) y se le aplica el proceso de discretización (iterativamente hasta encontrar la mejor discretización), una vez que se tiene la estructura con mejor discretización, ésta sirve de entrada al proceso de determinación del mejor número de estados para el nodo clase oculto (proceso iterativo), posteriormente se realiza la mejora estructural (proceso iterativo que aplica las operaciones de unión y eliminación de atributos) y finalmente se evalua la clasificación dinámica.

Este método denominado ACBD (Aprendizaje de clasificadores Bayesianos Dinámicos) incluye 2 fases del método ACBE, las cuales son: la etapa de *discretización* y la etapa de *mejora estructural*. Siendo la primera igual en ambos métodos, y la segunda se diferencia en que las estructuras se evaluan no a través de la exactitud de la clasificación sino atraves de la medida de *calidad* basada en el principio MDL. Esta medida es aplicada también para encontrar la mejor estructura en la determinación del mejor número de estados para el nodo clase oculto.

Una vez que se han realizado las primeras 4 etapas se considera que tenemos la estructura del clasificador inicial *Bo* y la red de transición $B\rightarrow$ es construida usando algún método general de RBD. Finalmente se evalúa el clasificador bayesiano dinámico en el dominio de interés.

Las principales aportaciones de este trabajo son: un método integral para aprender clasificadores bayesianos dinámicos y la metodología para determinar el número de estados para el nodo clase oculto.

Algoritmo 6.4 Etapa IV:Mejora estructural del método ACBD.

Entrada: NB mejor discretización
Salida: NB mejor estructura
Inicio:

1. Considerar todas las variables
   a. Calculo de la IM (información mutua) entre la variable y el nodo clase
   b. Ordenamiento de la IM (Descendente)
   c. Considerar "*N*" ramas abajo de un umbral (dado por el experto)
   d. Eliminar atributos que no proporcionan infornación

2. Considerar atributos restantes
   a. Calculo de la IM (información mutua) entre pares de variables dado el nodo clase
   b. Ordenamiento de la IM (Descendente)

3. Inicia ciclo considerando pares de variables con nmayor IM
   a. Operaciones de eliminación y unión
   b. Calculo de la calidad de la red en el modelo original despues de cada operación.

      - Aqui no podemos realizar la clasificación, por lo que calculamos la calidad, que debe mejorar o permanecer igual, para lo cual debemos determinar la mejor alfa, por lo que probamos con los posibles valores de esta $(0-1)$.
      - Determinación de la mejor alfa
      - a) Inicia ciclo: cálculo de la calidad con alfa = 0.1,...,0.9
      - b) Si mejor calidad
      - c) Determinamos mejor alfa
      - d) Realiza operación

Nota: Podemos considerar varios caminos a partir de cuando haya mas de una mejor medida de calidad.

Capítulo 7

Pruebas y resultados

En capítulos previos hemos propuesto 2 nuevos métodos: aprendizaje de clasificadores bayesianos estáticos y aprendizaje de clasificadores bayesianos dinámicos. En este capítulo validamos los métodos al aplicarlos en problemas del mundo real. Evaluamos primero la clasificación resultante al aplicar el método estático a dos dominios: clasificación de piel y detección de cáncer cervical, y el método dinámico al reconocimiento de gestos. En ambos casos, comenzamos describiendo la metodología y los experimentos realizados, así como los datos utilizados en estos. Posteriormente, presentamos los resultados obtenidos en los experimentos y un análisis de los mismos. Terminamos con una sección dedicada a las conclusiones.

7.1. Clasificadores bayesianos estáticos

El aprendizaje incluye dos fases principales, la de discretización y la de la mejora estructural, que se repiten alternativamente hasta que la exactitud de la clasificación no puede ser mejorada. En las pruebas se evalúan ambas fases por separado. Se comparan los clasificadores obtenidos con nuestra metodología contra el clasificador bayesiano simple, el TAN y C4.5.

7.1.1. Metodología

El método se probó en dos aplicaciones:

- Clasificación de Piel en imágenes, usando 3 modelos de color, y
- Detección de Cáncer Cervical, usando 3 modelos matemáticos

Nuestro método aprende diferentes clasificadores Bayesianos que combinan los atributos de los tres modelos y obtiene un alto porcentaje de exactitud en las imágenes de pruebas. A continuación se da una descripción de estas aplicaciones.

7.1.2. Clasificación de piel

Un importante problema en visión computacional es discriminar piel y no piel en imágenes. Los modelos comúnmente usados para el reconocimiento de piel son: RGB, HSV y YIQ, entre otros.

- **Modelo RGB (*Red, Green, Blue*)**

El modelo RGB Es uno de los modelos más utilizados por los sistemas informáticos para reproducir los colores en el monitor y en el escáner. Está basado en la síntesis aditiva de las intensidades de luz relativas al rojo, al verde y al azul para conseguir los distintos colores; incluyendo el negro y el blanco.

- **Modelo HSV (*Hue, Saturation, Value*)**

El modelo HSV fue creado en 1978 y está pensado en la definición del color que realizaría un artista. Las siglas H, S y V corresponden a *tono* (*hue*), *saturación* (*saturation*) y valor (*value*) respectivamente. También se denomina HSB, siendo B el brillo (brighness).

- **Modelo YIQ (*Luminance, Inphase, Quadrature)*

Fue una recodificación realizada para la televisión americana (NTSC), la cual tenía que ser compatible con la televisión blanco y negro que solamente requiere del componente de iluminación. Los nombres de los componentes de este modelo son Y por iluminación (*Luminance*), I fase (*in-phase*) y Q cuadratura (*quadrature*). Estas últimas generan la cromaticidad del color.

En diferentes casos estos modelos son aplicados por separado en el reconocimiento de piel. Para nuestros experimentos realizamos una combinación de los tres modelos, misma que origina un modelo de color híbrido el cual fue probado con muy buenos resultados. En este caso se tienen (tabla 7.1) 9 atributos, todos continuos y 2 clases:

1. Piel y
2. No Piel

A continuación se describe el experimento realizado.

Datos

Corrimos nuestros experimentos con datos obtenidos de una aplicación real. Los datos se obtuvieron de los experimentos realizados en el reconocimiento de gestos por [Montero,2005]. Estos experimentos se realizaron sobre una base de 55 imágenes de piel y 50 de no piel. El tamaño de las imágenes es de 200 x 300. Las pruebas de clasificación y segmentación se realizaron con 10 imágenes de 500 x 500 (el Apéndice A describe la base de datos utilizadas para esta aplicación). Una muestra de estas imagenes de piel y no piel, las podemos ver en la figura 7.1. y 7.2, respectivamente.

Núm	Atributos	Valores
1	R	Continuo
2	G	Continuo
3	B	Continuo
4	H	Continuo
5	S	Continuo
6	V	Continuo
7	Y	Continuo
8	I	Continuo
9	Q	Continuo
10	Clase	Piel, NoPiel

Tabla 7.1: Atributos involucrados en el problema de reconocimiento de piel.

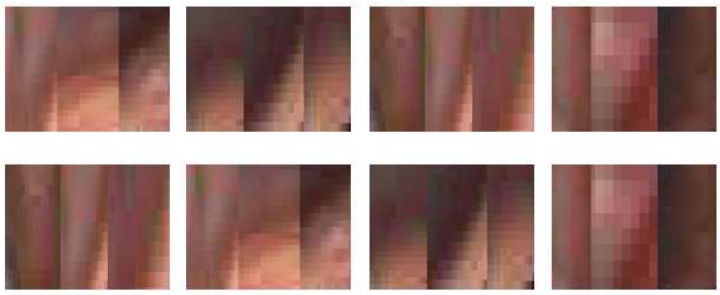


Figura 7.1: Muestras de la BD de piel

Evaluación

- Para evaluar la discretización se realiza una comparación entre la clasificación del modelo original contra el modelo con la mejor discretización.
- La mejora estructural se evalua comparando la exactitud del clasificador antes y despues de cada operación.

Las variables involucradas en los tres modelos están escaladas en forma similar por lo que presentan un rango de valores igual para todas, por lo tanto se uso una sola medida de cercanía para todas las variables en el mejoramiento de la estructura (criterio de paro).

Figura 7.2: Muestras de la BD de no piel

A continuación se describe el proceso de cada uno de los experimentos.

Resultados

En la tabla 7.2 se observan los resultados al probar el clasificador con la *mejor discretización* para diferentes valores de α. Se comprueba que el resultado de la clasificación baja conforme se le da mayor peso a la complejidad (longitud) y sube si le damos mayor peso a la exactitud $(\alpha = 0{,}9, 0{,}8)$.

α	Atributos	Exactitud %
0.9	RGBHSVYIQ	95
0.8	RGBHSVYIQ	95
0.5	RGBHSVYIQ	94
0.3	RGBHSVYIQ	93

Tabla 7.2: Resultados de la clasificación obtenida a partir de la *mejor* discretización probando con diferentes valores de α.

sec.	operación	Núm Atrib.	Atributos	Exactitud %
0		9	RGBHSVYIQ	95
1	Elim. B	8	RGHSVYIQ	95
2	Elim. Q	7	RGHSVYI	95
3	Elim. H	6	RGSVYI	96
4	Une RG	5	R-GSVYI	96
5	Elim. V	4	R-GYI	97
6	Elim. S	3	R-GY	98

Tabla 7.3: Proceso de *Mejora estructural* para la obtención del modelo de color en el reconocimiento de Piel.

La tabla 7.3, muestra las operaciones realizadas en el proceso de *mejora estructural* y el porcentaje de clasificación para el modelo después de cada operación, considerando un $\alpha = 0{,}8$. A continuación se describen estas operaciones:

1. Considerando inicialmente los 9 atributos, se realiza la eliminación del atributo B, esta operación no afecta el porcentaje de la clasificación. El nuevo modelo es: R, G, H, S, V, Y, I, Q.

2. Se elimina Q, que tampoco afecta el porcentaje de la clasificación, Por lo que el nuevo modelo es: R, G, H, S, V, Y, I.

3. La siguiente operación es la eliminación de H, ya que incrementa el porcentaje de exactitud a 96 %. Entonces el nuevo modelo esta compuesto por los atributos: R, G, S, V, Y, I.

4. Esta operación involucra la unión de R y G, lo cual no afectaba la exactitud de la clasificación. Quedando el modelo: $R - G, S, V, Y, I$.

5. Sobre este modelo se elimina V, y se obtiene una exactitud de 97 %, el modelo queda como: R-G,S,Y,I.

6. En esta operación se elimina el atributo S, lo cual no afecta la exactitud y se obtiene como modelo: R-G,Y,I.

7. Por último, se elimina el atributo I, con lo cual se incrementa la exactitud y se obtiene como modelo final : R-G,Y.

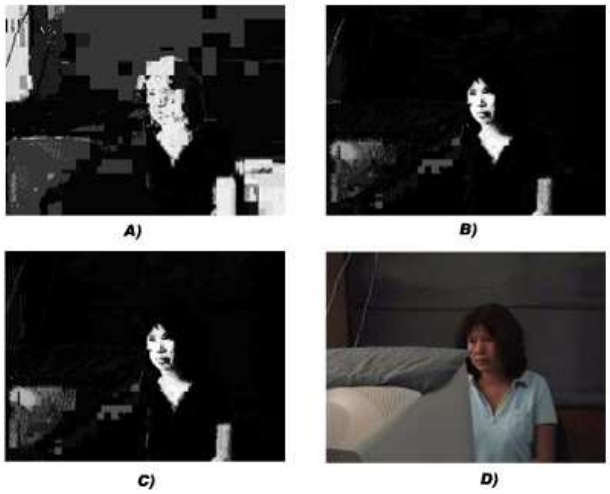


Figura 7.3: Experimento 1: Segmentación usando los modelos: (a) RGB normalizado, (b) R-GYI y (c) R-GY, (d) imagen original.

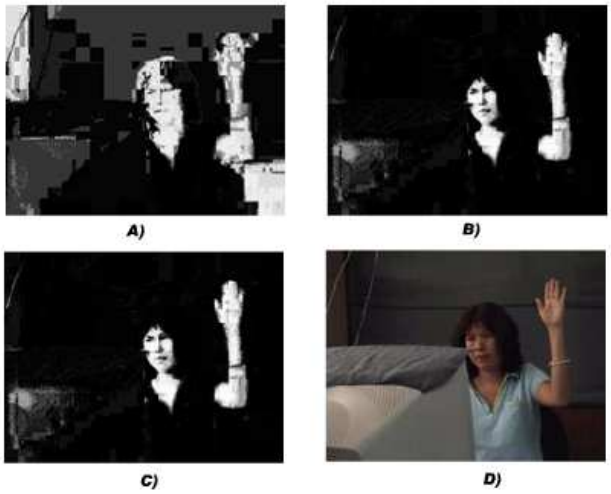


Figura 7.4: Experimento 2: Segmentación usando los modelos: (a) RGB normalizado, (b) R-GYI y (c) R-GY, (d) imagen original.

Este modelo se probó en la segmentación imágenes y se comparó con el modelo RGY normalizado [Montero and Sucar, 2004]. Los datos usados fueron, para piel: 2446933(pixeles) y para no piel: 53067(pixeles), obteniéndose 97.20 % de clasificación correcta. Las figuras 7.3, 7.4 y 7.5. ilustran una muestra de las pruebas de segmentación realizadas.

Para evaluar el modelo se realizaron unas segundas pruebas con diferentes datos, comparando la clasificación obtenida entre diferentes modelos de color y el modelo obtenido en nuestros experimentos, los resultados son mostrados en la tabla 7.4.

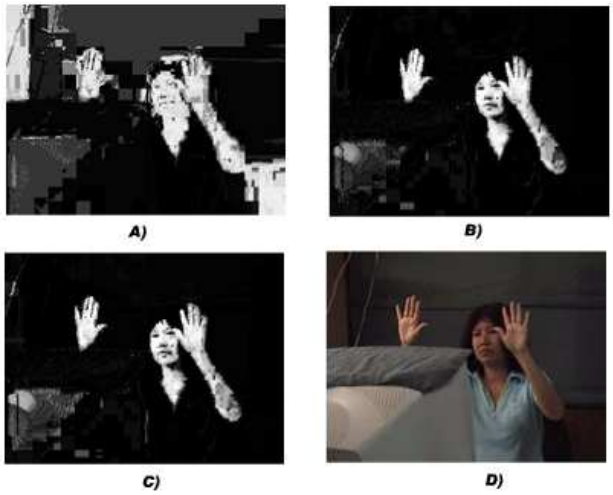


Figura 7.5: Experimento 3: Segmentación usando los modelos: (a) RGB normalizado, (b) R-GYI y (c) R-GY, (d) imagen original.

Por otro lado, comparamos los resultados de la clasificación usando el modelo obtenido contra otros clasificadores ampliamente probados pero que incrementan la complejidad del modelo, estos son el TAN y C4.5. La tabla 7.5 muestra el porcentaje de reconocimiento de cada clasificador, incluyendo el modelo inicial evaluado por el clasificador bayesiano simple (al cual tomamos como base para la mejora estructural), así como los modelos considerados en cada caso y el número de atributos que lo conforman para el proceso de aprendizaje. Es importante observar que la exactitud del nuevo modelo es superior al obtenido por C4.5 y TAN, y que éste se considera más simple por tener menos complejidad en la estructura y menor número de atributos.

Espacio de color	Porcentaje de acierto
RGB N	94
HSV	93
Y$c_r c_b$	91
R-GY	97

Tabla 7.4: Clasificador bayesiano usado para determinar la clase piel en diferentes espacios de color.

Clasificador	Número de atributos	Modelo	Exactitud %
Bayesiano Simple	9	RGBHSVYIQ	95
TAN	9	RGBHSVYIQ	96
C4.5	9	RGBHSVYIQ	97
Modelo Obtenido	3	R-GY	98

Tabla 7.5: Comparación del nuevo modelo para reconocimiento de piel con otros clasificadores.

7.1.3. Clasificación de cáncer cervical

Aplicamos este método a un problema de análisis de imágenes de colposcopia para hacer mas robusto el diagnóstico de un experto en la detección de cáncer cervical. El problema consiste en realizar un análisis del comportamiento espectral del epitelio escamoso normal del cérvix y el epitelio acetoblanco por infección de virus del papiloma, mediante el procesamiento digital de imágenes colposcópicas [Acosta-Mesa et al., 2005]. El objetivo de dicha investigación es estudiar el comportamiento temporal del epitelio escamoso normal del cérvix y el epitelio acetoblanco por infección de VPH ante la acción deshidratadora del ácido acético utilizando diferentes longitudes de onda. Se establece como hipótesis que una lesión con potencial maligno puede ser caracterizada por un modelo matemático (algoritmo computacional) que a partir de imágenes colposcópicas pueda identificar el tejido anormal en forma automática.

La figura 7.6 muestra un ejemplo de una imagén de colposcopía. Para analizar dichas imágenes se usan tres modelos matemáticos:

1. **Modelo 1- Puntos importantes**: Este modelo considera 3 puntos importantes en la curva:
   a) Ts= tiempo de reacción máximo
   b) Tb= valor máximo
   c) Tc= tiempo que tarda en retornar a su valor original
2. **Modelo 2- Polinomio de grado 5**: representa toda la serie de tiempo (cada elemento del polinomio: P1, P2, P3, P4 y P5). Un ejemplo del modelo lo podemos observar en la figura 7.7.
3. **Modelo 3- Parábola:** Se analiza la señal geométricamente mediante una parábola con parámetros: Ph,Pk y Pp. Algunos ejemplos de la representación a través del modelo de parábola lo podemos observar en la figura 7.8.

En este caso todos los atributos (tabla 7.6) son continuos y hay 3 clases:

1. típica zona sin afectación,
2. bajo grado de lesión y
3. alto grado de lesión

En un trabajo previo los 3 modelos descrito, son aplicados en forma independiente [Acosta-Mesa et al., 2005] para la predicción del cáncer caervical. En nuestros experimentos los 3 modelos son conjutados en un clasificador para formar un nuevo modelo hibrido.

Datos

Corrimos nuestros experimentos con datos obtenidos de imágenes reales de colposcopía. Para la clasificación de puntos cáncerígenos se obtuvieron los parámetros para los tres modelos [Acosta-Mesa et al., 2005] en una secuencia de imágenes de colposcopía mediante la interface (figura 7.9) usada por el experto para etiquetar los casos de entrenamiento. Estos experimentos se llevaron a cabo sobre una base de 1055 datos, 800 para entrenamiento y una BD de 255 datos para prueba (el Apéndice B describe la base de datos utilizadas para esta aplicación).

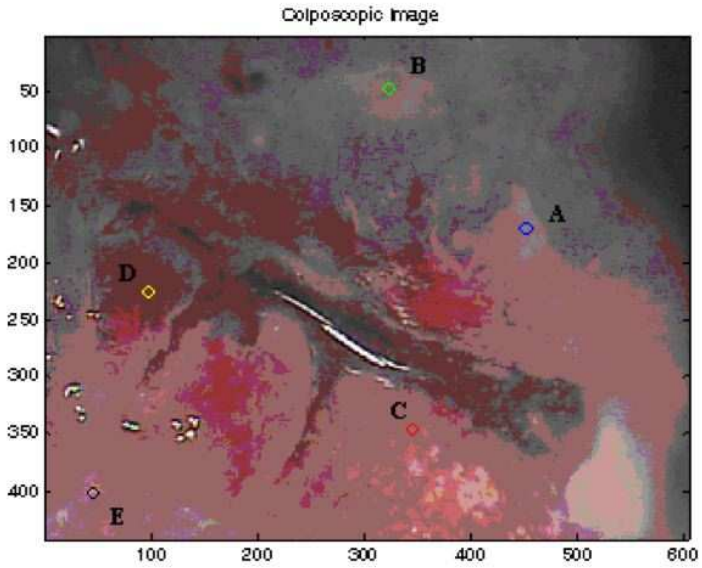


Figura 7.6: Ejemplo de una imagen de colposcopia, donde hay 5 regiones (A, B, C, D, E) que son seleccionadas a partir de una secuencia de imagenes.

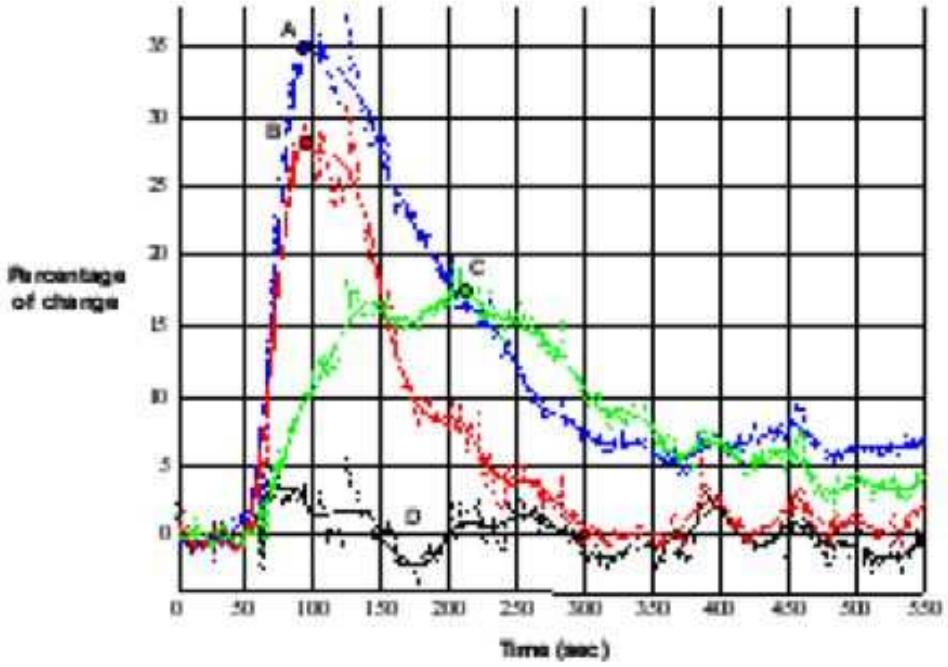


Figura 7.7: Ejemplo de la representación de las imágenes de colposcopía mediante el modelo 2.

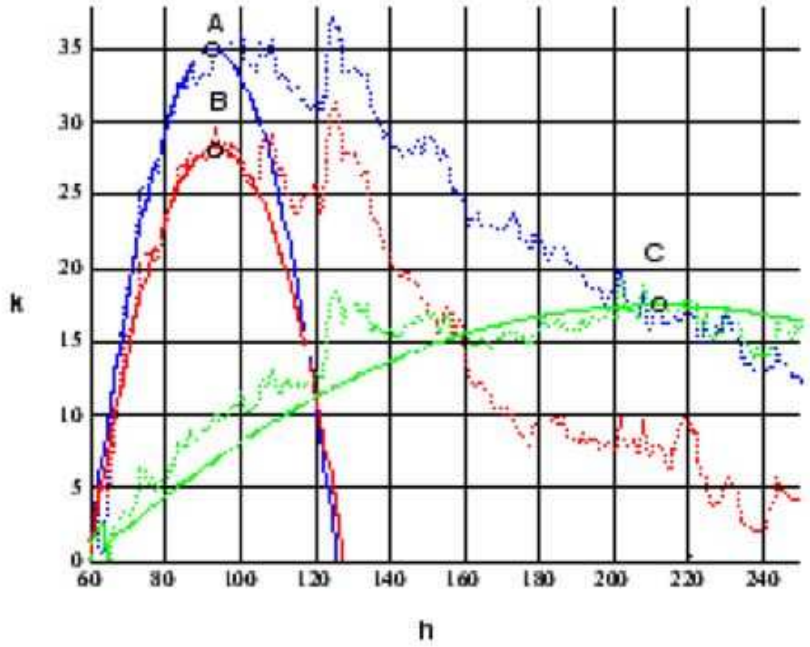


Figura 7.8: Modelo de parábola para evidencia en una imagen de colposcopía.

Núm	Atributos	Valores
1	Ts	Continuo
2	Tb	Continuo
3	Tc	Continuo
4	P1	Continuo
5	P2	Continuo
6	P3	Continuo
7	P4	Continuo
8	P5	Continuo
9	Ph	Continuo
10	Pk	Continuo
11	Pp	Continuo
12	Clase	Típico, Bajo, Alto

Tabla 7.6: Atributos involucrados en el problema de cáncer cervical.

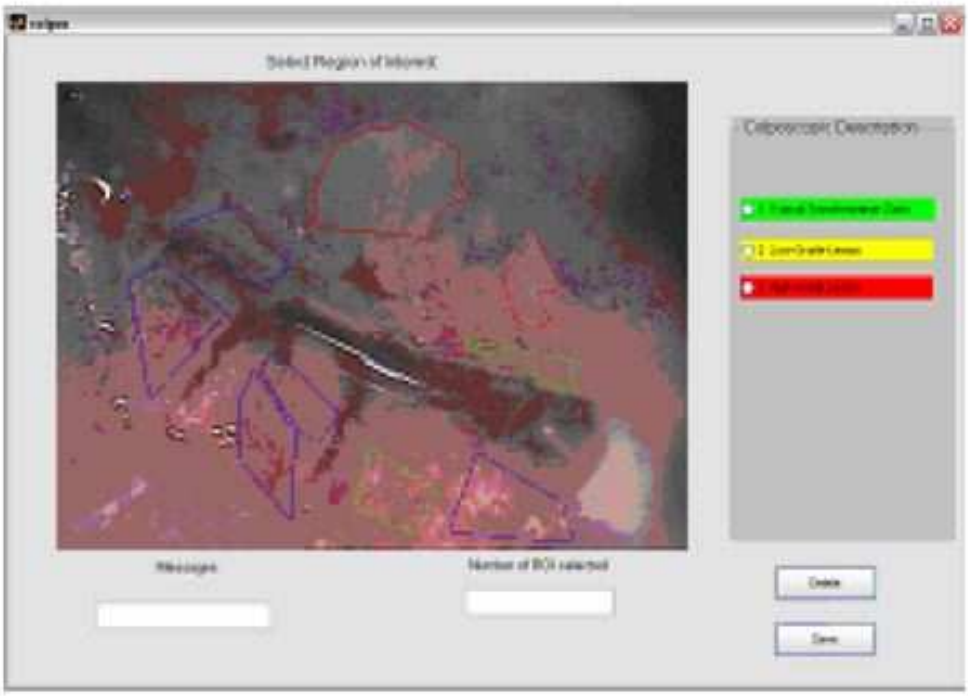


Figura 7.9: Muestra de una imagen de colposcopia y la interface usada por el experto para etiquetar los casos de entrenamiento.

Evaluación

- La discretización se evalúa comparando la clasificación del modelo original (discretización inicial) contra el modelo con la mejor discretización.
- La mejora estructural se evalúa comparando la exactitud del clasificador antes y después de cada operación.

Las variables involucradas en los tres modelos presentan diferente distribución de datos, por lo que las medidas de cercanía usadas para determinar el mejoramiento del MDL en la discretización de las variables son diferentes. A continuación se describe el proceso de cada uno de los experimentos.

Resultados

Los resultados de la clasificación del modelo con mejor discretización se muestran en la tabla 7.7, se prueba el modelo con diferentes valores de α.

Comenzamos las pruebas después de la etapa de mejora estructural. Partiendo del modelo con discretización óptima, el método aplica las etapas de eliminación y combinación de variables hasta que no se puede simplificar más y buscando la máxima exactitud en la clasificación. Consideramos que en casos donde al llevarse acabo la eliminación de un atributo y la unión del par de atributos involucrados, la clasificación arroja resultados iguales, se prefiere la eliminación ya que nos simplifica el modelo. En casos donde la(s) operaciones no afectan a la clasificación, éstas se pueden realizar

α	Atributos	Exactitud
0.9	TsTbTcP1P2P3P4P5PhPkPp	88
0.8	TsTbTcP1P2P3P4P5PhPkPp	88
0.5	TsTbTcP1P2P3P4P5PhPkPp	87
0.3	TsTbTcP1P2P3P4P5PhPkPp	85
0.1	TsTbTcP1P2P3P4P5PhPkPp	85

Tabla 7.7: Resultados de la discretización con diferentes α.

buscando hacer más simple la estructura y sólo cuando las operaciones afectan negativamente a la clasificación, estas operaciones no son consideradas.

La secuencia de operaciones, el porcentaje de clasificación y el clasificador resultante se presentan en la tabla 7.8. Podemos observar que el algoritmo inicia combinando dos atributos dependientes y después elimina otros dos atributos, hasta que llega a la estructura final con 3 atributos.

A continuación se describen estas operaciones:

1. Se unen los atributos Ts y Pk, y se obtiene un pequeño incremento en la exactitud (89 %) de la clasificación. El nuevo modelo es Ts-PkTbTcP1P2P3P4P5PhPp.
2. Se elimina el atributo P1, lo cual incrementa el porcentaje de clasificación a 89, el nuevo modelo esta compuesto por: $Ts-PkTbTcP2P3P4P5PhPp$.
3. Esta operación elimina la variable Pp, el resultado de la clasificación es: 90 %, el nuevo modelo es Ts-PkTbTcP2P3P4P5Ph.
4. En este paso, se elimina la variable Ph, obteniendose un 90 % de exactitud. El modelo generado es: Ts-PkTbTcP2P3P4P5.
5. Se elimina la variable Tc, incrementándose la exactitud del clasificador a 91 %, el nuevo modelo es: Ts-PkTbP2P3P4P5.
6. Esta operación elimina la variable P3, el porcentaje de clasificación es 92 % y el modelo resultante Ts-PkTbP2P4P5.
7. Por último se elimina la variable P5 y el modelo final es: Ts-PkTbP2P4, obteniendo un porcentaje de clasificación correcta de 94 %.

Esta prueba fue realizada con un α=0.8. El número de parámetros independientes para este modelo es de 56 (3 clases, 4 intervalos para Ts, 4 intervalos para Pk, unidas Ts-Pk 16 intervalos, 4 intervalos para Tb, 4 intervalos para P2,4 intervalos para P4). Se obtiene como modelo final la estructura conformada por 4 atributos: Ts-Pk,Tb,P2,P4. Donde un atributo es compuesto: Ts-Pk. El modelo obtenido con una exactitud del 94 %, mejora significativamente en la clasificación e incluye menos parámetros.

Para evaluar el modelo se realizarón pruebas comparando el porcentaje de clasificación correcta con otros clasificadores como TAN y C4.5, así como con el clasificador bayesiano simple. Los resultados se muestran en la tabla 7.9. Se observa que los porcentajes de clasificación de TAN y el nuevo modelo son similares, pero C4.5 tiene una ligera mejoría en la exactitud. Esto se debe en gran

sec.	operación	Núm Atrib.	Atributos	Exactitud
0		11	TsTbTcP1P2P3P4P5PhPkPp	88
1	Une Ts-Pk	10	Ts-PkTbTcP1P2P3P4P5PhPp	89
2	Elim. P1	9	Ts-PkTbTcP2P3P4P5PhPp	89
3	Elim. Pp	8	Ts-PkTbTcP2P3P4P5Ph	90
4	Elim. Ph	7	Ts-PkTbTcP2P3P4P5	90
5	Elim. Tc	7	Ts-PkTbP2P3P4P5	91
6	Elim. P3	6	Ts-PkTbP2P4P5	92
7	Elim. P5	4	Ts-PkTbP2P4	94

Tabla 7.8: Clasificación de cáncer cervico uterino.

medida a que este modelo es más complejo que un clasificador bayesiano simple. Sin embargo, la diferencia no es significativa y se podría sacrificar exactitud por simplicidad.

Clasificador	Número de atributos	Modelo	Exactitud %
Bayesiano Simple	9	TsTbTcP1P2P3P4P5PhPkPp	88
TAN	9	TsTbTcP1P2P3P4P5PhPkPp	94
C4.5	9	TsTbTcP1P2P3P4P5PhPkPp	96
Modelo Obtenido	3	Ts-PkTbP2P4	94

Tabla 7.9: Comparación del nuevo modelo para la apliación de Cáncer cervical con otros clasificadores.

7.1.4. Análisis del método estático

Las pruebas se realizaron en dos contextos reales: reconocimiento piel y clasificación de cáncer cervical. Cada aplicación evaluó por separado la parte de construcción del mejor modelo y la clasifiación del mismo. El resultado de esta clasificación, fue comparado con los obtenidos con otros clasificadores: TAN y C4.5, así como también se comparó con los resultados obtenidos a través del modelo original con clasificador bayesiano simple, obteniéndose buenos resultados.

En el reconocimiento de piel, el modelo obtenido fue el R-GY (considerando sólo 3 atributos, dos de ellos unidos en uno sólo), lo cual es un modelo más compacto que el original que considera 9 atributos de los 3 modelos de color (RGB, HSV, YIQ) además de que obtiene un porcentaje correcto de clasificación más alto, 98 % contra 95 %, que fue el obtenido por el modelo original. Así mismo se comparó con otros modelos de color (RGB normalizado, HSV y Yc_rc_b) que poseen igual número de atributos y se obtuvo un mejor porcentaje. Tambien se realizaón comparaciones del porcentaje de clasificación correcta obtenido con el nuevo modelo (R-GY) con otros clasificadores, donde el TAN obtuvo 96 % y C4.5 obtuvo 97 %, mientras que la precisión obtenida por nuestro modelo es superior 98 %, además de considerar las otras estructuras son más complejas que nuestro modelo. Por último se compara el porcentaje de clasificación que obtuvo el clasificador bayesiano simple inicial (95 %) con el nuevo modelo, y observamos que la clasificación mejoró con la aplicación del método ACBE.

Para el caso de la clasificación de cáncer cervical, en la comparación del porcentaje de clasificación correcta obtenido por el modelo inicial (considerando 11 atributos) fue de 88 % misma que mejoró al aplicar nuestro método y obtener el nuevo modelo Ts-PkTbP2P4 (con 5 atributos, dos de ellos unidos en uno sólo) que obtuvo 94 %. En las pruebas realizadas comparando este nuevo modelo

a través del clasificador bayesiano simple con otros clasificadores, se obtuvieron; para el TAN 94 %, para el C4.5 96 % y para nuestro modelo 94 %, donde podemos observar que sólo el C4.5 obtuvo una ligera mejora en la exactitud. Si tomamos en cuenta que estas estructuras son más complejas que un clasificador bayesiano simple, consideramos que los resutados obtenidos con nuestro modelo son aceptables, ya que se redujo la complejidad el modelo y la clasificación es buena.

Con respecto a la discretización se observó, para ambas aplicaciones, que la clasificación mejora si damos un mayor peso al parámetro que considera la exactitud que a la complejidad, esto lo pudimos constatar con las pruebas realizadas variando los valores de alfa. En general, podemos concluir que nuestra metodología genera clasificadores competitivos con métodos como TAN y C4.5, pero que son en general más simples y eficientes.

La complejidad temporal de este algoritmo es bastante aceptable. En el peor de los casos su complejidad es O*(NA2P*) (ver **Apéndice A**), que es la correspondiente a la fase de mejora estructural, la cual involucra realizar las operaciones de unión y eliminación de atributos (*A*) considerando *N* estancias y *P* particiones , una vez que se ha discretizado cada atributo. Lo que supone que la complejidad computacional es cudrática en función del número de atributos y lineal respecto al número de datos, lo que en términos generales es aceptable.

Los tiempos de aprendizaje del método ACBE para las aplicaciones de reconocimiento de piel y detección de cáncer cervical, son mostrados en la tabla siguiente:

Aplicación	Número de instancias	Número de atributos	Clases	Tiempo de Proc. (h.m.s)
Detección de piel	6,300,000	9	2	0.59.06
Detección de cáncer	1055	11	3	0.02.45

Tabla 7.10: Tiempo de aprendizaje para el modelo estático.

Los tiempos de aprendizaje del método ACBE para las aplicaciones de reconocimiento de piel y detección de cáncer cervical, son mostrados en la tabla 7.10.

Por otro lado, para diagnósticos médicos es importante garantizar no solo un alto porcentaje de clasificación, hay que tomar también los costos de falsos positivos y falsos negativos que pueden no ser iguales. En la aplicación de cáncer cervical sería interesante buscar un compromiso entre la fracción de verdaderos positivos (probabilidad de clasificar correctamente si tiene cáncer) y la fracción de verdaderos negativos (probabilidad de clasificar correctamente si no tiene cáncer). Para ello se tendría que valorar el costo que supone un error de falso positivo (una paciente sana que ha sido diagnosticada con cáncer) y un error de falso negativo (paciente enferma diagnosticada como sana).

En otras aplicaciones este costo podría medirse en términos económicos o en cualquier otra unidad de medida. Sin embargo, en esta aplicación (y en forma general en el campo de la medicina), el costo que supone un error de falso positivo frente a un falso negativo difícilmente se puede medir.

- Un falso positivo significaría mayor costo económico ya que implicaría la realización de otras pruebas, mayores molestias y angustia.
- Un falso negativo ocasionará un retraso en la detección de la enfermedad o la muerte si no se detecta a tiempo, ya que dificultaría el tratamiento.

Prácticamente mientras que el costo de un falso positivo representa un mayor costo económico y en mayores molestias para el paciente, el costo de un falso negativo puede ser, en un caso extremo, la propia vida del paciente.

7.2. Clasificadores bayesianos dinámicos

7.2.1. Metodología

Para realizar las pruebas del método de aprendizaje de clasificadores bayesianos dinámicos (ACBD) se requiere de dos partes:

1. Creación del clasificador inicial
2. Clasificación dinámica.

El clasificador inicial se crea mediante los métodos de discretización, mejora estructural y determinación del nodo clase oculto, obteniendo así, el *mejor modelo.*

Para el segundo punto, el proceso de clasificación dinámica se lleva a cabo a través de los HMM, conformado por los atributos del *mejor modelo* obtenido en la primera etapa. Se *entrena* con 50 ejecuciones de cada gesto a través del algoritmo *Baum-Welch* para obtener los parámetros del modelo. Para la *evaluación* o reconocimiento se usa el algoritmo de *Viterbi*. Finalmente se obtiene el porcentaje de reconocimiento para el gesto, es decir, el número de veces que se reconoció el gesto entre el número total de videos (secuencias).

7.2.2. Reconocimiento de gestos

El método ha sido aplicado a un problema real, como el Reconocimiento de Gestos [Montero and Sucar, 2004], donde se modela el movimiento de la mano del usuario por medio de 5 características del movimiento. Los gestos reconocidos fueron 7. El proceso de aprendizaje se llevó a cabo aplicado para cada gesto, construyendo un modelo para cada uno de estos.

Los gestos considerados se obtienen de una camara observando gestos *manipulativos* realizados por una persona [Montero and Sucar, 2004], algunos de estos gestos se ilustran en la figura 7.10. Son los siguientes:

1. Abrir
2. Borrar
3. Escribir
4. Hojear
5. *Mouse*
6. Impresora
7. Teléfono

Para realizar el reconocimiento de actividades humanas se consideran diversos aspectos [Davis, 1998], como el que una actividad se ejecuta en un lapso variante de tiempo y que pueden existir oclusiones de diferentes partes del cuerpo durante la ejecución de la actividad, por lo que para modelar este tipo de actividades es conveniente utilizar un modelo dinámico.

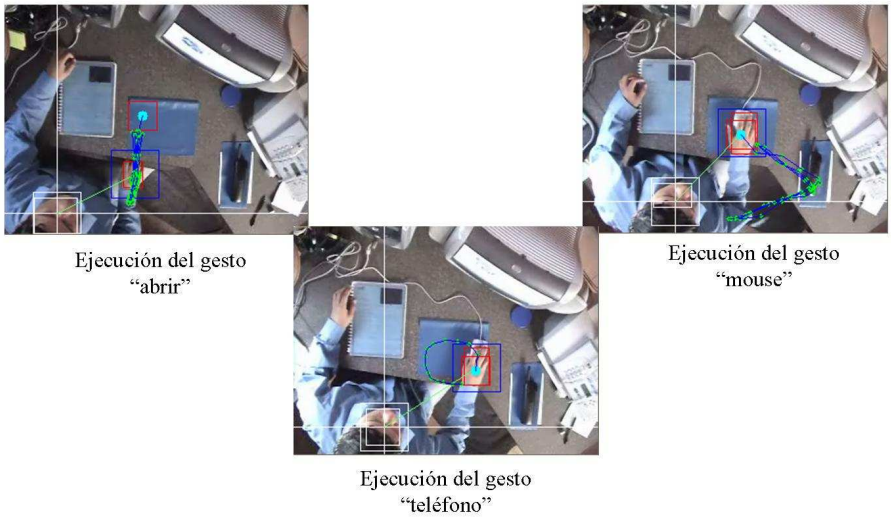


Figura 7.10: Ejecución de gestos.

Datos

Para llevar a cabo la tarea del reconocimiento de gestos, se tiene una secuencia de imágenes procesadas de video de la ejecución de cada gesto (figura 7.10) y un modelo inicial de la estructura, que incluye un nodo de estado en cada tiempo el cual no es observable (oculto). El nodo de estado representa la evolución temporal del gesto o demán reconocer y es oculto debido a que se desconocen los posibles estados para ese nodo. Para analizar las imágenes se usa un modelo con 5 características que representan el movimiento de una mano, sobre el eje *x* y sobre el eje *y* de la imagen, el cambio de *magnitud, dirección* y *velocidad* del movimiento. Los nombres de las variables del modelo son:

- Variable 0: Magnitud
- Variable 1: Dirección
- Variable 2: Velocidad
- Variable 3: Coordenada X
- Variable 4: Coordenada Y

En este caso todos los atributos son continuos y los valores del nodo clase se desconocen:**Nodo clase oculto**.

El entrenamiento del clasificador dinámico con los siguientes datos por cada gesto:

- 50 videos para entrenamiento y
- 50 videos para pruebas.

7.2.3. Evaluación

En esta sección se realizan las pruebas para evaluar los diferentes procesos del clasificador bayesiano dinámico. Cada gesto se valua en forma separada, partiendo del modelo con mejor discretización, y aplicandose posteriormente los procesos de: **determinación del mejor número de estados, mejora estructural y clasificación dinámica**.

En el proceso de **determinación del mejor número de estados** del clasificador bayesiano simple, se involucra un ciclo que genera diversas estructuras donde se varía el número de estados. Dichas estructuras se evaluan a traves de la medida de *calidad* (con base al MDL), inicialmente se consideran *2* estados y se varía el número hasta *N* (determinado por el experto). La estructura que obtiene mejor *calidad* es considerada como la mejor estructura con el mejor número de estados. En las pruebas se considera evaluar la *calidad* de las estructura considerando un compromiso entre exactitud y longitud, por lo cual cosideramos darle la misma importancia a los dos parametros, considerando α= 0.5.

El proceso de **mejora estructural** considera evaluar la estructura antes y después de cada operación, a traves de la medida de *calidad*, al eliminar o combinar un atributo se busca incrementar la calidad pero también puede resultar en un decremento, lo que puede o no afectar a la clasificación, lo cual no podemos saber hasta que se realice la clasificación dinámica, que realizamos como último paso, por lo que en forma momentánea se considera un umbral de decremento máximo permitido; en estas pruebas tomamos un decremento del 5 por ciento.

Por último **la clasificación dinámica** se prueba comparando el clasificador inicial con el mejor clasificador. Para ejemplificar el proceso se muestra el aprendizaje de 3 gestos: abrir, teléfono y *mouse*.

Gesto Abrir

En este caso, el clasificador que obtiene la mejor *calidad* es la estructura constituida por un nodo clase con 7 estados. En la mejora estructural se consideran pares de variables dependientes entre si. En la tabla 7.10 se muestran las operaciones realizadas, donde se observa que las mejores calidades (considerándose un umbral de decremento máximo) se obtiene al eliminar atributos. Inicialmente se elimina el atributo velocidad, donde se observa que la calidad no se incrementa pero sin embargo no baja en un porcentaje considerable (mayor del 5 por ciento), posteriormente se elimina el atributo coordenada Y y al final el atributo Coordenada X, observandose un decremento menor del 5 por ciento en ambos casos. Por lo que el modelo final se contituye de dos variables: ***Magnitud*** y ***Dirección***.

sec.	op.	N. Atrib.	Atributos	Calidad
0		5	Magnit,Direc,Vel,CoordX,CoordY	16.6803
1	Elim. Vel.	4	Magnit,Direc,CoordX,CoordY	27.6568
2	Elim. CoordY	3	Magnit,Direc,CoordX	26.5021
3	Elim. CoordX	2	Magnit,Direc,	25.6844

Tabla 7.11: Reconocimiento del gesto *abrir*.

Gesto *mouse*

En este gesto se observa que la mejor *Calidad* es obtenida por la estructura cuyo nodo clase tiene 6 estados. Con respecto a la mejora estructural, las operaciones se muestran en la tabla 7.11. Se realiza la eliminación de 3 atributos y se observa que no hay incremento en la calidad pero el decremento es menor del 5 porciento. El modelo final esta formado por las variables: ***Dirección y Velocidad***

sec.	op.	N. Atrib.	Atributos	Calidad
0		5	Magnit,Direc,Vel,CoordX,CoordY	
1	Elim. CoordY	4	Magnit,Direc,Vel,CoordX	27.6568
2	Elim. CoordX	3	Magnit,Direc,Vel	26.5021
3	Elim. Magnit	2	Direc,Vel	25.6844

Tabla 7.12: Reconocimiento del gesto *mouse*.

Gesto Teléfono

El mejor número de estados para el gesto *Teléfono* (7 estados) coincide con el del gesto *Abrir*. La tabla 7.12 muestra las operaciones realizadas para la mejora estructural, donde se eliminan 3 atributos. Se observa que el modelo generado: ***Magnitud*** y ***Dirección***, que también coincide con el modelo del gesto *Abrir*, aunque la secuencia de operaciones no es la misma.

sec.	op.	N. Atrib.	Atributos	Calidad
0		5	Magnit,Direc,Vel,CoordX,CoordY	
1	Elim. Vel	4	Magnit,Direc,CoordX,CoordY	27.6568
2	Elim. CoordX	3	Magnit,Direc,CoordY	26.5021
3	Elim. CoordY	2	Magnit,Direc,	25.6844

Tabla 7.13: Reconocimiento del gesto *teléfono*.

Clasificación dinámica

Una vez que se tiene el mejor modelo, con la mejor discretización, el mejor número de estados y la mejora estructural, se prueba el clasificador en forma dinámica. Este proceso se realiza a través de los HMM. En la figura 7.11 se muestra el clasificador bayesiano dinámico para el gesto abrir, donde el modelo tiene 2 atributos y el nodo estado tiene 6 valores. La tabla 7.13 muestra los porcentajes de clasificación obtenidos por el clasificador bayesiano dinámico inicial y los porcentajes de clasificación obtenidos para el mejor modelo para cada gesto.

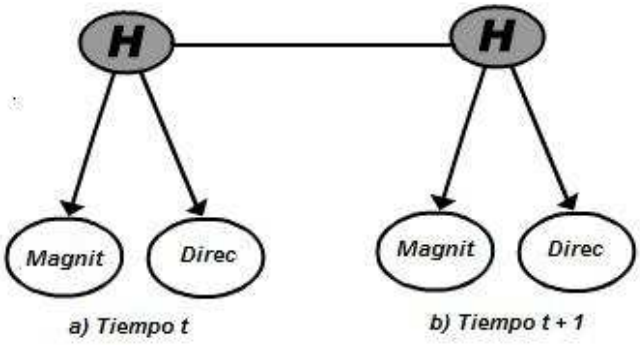


Figura 7.11: Clasificador dinámico para el gesto *Abrir*.

Gesto	Modelo inicial	% de clasificación clasicador inicial
Abrir	Magnitud, Dirección, Velocidad, CoordX, CoordY	100 %
Borrar	Magnitud, Dirección, Velocidad, CoordX, CoordY	80.39 %
Escribir	Magnitud, Dirección, Velocidad, CoordX, CoordY	100 %
Hojear	Magnitud, Dirección, Velocidad, CoordX, CoordY	100 %
Impresora	Magnitud, Dirección, Velocidad, CoordX, CoordY	96 %
mouse	Magnitud, Dirección, Velocidad, CoordX, CoordY	78 %
Teléfono	Magnitud, Dirección, Velocidad, CoordX, CoordY	100 %
Promedio		93.48 %
Gesto	Mejor modelo	% de clasificación mejor modelo
Abrir	Magnitud y Dirección	100 %
Borrar	Dirección y Velocidad	93.88 %
Escribir	Dirección y Velocidad	97.88 %
Hojear	Magnitud y Dirección	98 %
Impresora	Magnitud y Dirección	100 %
mouse	Dirección y Velocidad	98 %
Teléfono	Magnitud y Dirección	100 %
Promedio		98.25 %

Tabla 7.14: Porcentajes de reconocimiento para todos los gestos.

7.2.4. Análisis método dinámico

Durante las pruebas cada gesto (7 gestos) fue probado en forma separada. Los modelos iniciales consideran los 5 atributos, mientras los modelos obtenidos fueron más compactos ya que consideran solo 2 atributos. Las variables más importantes fueron: magnitud, velociad y dirección, de las cuales 2 aparecen en los diferentes modelos. Los gestos *Abrir, Hojear, Impresora y Telefono* consideraron sólo *Magnitud y Dirección*, mientras que los gestos restantes (*Borrar, Escribir y mouse*), consideraron *Dirección y Velocidad*. Los porcentajes de reconocimiento fueron en promedio 98.25 % para los nuevos modelos contra 93.48 % que fue lo obtenido por los modelos originales.

El metodo propuesto nos permite determinar en forma automática la estructura, discretización numero de estados para modelos dinámicos como el clasificador bayesiano simple dinámico y el HMM. Los modelos resultantes son a la vez mas simples y con mejor porcentaje de reconocimiento.

La complejidad en el aprendizaje del clasificador bayesiano dinámico implica dos partes: la creación de la estructura inicial (que corresponde al método implementado) y la creación de la red de transición (donde se utilizó un HMM). Por tanto, para la primera parte, la compejidad es similar a la del método estático (véase apéndice A) la cuál es O(N$A-2$P). Para la creación de la red de transición la complejidad computacional corresponde al tiempo de aprendizaje del algoritmo Baum−Welch *O(QXT)* con una topología de izquierda-derecha, donde Q es el número de estados, X el espacio de observaciones y T número de vectores de la secuencia. Lo cual supone un costo computacional también aceptable. El único parámetro de ACBE que afecta a la complejidad y que es ajeno a los datos de entrenamiento es *r*, el número de estados que, en el peor de los casos alcanzará el valor máximo de 10.

Los tiempos de aprendizaje del método ACBE para las aplicaciones de reconocimiento de piel y detección de cáncer cervical, son mostrados en la tabla 7.15.

Aplicación	Número de instancias	Número de atributos	Tiempo de Proc. Prom. (h.m.s)
Reconocimiento de gestos	5,051,200	5	1.26.05

Tabla 7.15: Tiempo de aprendizaje para el modelo dinámico.

7.3. Implementación

Las pruebas realizadas al método estatico y al metodo dinamico fueron implementadas en una computadora Pentium IV, de 3.1 GHz, con una capacidad de memoria de 1GB, sobre una plataforma Windows XP. Los algoritmos fueron realizados en Java usando las clases principales de la herramienta ELVIRA [Lacave and Díez, 2005] e implementando otras. Las pruebas para comparación del clasificador estático con otros clasificadores, se realizaron en WEKA [Witten and Frank, 2005].

7.4. Limitaciones

Ambos métodos se probaron con datos reales, considerando aplicaciones con un máximo de 11 atributos, y hasta con 6,300,000 registros. Sin embargo para aplicaciones con un mayor número de atributos y con pocos datos la exactitud del método se ve afectada.

La complejidad en la implementación (espacio/memoria) del algoritmo crece en forma lineal a medida que se consideran grandes bases con muchas instancias. La etapa de aprendizaje estructural que es la de mayor costo computacional, este se incrementa en forma cuadrática de acuerdo al orden de las posibles combinaciones a medida que se considera un mayor número de atributos, lo cual implica mayor tiempo de aprendizaje, sobre todo cuando se trata de atributos continuos que poseen una distribución dispersa en los datos ya que ocasiona el crecimiento de las tablas de probabilidad (mayor número de posibles valores o particiones para cada atributos). Otra consideración es que el método de discretización también es sensible a la semilla inicial.

El aprendizaje estructural para el método dinámico considera la medida MDL para evaluar el clasificador, sin embargo podría considerarse realizar la clasificación dinámica después de cada operación, lo cual implicaría mayor tiempo en el proceso de aprendizaje, como se puede observar en la tabla de tiempos de aprendizaje.

Capítulo 8

Conclusiones y Trabajo futuro

La motivación principal para desarrollar este trabajo la constituye considerar que aunque el clasificador bayesiano simple ha sido ampliamente utilizado debido a que es un modelo de clasificación eficiente, fácil de aprender y con gran exactitud en muchos dominios, este presenta dos principales desventajas: la exactitud de la clasificación disminuye cuando los atributos no son independientes, y no puede ocuparse de atributos continuos. Además de que existen otras consideraciones que afectan el proceso de aprendizaje, tales como trabajar con información incompleta o faltante (en el mundo real los fenómenos raramente son observados completamente), el manejo de grandes cantidades de datos y/o variables que puede reducirse mediante la selección de atributos (representativos o la eliminación de atributos que no son relevantes al problema o representar ambientes dinámicos que modelan la evolución de las variables a través del tiempo (considerando que la mayoría de los fenómenos están en constante evolución).

Se ha presentado en este trabajo una metodología para el aprendizaje de clasificadores bayesianos. El objetivo ha sido proporcionar un método integral de aprendizaje que mejora la exactitud del clasificador manteniendo una estructura simple, que considera aspectos de discretización de atributos continuos no paramétrizados, determinación de atributos dependientes, selección de atributos relevantes y manejo de información oculta. Concretamente se presentaron dos métodos de aprendizaje de clasificadores bayesianos simples, un método que genera clasificadores estáticos y otro que genera clasificadores dinámicos. Estos métodos combinan aspectos de aprendizaje y mejora en la estructura con el objetivo de lograr una mayor exactitud en la clasificación. El primer método cubre aspectos de discretización y mejora estructural, el segundo, demás de los aspectos mencionados, la determinación del nodo clase oculto y la clasificación dinámica.

8.1. Resumen

Para el aprendizaje automático los clasificadores Bayesianos ofrecen algunas ventajas, debido a que son fáciles de construir y de entender. Sin embargo presentan dos problemas principales: consideran que todas las variables involucradas son independientes dada la clase y no manejan atributos continuos, lo cual afecta en la exactitud del clasificador. En el clasificador bayesiano los atributos continuos son tratados comúnmente asumiendo una distribución normal de los datos, sin embargo esto no siempre es así, la realidad es que la distribución de los datos reales generalmente se desconoce.

La discretización es una alternativa para procesar datos continuos, convirtiéndolos a discretos. Un método de discretización prueba su efectividad en el clasificador bayesiano reduciendo el error de clasificación. Existen diferentes factores que afectan el proceso de discretización, entre estos encontramos la determinación de los puntos de corte (limites), el error de tolerancia en la estimación de la probabilidad, la relación entre atributos, el número de atributos que conforman el problema, el número de instancias que conforman el conjunto de entrenamiento y el número de clases a clasificar. En la literatura podemos encontrar diferentes métodos de discretización, en su mayoría, aplicados en forma separada del aprendizaje. En este trabajo presentamos un nuevo método de discretización basado en el principio MDL, el cual considera discretizar los atributos continuos en forma individual al momento de aprender la estructura con la finalidad de mejorar la exactitud del clasificador, el cual se describe en el capitulo 5.

El incremento de procesos automatizados resulta en la generación de grandes volúmenes de información, mismos que al momento de procesar no pueden aplicárseles algoritmos tradicionales por diversas inconveniencias (implican muchos ciclos, tiempo de procesamiento, mayor costo, etc), la clasificación o selección de datos permite enfocar la búsqueda en subconjuntos de variables y/o muestras de datos en donde realizar el proceso reduciendo el problema, este proceso se puede llevar a cabo eliminando variables que no aportan mucha información al problema y/o seleccionado aquellas que estén mas estrechamente relacionadas. La clasificación no es un proceso trivial ya que debe identificar un conjunto de categorías o clases que describen realmente al conjunto de datos. El clasificador bayesiano simple considera que todas las variables son independientes aunque esto no sea así en muchos dominios y aunque existen otros tipos de clasificadores Bayesianos que consideran relaciones entre las variables (como el TAN, BAN o la GBN) estos poseen involucran mayor complejidad ya que involucran conexiones entre las variables.

Por otro lado, diferentes dominios que involucran tiempo, como en el reconocimiento de actividades humanas (gestos), procesamiento de voz, diagnostico medico, bioinformática y otras aplicaciones han sido modeladas con el clasificador bayesiano simple. Los clasificadores bayesianos dinámicos pueden llevar a cabo la clasificación por medio de los algoritmos habituales de inferencia que se aplican a clasificadores bayesianos estáticos. Sin embargo, cuando se trata de observaciones parcialmente observadas, principalmente cuando no se conoce el nodo a predecir (nodo clase) nos encontramos con que en muchos casos la determinación de los mejores estados se encuentran dados por el experto con base a su experiencia. Los clasificadores bayesianos dinámicos son una generalización de los modelos ocultos de Markov (HMM), donde se desconoce el nodo de observación. El nodo de estado representa la evolución temporal y es oculto debido a que se desconocen los posibles estados para ese nodo. Al igual que cuando los datos son completos, cuando se trata de atributos oculto, los parámetros asociados al nodo deben ser estimados. Un método para esto es el algoritmo EM [Dempster et al.,1997; Lauritzen, 1995].

En esta tesis proponemos dos métodos de aprendizaje de clasificadores bayesianos simples: uno aplicado a dominios estáticos y otro aplicado a dominios dinámicos. El aprendizaje de clasificadores bayesianos estáticos (ACBE) está compuesto por 4 etapas: inicialización, discretización, mejora estructural y clasificación, etapas que se repiten alternativamente hasta que la exactitud de la clasificación no puede ser mejorada. Las principales aportaciones de esta metodología es la integración de los procesos de mejora estructura y discretización. así como la variante del principio MDL para discretización.

La fase de la discretización se basa en el principio de descripción de longitud mínima (MDL), este principio hace un compromiso entre la exactitud y la complejidad de la estructura.El proceso de discretización busca el número de intervalos que reduce al mínimo el MDL a través de una medida

de calidad de la estructura. Cada atributo es procesado en forma independiente, y a partir de una discretización inicial uniforme, se generan particiones adicionales en cada iteración dividiendo el intervalo en dos nuevos intervalos, posteriormente se calcula la medida de calidad. de la rama y el proceso se repite en forma iterativa hasta que este no puede ser mejorado o no mejora en un porcentaje (medida de cercanía). La medida de calidad es similar a la utilizada en [Martinez and Sucar, 1998], la cual estima la exactitud midiendo la información mutua entre los atributos y la clase; y la complejidad contando el número de parámetros por atributo. Esta fórmula involucra una constante alfa, con valores en [0,1], que nos sirve para balancear el peso de cada aspecto, exactitud contra complejidad. Un valor alfa=0.5 da la misma importancia a complejidad y exactitud, mientras que alfa cercano a 0 considera darle mayor importancia a la complejidad y alfa cercano a uno mayor importancia a la exactitud. Esta fórmula también involucra un parámetro de longitud máxima que considera el número máximo de intervalos por atributo (dependiendo de la distribución de los datos) y un peso máximo, que corresponde a la sumatoria de información mutua de la estructura completa, que se calcula a partir de la longitud máxima. Esto es con base a un número máximo de posibles intervalos en los que se puede discretizar la variable.

La fase de la mejora estructural elimina y/o une pares de atributos, este proceso se basa en el resultado de la clasificación después de haber aplicado una operación comparando con el modelo anterior. La etapa de la mejora estructural utiliza reglas de decisión simples, considerando solo ciertos pares de atributos de acuerdo a la medida de información mutua condicional. Cuando existen aplicaciones que involucran una gran cantidad de atributos, y el número de pares de variables a considerar es muy grande, se considera utilizar un umbral (definido por el usuario) para evaluar únicamente un N relaciones. Sin embargo, aunque el proceso de búsqueda esta guiado considerando aquellas ramas que son más fuertes (variables altamente relacionadas), el proceso de ejecutar las operaciones depende en gran medida del orden en que se toman las variables. Este proceso se repite hasta que no haya mas atributos superfluos o dependientes, mientras se mejore el resultado de la exactitud del clasificador.

El aprendizaje de clasificadores bayesianos dinámico (ACBD) también es un proceso iterativo que busca mejorar la estructura, e incluye 5 fases, inicialización, discretización, determinación del nodo clase oculto, mejora estructural y la clasificación dinámica. El criterio de paro usado en búsqueda de la mejor estructura esta dado por una medida de calidad basado en el MDL. Las principales aportaciones de este trabajo son: un método integral para aprender clasificadores bayesianos dinámicos y la metodología para determinar el número de estados para el nodo clase oculto.

Este método incluye 2 fases del método, las cuales son: la etapa de discretización (similar en ambos métodos) y la etapa de mejora estructural (la cual se diferencia en que las estructuras se evalúan no a través de la exactitud de la clasificación sino a través de la medida de calidad basada en el principio MDL. La etapa para determinar el mejor número de estados para el nodo oculto también involucra esta medida. Una vez que se han realizado las primeras 4 etapas se considera que tenemos la estructura del clasificador inicial Bo y la red de transición B? es construida usando algún método general de RBD, en este caso usando un HMM. Finalmente se evalúa el clasificador bayesiano dinámico en el dominio de interés.

8.2. Aportaciones

Las principales aportaciones de este trabajo de tesis, son las siguientes:

1. Se propone un método integral para aprendizaje de clasificadores bayesianos estáticos, que aborda los dos principales desventajas del clasificador bayesiano simple, el manejo de datos continuos y la determinación de atributos que no son independientes. El método considera aspectos de discretización y mejora estructural (mediante dos operaciones básicas; la unión de atributos dependientes y la eliminación de atributos irrelevantes), cuidando de conservar la simplicidad de la estructura y buscando mejorar la exactitud del clasificador.

2. Un método integral para aprendizaje de clasificadores bayesianos dinámicos, que comprende además de las técnicas de discretización y mejora estructural, el manejo de información incompleta (nodo clase oculto) y la representación de ambientes dinámicos, buscando al mismo tiempo, la mejora en la exactitud del clasificador y el conservar la simplicidad de la estructura. En si, la propia integración de estos diferentes métodos para el aprendizaje, ya que generalmente se han aplicado en forma separada y por otro lado referente al aprendizaje de clasificadores bayesianos dinámicos existen pocos trabajos relacionados.

3. Una nueva técnica de discretización supervisada basada en el principio MDL, aplicada durante el proceso de aprendizaje, en busca de una estructura que mejore la exactitud del clasificador, la cual es aplicada a cada atributo

4. El manejo de información incompleta (considerando que la mayoría de los fenómenos son raramente observados en su totalidad) a través de una técnica para la determinación de un número de estados para el nodo clase oculto que incremente la exactitud del clasificador (apropiado para aplicaciones donde no se sabe con exactitud lo que se va a clasificar) en clasificadores bayesianos dinámicos.

5. A través de los experimentos se en la aplicación de detección de piel y no piel se obtuvo un nuevo modelo de color RGY, el cual está siendo aplicado y obtiene buenos resultados. Para la aplicación de cáncer cervical la obtención de un modelo híbrido que permite la clasificación de 3 clases: típica zona sin afectación, bajo grado de lesión cancerígeno y alto grado de lesión cancerígeno.

   La investigación llevada a cabo durante la realización de esta tesis, ha producido las publicaciones que se citan en el Apéndice B.

8.3. Limitaciones

La complejidad temporal de los algoritmos crece en forma lineal a medida que se consideran grandes bases de datos y cuando se trata de un gran número de atributos crece en forma cuadrática, sobre todo cuando se trata de atributos continuos que poseen una distribución dispersa en los datos ya que ocasiona el crecimiento de las tablas de probabilidad (mayor número de posibles valores o particiones para cada atributos). Por otro lado tenemos que aunque ambos métodos se probaron con datos reales, solo se consideraron aplicaciones con un máximo de 11 atributos, y hasta con poco mas de 6,000,000 de registros. Sin embargo para aplicaciones con un mayor número de atributos y con pocos datos la exactitud del método puede verse afectada.

8.4. Trabajo futuro

Aún cuando la metodología propuesta ha obtenido buenos resultados en las pruebas experimentales, estos métodos se podrían mejorar en diversos aspectos, tal como:

- Consider cambiar el orden de ejecución de las etapas. Aún cuando las etapas para el proceso de aprendizaje de los clasificadores bayesianos simples, se han ejecutado en un orden especificado, se podría invertir el orden de las etapas de discretización y mejora estructural, para observar como influye en la clasificación. Se podría realizar la mejora estructural con una discretización inicial y posterior a esta, aplicar el método de obtención de la mejor discretización solo con los atributos seleccionados. Lo cual sería conveniente en casos de aplicaciones con gran número de atributos y datos.
- Durante el proceso de aprendizaje de los clasificadores bayesianos simples dinámicos, se podría considerar como un proceso iterativo la etapa de determinación del mejor número de estados y la mejora estructural, es decir con base a la mejor estructura, variar el número de estados para evaluarlos dinámicamente.
- El método de discretización podría ser mejorado particionando no solo en 2 nuevos intervalos, sino considerando particionar aquellos intervalos con mayor frecuencia, así como también se podría experimentar con diferentes particiones iniciales.
- Por otro lado, el método de clasificación dinámica, aprende únicamente la estructura, se podrían aprender las redes de transición de clasificadores bayesianos dinámicos y extender a otro tipo de estructuras, no solo a clasificadores bayesianos simples.
- Finalmente todo el proceso de aprendizaje podría considerarse iterativo, es decir, podríamos considerar una discretización inicial (2 particiones) y dejar el método correr en su totalidad, posteriormente probar con otra partición inicial (p.e.. 3) y ejecutar todo el proceso, para comparar los clasificadores obtenidos.

Con respecto a los experimentos, en la aplicación de cáncer cervical podría introducirse una umbral o medida de utilidad para buscar un compromiso entre la fracción de verdaderos positivos (probabilidad de clasificar correctamente si tiene cáncer) y la fracción de verdaderos negativos (probabilidad de clasificar correctamente si no tiene cáncer). Para ello podríamos tener un costo para indicar el beneficio de un verdadero positivo (que podría ser 1) y un verdadero negativo y costos de un falso negativo y un falso positivo. El costo en cada uno de estos casos podría depender de las consecuencias de la decisión. Donde podríamos calcular la utilidad esperada de la prueba como el promedio ponderado de los diferentes costos y beneficios. El umbral permitido sería aquél que maximizara esta función de utilidad. Aunque este umbral puede ser totalmente subjetivo.

Para probar los métodos se podrían buscar otras aplicaciones, que consideraran un mayor número de atributos o áreas donde existan pocos datos (casos) para observar el comportamiento de estos.

Apéndice A

Complejidad Temporal

En este apéndice estudiaremos la complejidad temporal de los métodos estático y dinámico, y veremos que se trata de un algoritmo eficiente incluso en el peor caso.

Para la parte estática, el algoritmo ACBD aplica secuencialmente cada una de las etapas siguientes:

- Inicialización: el tiempo de complejidad para el aprendizaje del clasificador bayesiano simple crece en forma lineal con respecto a los datos, esto es *O(NA*), donde N es el número de instancias (casos) en la BD y A es el número de atributos.
- Discretización: aquí se considera, además del número de instancias y el número de atributos, el número de particiones en que se discretiza un atributo continuo, por lo que tenemos *O(NAP)*, donde P en el número de particiones en cada atributo.
- Mejora estructural: la complejidad para esta etapa crece en forma cuadrática en función del número de atributos, considerando que en el peor de los casos se construyen y evalúan $O(A_2)$ clasificadores, donde A es el número de atributos involucrados y con los cuales se pueden realizar dos operaciones (eliminación o unión).

La complejidad temporal del método ACBE se corresponde con la más alta de los pasos anteriores. Es decir, ACBE tiene complejidad de $O(NA_2P)$, lo que supone que es cuadrática con respecto al número de atributos y lineal en cuanto al número de datos.

El análisis para el método dinámico lo podemos separar en dos partes:

- Creación de la red inicial
- Creación de la red de transición

Para la Creación de la red inicial, se consideran (método propuesto) las siguientes etapas:

- Inicialización.
- Discretización.
- Mejora estructural.
- Determinación de nodos ocultos.

Las tres primeras etapas son similares al método estático, por lo que representan la misma complejidad, la etapa de determinación del nodo oculto utiliza el algoritmo EM el cual es rápido pero requiere de reescribir los ejemplo para completar la BD, por lo que la complejidad esta dada por *O(AN)*. La complejidad del método ACBD se corresponde también con la más alta de los pasos anteriores. Es decir, ACBD tiene la complejidad $O(NA_2P)$, lo que supone un costo computacional aceptable. El único parámetro de ACBD que afecta a la complejidad y que es ajeno a los datos de entrenamiento es *r*, el número de estados que, en el peor de los casos alcanzará el valor máximo de *M* (10 en nuestros entrenamientos).

Para la Creación de la red de transición se considera la etapa:

- Çlasificación dinámica:

En esta última etapa se utilizó una generalización de redes bayesianas dinámicas los HMM y la complejidad temporal para el cálculo de este es la correspondiente a el algoritmo Backward−Forward o Baum−Welch *O(QXT)* ya que la topología del modelo es izquierda-derecha (topología lineal en la que una vez se abandona un estado ya no se puede regresar al mismo), donde Q es el número de estados, X el espacio de observaciones y T número de vectores de la secuencia.

Apéndice B

Publicaciones originadas

La investigación llevada a cabo durante la realización de esta tesis, ha producido las publicaciones que se citan a continuación:

- [Martinez−Arroyo, M. et al., 2006]: Bayesian Model Combination and Its Application to Cervical Cancer Detection. Book Series Lecture Notes in Computer Science, Publisher Springer Berlin / Heidelberg, ISSN 0302-9743, Computer Science, Volume Volume 4140/2006. Book Advances in Artificial Intelligence - IBERAMIA-SBIA 2006 Pages 622-631
- [Martinez−Arroyo, M.; Sucar, L.E.,2006]: Learning an Optimal Naive Bayes Classifier . Pattern Recognition, 2006. ICPR 2006. 18th International Conference on Volume 4, Issue , 20-24 Aug. 2006 Page(s): 958 958, Digital Object Identifier 10.1109/ICPR.2006.749
- [Martinez−Arroyo, M.; Sucar, L.E.,2006]: Learning an Optimal Naive Bayes Classifier . 36° Congreso de Investigación y Desarrollo del Tecnológico de Monterrey congreso CIDTEC 2006, Enero de 2006.
- [Martinez−Arroyo, M.; Sucar, L.E.,2002]: Aprendizaje de clasificadores bayesianos dinámicos . Open Discusión Track Proceedings, IBERAMIA 2002, VIII Iberoamerican Conference on Artificial Intelligence, Sevilla España. Nov. 12-15 2002. Ed. F. Garijo, J.C. Riquelme, M. Toro.

Apéndice C

Glosario de términos

- Árboles: estructura de red bayesiana donde los nodos tienen un solo padre [Pearl, 1988].
- Aprendizaje estructural: consiste en obtener la estructura de una red bayesiana, desribiendo las relaciones de dependencia e independencia entre las variables involucradas [Neapolitan, 1990].
- Aprendizaje paramétrico: consiste en que a partir de datos y de una estructura conocida de una red bayesiana se obtengan las probabilidades *a priori* y condicionales requeridas [Neapolitan, 1990].
- Discretización: proceso que consiste en transformar valores de una variable continua en un conjunto de valores discretos, esto es, divide el dominio de la variable continua en un número finito de intervalos que cubren completamente dicho dominio [Yang and Weeb, 2006].
- Red Bayesiana: grafo acíclico dirigido (DAG) en la que se tiene un conjunto de nodos que representan a las variables y un conjunto de arcos dirigidos entre pares de nodos. Cada variable, un conjunto finito de valores y para cada nodo existe una función de probabilidad condicional [Pearl, 1988].

Referencias

[Acosta-Mesa et al., 2005] Acosta-Mesa, H. G., Zitová, B., Ríos-Figueroa, H., Cruz-Ramírez, N., Marín-Hernández, A., Hernández-Jiménez, R., Cocotle-Ronzón, B., and Hernández-Galicia., E. (2005). Cervical cancer detection using colposcopic images: a temporal approach. *Proc. ENC, IEEE Press*.

[Bay, 2000] Bay, P. (2000). Multivariate discretization of continuous variables for set mining. *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*, page 315319.

[Binder et al., 1997] Binder, J., Koller, D., Russell, S., Kanazawa, K., and Binder, P. (1997). Adaptive probabilistic networks with hidden variables. *Kluwer Academic Plublishers, Boston.*

[Birdwell et al., 2005] Birdwell, R., Bandodkar, P., and Ikeda, D. (2005). Computer-aided detection with screening mammography in a university hospital setting. *Journal of Radiology, Vol. 236, No. 2*, pages 451–457.

[Bluman, 1992] Bluman, A. G. (1992). Elementary statistics, a step by step approach. *Brown Publishers.*

[Campos and Puerta., 2000] Campos, L. M. and Puerta., J. M. (2000). Learning dynamic belief networks using conditional independence tests. *proceedings of the 8th International Conference on Information Processing and Management of Uncertainty. Vol I*, pages 325–332.

[Catlett, 1991] Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. *Proceedings Eds. Y. Kodratoff of the eropean Working Sesion on Learning, Berlin, Germany: Springer-Verlag*, pages 164–178.

[Cestnick, 1990] Cestnick, B. (1990). Estimating probabilities: A crucial task in machine learning. *Proceedings of the 9th European Conference on Artificial Intelligence.*, page 147149.

[Cheng, 1998] Cheng, J. (1998). Powerconstructor systems. *http:// www. Cs. aulberta. Ca/ jcheng/bnpc.html*.

[Cheng et al., 1997] Cheng, J., Bell, D. A., and Liu, W. (1997). An algorithm for bayesian belief network construction from data. *In Proceedings of AI STAT97*, pages 83–90.

[Cheng and Greiner, 1999] Cheng, J. and Greiner, D. A. R. (1999). Comparing bayesian network classifiers. *Proceedings of the Fifteenth Conference on Uncertainty in Arificial intelligence, UAI'99.*, pages 101–108.

[Chmielewski, 2005] Chmielewski, P. (2005). Ocultas. *Proc.*, pages 11–20.

[Chow and Liu, 1968] Chow, C. K. and Liu, C.Ñ. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Trans. On Information Theory*, 14:462–468.

[Cooper and Herskovits, 1992] Cooper, G. and Herskovits, F. (1992). A bayesian method for the induction of probabilistic network from data. *Machine Learning*, 9:309–347.

[Dempster and Laird, 1977] Dempster, A. P. and Laird, N. M. (1977). Maximum likelihood estimation from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38.

[Dougherty et al., 1995] Dougherty, J., Kohavi, R., and Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. *Machine Learning: Proceedings of the Twelfth International Conference on Artificial Intelligence*, pages 194–202.

[Duda and Hart, 1963] Duda, R. and Hart, P. (1963). Pattern classification and scene analysis. *John wiley and Sons.*

[Ezawa and Norton, 1995] Ezawa, K. and Norton, S. W. (1995). Knowledge discovery in telecommunication services data using bayesian network models.,. *Proceedings 1^a International Conference on Knowledge Discovery Data Minin (KDD).*, pages 100–105.

[Fayyad et al., 1996] Fayyad, U. M., Piatetsy-Shapiro, G., and Smyth, P. (1996). From data minning to knowledge discovery: An overview. *Advances in knowledge Discovery and Data Minning. AAAI Press and the MIT Press. Chapter 1.*, pages 1–34.

[Ferrarri, 2005] Ferrarri, L. D. (2005). Minning housekeeping genes with a naive bayes classifier. *MSc Thesis, University of Edinburgh, Shool of Informatics*.

[Ferrarri, 2006] Ferrarri, L. D. (2006). Tesis doctoral: Aportaciones al diagnóstico de cáncer asistido por ordenador. *Universidad Politécnica de Valencia,Depto. de Sistemas Informáticos y Comuputación.*

[Frasconi and Vullo., 2001] Frasconi, P. and Vullo., A. (2001). Text categorization for multi-page documents; a hybrid naive bayes hmm aproach. *Proc. of the ACM/IEEE Joint Conf. on Digital Libraries*, pages 11–20.

[friedman et al., 1999] friedman, N., Boyen, X., and Koller, D. (1999). Discovering the hidden structure of complex dynamic systems. *Proceedings of the Fifteenth Conference on Uncertainty in Arificial intelligence, UAI99*, pages 139–147.

[Friedman et al., 1997] Friedman, N., Geiger, D., and Goldszmid, M. (1997). Bayesian networks classifiers. *Machine Learning No. 29*, pages 131–161.

[Ghahramani and Jordan, 1994] Ghahramani, Z. and Jordan, M. (1994). Learning from incomplete data ai. *Memo 1509, MIT AI Lab*, pages 11–20.

[Gillies et al., 1996] Gillies, F., Duncan, and Kwoh, C.-K. (1996). Using hidden nodes in bayesian networks. *Artificial intelligence 88*, pages 1–38.

[Holte, 1993] Holte, R. C. (1993). Very simple classification rules perform well on most commoly used databases. *Machine Learning 11*, pages 63–90.

[Holte and Porter, 1989] Holte, R. C. and Porter, B. W. (1989). Concept learning and the problem of small disjuncts. *Proceedings of rhe eleventh international Joint Conference on Artificial Intelligence*, pages 813–818.

[Jhon and Kohavi, 1997] Jhon, G. and Kohavi, R. (1997). Wrappers for feature subset selection. *Artificial Intelligence and Proceeding of the Eleventh International Conference on Machine Learning (1994).*, pages 121–129.

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Trans. ASME, Journal of Basic Engineering*, pages 34–45.

[Kerber, 1992] Kerber, R. (1992). Chimerge: discretization of numeric attributes. *Proceeding of tenth National Conference on Artificial Intelligence, MIT Press.*, pages 123–128.

[Kohavi, 1997] Kohavi, P. (1997). Mlc++. *Proc. of the ACM/IEEE Joint Conf. on Digital Libraries*, pages 11–20.

[Kononenko, 1991] Kononenko, P. (1991). Semi-naivebayesian classifiers. *Proc. Sixth European Working Session on Learning. berling: Springer-Verlag.*, pages 206–219.

[Kullback and Leiber, 1951] Kullback, S. and Leiber, R. A. (1951). On information and sufficiency. *Annals of Mathematicals Static*, 22:76–86.

[Lacave and Díez, 2005] Lacave, C. and Díez, J. (2005). Elvira 0.11. graphical user interface. *Manual de usuario, UCLM and UNED, España.*

[Lam and Bacchus, 1994] Lam, W. and Bacchus, F. (1994). Learning bayesian belief netwoks: An aproach based on the mdl principle. *Computational Intelligence*, 10:269–293.

[Langley and Sage, 1994] Langley, P. and Sage, S. (1994). Induction of selective bayesian classifiers. *Proceeding of the Tenth Conference on Uncertainty in Artificial Intelligence. Seattle, WA.*

[Lauritzen, 1995] Lauritzen, S. L. (1995). The em algorithm for grafical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201.

[Little and Rubin, 1997] Little, R. J. A. and Rubin, D. B. (1997). h. *Statistical Analysis with Missing Data. John Wiley Sons*.

[Lowd and Domingos, 2005] Lowd, D. and Domingos, P. (2005). Naive bayes models for probability estimation. *Proceedings of the Twenty-Second International Conference on Machine Learning*, pages 529–536.

[Lowd, 1996] Lowd, F. (1996). Comparing bayesian network classifiers. *Using hidden Nodes in Bayesian Networks.*, pages 1–38.

[Maass, 1994] Maass, W. (1994). Efficient agnostic pac-learning with simple hypothese. *Proceedings of the 7th Annual ACM Conference on Computational Learning*, pages 67–75.

[Martinez and Sucar, 1998] Martinez, M. and Sucar, L. E. (1998). Interactive structural learning of bayesian networks. *Expert Systems with Applications, Editors PERGAMON.*, pages 325–332.

[Montero and Sucar, 2004] Montero, J. A. and Sucar, L. E. (2004). Feature selection for visual gesture recognition using hidden markov models. *Fifth Mexican International Conference on Computer Science*, pages 196–203.

[Moore and S., 1994] Moore, A. W. and S., L. M. (1994). Efficient algorithms for minimizing cross validation error. *International Conference On Machine Learning*.

[Murphy and Aha., 1994] Murphy, P. M. and Aha., D. W. (1994). Uci repository of machine learning databases. *For Information Contact ml-repository@ics.uci.edu.*, pages 11–20.

[Neapolitan, 1990] Neapolitan, R. E. (1990). Propabilistic reasoning in expert systems: Theory and algoritms. *John Willey Sons, INC. New York, E.U.A.*

[Pazzani, 1996] Pazzani, M. J. (1996). Searching for atribute dependencies in bayesian classifiers. *Preliminary Paper of the Intelligence and Statistic*, pages 424–429.

[Pearl, 1988] Pearl, J. (1988). Probabilistic reasoning in intelligent systems. *Morgan Kaufmann, palo Alto Calif. U.S.A.,*.

[Quinlan, 1993] Quinlan, J. R. (1993). C4.5: programs for machine learning. *Morgan Kaufmann*.

[Rissanen, 1978] Rissanen (1978). Modeling by shortest data description automatic. pages 465–471.

[Rubin, 1987] Rubin, D. (1987). Multiple imputation for nonresponse in surveys. *New York: John Wiley sons.*, pages 11–20.

[Samuels and Witmer, 1999] Samuels, M. L. and Witmer, J. A. (1999). Statistics for the life sciences. *Second Edition. Prentice-Hall.*

[Singh, 1994] Singh, M. (1994). Learning bayesian networks from incomplete data. *American Association for Artificial Intelligence (www.aaai.org). Copyrigth © 1997*, pages 67–75.

[Spirtes et al., 1993] Spirtes, P., Glymour, C., and R. Scheines, R. (1993). Causation, prediction and seach. *http://hss.cmu.edu/html/departments/philosophy/TETRAD.BOOK/book.html*, pages 11–20.

[Sucar and Gillies, 1993] Sucar, L. E. and Gillies, D. F. (1993). Probabilistic reasoning in high-level vision. *Image and Vision Computing Journal*, 12:42–60.

[Sucar et al., 1994] Sucar, L. E., Gillies, D. F., and Guillies, D. A. (1994). Objetive probabilistic in expert systems. *Artificial Intelligence*, 61:187–208.

[Ting, 1994] Ting, K. M. (1994). Discretization of continuous-valued attributes and instance-based learning. *Technical Report 491, University of Sydney.*

[Valdés et al., 2003] Valdés, J., Molina, L. C., and N. Peris, N. (2003). An evolution strategies approach to the simultaneous discretization of numeric attributes in data mining. *Proceedings of the World Congress on Evolutionary Computation, NRC 46536. Canberra, Australia. December 8-12, 2003. IEEE Press 03TH8674C, ISBN 0-7803-7804-0.*, pages 1967–1964.

[Witten and Frank, 2005] Witten, I. H. and Frank, E. (2005). Data mining: Practical machine learning tools and techniques, 2nd edition, morgan kaufmann, san francisco. *Data mining book.*

[Yang and Weeb, 2002] Yang, Y. and Weeb, G. (2002). A comparative study of discretization methods for naive-bayes classifiers. *In Proceedings of the Pacific Rim Knowledge Acquisition Workshop (PRKAW)*.

[Yang and Weeb, 2005] Yang, Y. and Weeb, G. (2005). Discretization methods. *In O Maimon and L Roakch, Data Mining and Knoeledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, Kluwer Academic Publishers.*

[Yang and Weeb, 2006] Yang, Y. and Weeb, G. (2006). Discretization. *In J Wang (eds)Encyclopedia of Data Warehousing and Mining Information Science Publishing*.

[zhang et al., 2005] zhang, H., Jiang, L., and Su, J. (2005). Hidden naive bayes. *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*.

[Zhang et al., 2007] Zhang, J., Khang, D.-K., Silvescu, A., and Honavar, V. (2007). Extended version of learning compact and accurate naive bayes classifiers from attribute value taxonomies and data journal of knowledge and information systems. *This paper is an extended version of a paper published in the 4th IEEE International Conference on Data Mining.*