# Clasificadores Bayesianos: de Datos a Conceptos

Luis Enrique Sucar, Investigador Titular
Instituto Nacional de Astrofśica, Óptica y Electrónica
Tonantzintla, Puebla, México
esucar@inaoep.mx

**Resumen**

Clasificar es transformar datos en conceptos. El desarrollo de clasificadores eficaces y eficientes es un problema actual relevante en ingeniería computacional, dada la gran cantidad de datos que se generan cada día. Las cámaras digitales, los sistemas de video-vigilancia, las bases de datos de empresas y las redes socieales, entre otras, generan grandes cantidades de datos que es necesario abstraer de forma de tener una representación más compacta para la toma de decisiones.

En este trabajo desarrollamos una familia de clasificadores basados en el paradigma bayesiano y en modelos gráficos. Además de ser eficaces para resolver diversos problemas de clasificación, todo los modelos propuestos son eficientes en términos computacionales tanto en espacio de almacenamiento como en tiempo de cómputo. Desarrollamos tres tipos de clasificadores: (i) el clasificador bayesiano semi-simple que mejora al clasificador bayesiano simple; (ii) el clasificador bayesiano en cadena, para problemas de clasificación multidimensional; y (iii) el clasificador jerárquico multidimensional, para dominios en que las clases forman una jerarquía.

Mostramos la aplicación de los tres tipos de clasificadores en diversos problemas prácticos: detección de personas en imágenes, selección de fármacos para pacientes con VIH, y clasifiación de galaxias en placas astronómicas.

**Palabras Clave**: computación, clasificación, redes bayesianas, aprendizaje.

## 1 Introducción

Clasificación consiste en asignar un *objeto* (instancia, dato) a una *clase* (categoría). Por ejemplo, podemos clasificar una imagen como paisaje, retrato, urbana, etc. Otro ejemplo es asignar palabras a categorías gramaticales: sustantivo, verbo, adjetivo, etc. El clasificar lo que percibimos con los sentidos es algo natural en el ser humano; básicamente esto nos permite abstraer la información, llevándola a una representación más adecuada para la toma de decisiones. Esta capacidad es esencial para la supervivencia; clasificando, por ejemplo, un animal como *comida* o *peligro*.

La clasificación es también muy importante en el desarrollo de sistemas computacionales para muchas aplicaciones, por ejemplo:

- Control de calidad en la industria: clasificar una pieza o producto como correcta o defectuosa.
- Sistemas de seguridad: identificar, por ejemplo, si una persona tiene acceso o no cierto lugar.
- Vehículos inteligentes: detectar peatones en el camino, clasificando los *objetos* que se detectan usando cámaras u otros sensores.
- Lectores de correo electrónico: filtrar mensajes que sean "basura" (spam).
- Análisis de imágenes médicas: detectar tumores en rayos-X.
- Sistemas biométricos: asignar una imagen de una huella a la persona correspondiente.

Por lo tanto, es importante diseñar clasificadores, ya sea en *hardware* o *software*, que puedan ayudar a resolver dichos problemas.

Desde un punto de vista matemático, el proceso de clasificación consiste en asignar una clase, $c$, de un conjunto de clases, $C$, a cierta instancia, representada por un vector de características o atributos, $\mathbf{X} = X_1, X_2, ..., X_m$. Hay dos tipos básicos de clasificadores:

**No supervisado o agrupamiento:** en este caso las clases son desconocidas, y el problema consiste en dividir un conjunto de $n$ objetos en $k$ clases, de forma que a objetos *similares* se les asigna la misma clase.

**Supervisado:** las clases se conocen *a priori*, y el problema consiste en encontrar una función que asigne a cada objeto su clase correspondiente.

En este trabajo nos enfocamos a clasificación supervisada. Entonces, el problema consiste en encontrar una función que realice un mapeo de los atributos del objeto a su clase correspondiente, esto es: $c = f(\mathbf{X})$. En general, es difícil construir dicha función, por lo que se utilizan técnicas de aprendizaje computacional para obtener la función a partir de datos –ejemplos de objetos en que se especifican sus características y la clase correspondiente. Este conjunto de datos, $D$, se compone de $n$ ejemplos, cada uno a su vez compuesto de un vector de atributos y la clase correspondiente: $(\mathbf{X_1}, c_1), ..., (\mathbf{X_n}, c_n)$.

Hay varios criterios en base a los cuales se evalúa un clasificador:

- Exactitud: proporción de clasificaciones correctas.
- Rapidez: tiempo que toma hacer la clasificación.
- Claridad: que tan comprensible es para los humanos.
- Tiempo de aprendizaje: tiempo para entrenar o ajustar el clasificador a partir de datos.

En este trabajo nos enfocamos a los primeros dos criterios: eficacia del clasificador y eficiencia tanto en espacio de memoria como en tiempo de cómputo.

Existen diversas técnicas para desarrollar clasificadores, como árboles de decisión, reglas de clasificación, redes neuronales, etc. [7]. Entre estas, el enfoque probabilístico bayesiano provée un marco formal para construir clasificadores *óptimos* bajo ciertos criterios (como el minimizar el error clasificación o el costo de una mala clasificación). Sin embargo, si aplicamos el enfoque bayesiano en una forma directa, la complejidad computacional (en memoria y tiempo) crece exponencialmente con el número de atributos de los objetos. Una alternativa para enfrentar este problema, es el *clasificador bayesiano simple*, que asume que todos los atributos de un objeto son estadísticamente independientes dada la clase. Esto hace que la complejidad crezca linealmente con el número de atributos, pero puede hacer que la efectividad del clasificador decrezca si los atributos no son realmente independientes dada la clase.

En este trabajo planteamos un método de mejora de un clasificador bayesiano simple, de forma que se mantenga la eficacia aunque los atributos no sean independientes, y al mismo mantenga una complejidad lineal. Posteriormente, extendemos este método para construir clasificadores *multidimensionales*, donde cada objeto puede pertenecer a varias clases a la vez (por ejemplo, un documento puede a la vez hablar de política y finanzas); y a clasificadores *jerárquicos*, donde las clases están organizadas en una jerarquía (por ejemplo, una taxonomía de animales). Todos estos clasificadores mantienen una alta eficacia y eficiencia, y se ilustra su aplicación para resolver diversos problemas prácticos.

# 2 Clasificación Bayesiana

Desde un enfoque bayesiano, el problema de clasificación supervisada consiste en asignar a un objeto descrito por un conjunto de atributos o características, $X_1, X_2, ..., X_n$, a una de $m$ clases posibles, $c_1, c_2, ..., c_m$, tal que la probabilidad de la clase dados los atributos se maximiza:

$$Arg_C[MaxP(C \mid X_1, X_2, ..., X_n)] \tag{1}$$

Si denotamos el conjunto de atributos como: $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, la ecuación 1 se puede escribir como: $Arg_C[MaxP(C \mid \mathbf{X})]$. La formulación del *clasificador bayesiano* se base en utilizar la regla de Bayes para calcular la probabilidad posterior de la clase dados los atributos:

$$P(C \mid X_1, X_2, ..., X_n) = P(C)P(X_1, X_2, ..., X_n \mid C)/P(X_1, X_2, ..., X_n) \tag{2}$$

Que se puede escribir de forma más compacta como:

$$P(C \mid \mathbf{X}) = P(C)P(\mathbf{X} \mid C)/P(\mathbf{X}) \tag{3}$$

Así que el problema de clasificación basado en la ecuación 3, se puede expresar como:

$$Arg_C[Max[P(C \mid \mathbf{X}) = P(C)P(\mathbf{X} \mid C)/P(\mathbf{X})]] \tag{4}$$

El denominador, $P(\mathbf{X})$, no varía para las diferentes clases, por lo que se puede considerar como una constante si lo que interesa es maximizar la probabilidad de la clase:

$$Arg_C[Max[P(C \mid \mathbf{X}) = \alpha P(C)P(\mathbf{X} \mid C))]] \tag{5}$$

Basado en la ecuación 5, para resolver un problema de clasificación bajo el enfoque bayesiano, se requiere la probabilidad *a priori* de cada clase, $P(C)$, y la probabilidad de los atributos dada a clase, $P(\mathbf{X} \mid C)$, conocida como *verosimilitud*; para obtener la probabilidad posterior$P(C \mid \mathbf{A})$. Entonces, para *aprender* este clasificador de un conjunto de datos, se require estimar estas probabilidades, *a priori* y verosimilitud, a partir de los datos, conocidos como los parámetros del clasificador.

Como mencionamos en la introducción, la aplicación directa de la ecuación 5, resulta en un sistema muy complejo al implementarlo en una computadora, ya que el término $P(X_1, X_2, ..., X_n \mid C)$, incrementa exponencialmente de tamaño en función del número de atributos; resultando en un requerimiento muy alto de memoria para almacenarlo en una computadora, y también el número de operaciones para calcular la probabilidad crece significativamente. Una alternativa es considerar ciertas relaciones de dependencia mediante lo que se conoce como el *clasificador bayesiano simple*.

## 2.1 Clasificador Bayesiano Simple

El clasificador bayesiano simple (CBS) se basa en la suposición de que todos los atributos son independientes dada la clase; esto es, cada atributo $X_i$ es condicionalmente independiente de los demás atributos dada la clase: $P(X_i \mid X_j, C) = P(X_i \mid C)$, $\forall j \neq i$. Bajo estas consideraciones, la ecuación 2 se puede escribir como:

$$P(C \mid X_1, X_2, ..., X_n) = P(C)P(X_1 \mid C)P(X_2 \mid C)...P(X_n \mid C)/P(\mathbf{X}) \tag{6}$$

donde $P(\mathbf{X})$ se puede considerar como una constante de normalización.

El CBS reduce drásticamente la complejidad del clasificador bayesiano en espacio y tiempo de cálculo. En cuanto a espacio de memoria, se requiere la probabilidad previa de las $m$ clases (vector de $1 \times m$), y las $n$ probabilidades condicionales de cada atributo dada la clase (si suponemos que los atributos son discretos con $k$ posibles valores, esto implica $n$ matrices de $m \times k$). Básicamente el espacio requerido aumenta linealmente con el número de atributos. También el cálculo de la probabilidad posterior se vuelve muy eficiente, ya que se requieren del orden de $n$ multiplicaciones para calcular la probabilidad posterior de cada clase dados los atributos (complejidad lineal).

Podemos representar gráficamente la estructura de un clasificador bayesiano simple utilizando los principios de los modelos gráficos probabilistas [6], donde las independencia condicionales entre las variables se representan mediante un grafo. El CBS tiene una estructura de estrella, con la clase en el medio y arcos dirigidos de la clase a cada atributo. Esto expresa que los atributos dependen de la clase y son independientes entre si dada la clase (no hay arcos directos entre los atributos). Una representación gráfica del CBS se muestra en la figura 1.

Para aprender un CBS se requiere la probabilidad previa de cada clase, $P(C)$, y la probabilidad condicional de cada atributo dada la clase, $P(X_i \mid C)$. Estas probabilidades se pueden obtener mediante estimados subjetivos de expertos en el dominio,

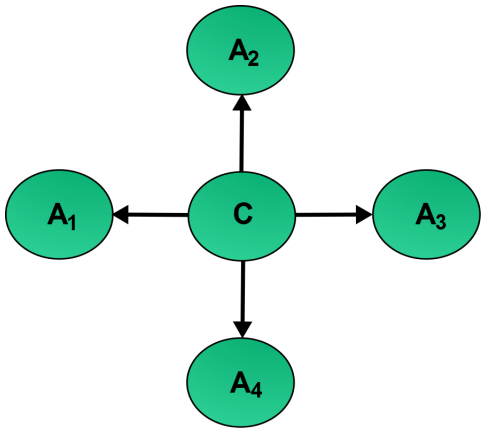


Figura 1: Representación gráfica de un CBS, con la variable clase, $C$, y 4 atributos, $A_1, ..., A_4$.

o a partir de datos mediante *máxima verosimilitud* (consiste en que las probabilidades se aproximan por las estadísticas de los datos). Por ejemplo, consideremos un clasificador para decidir si jugar golf (dos clases, jugar y no-jugar) en base a las condiciones ambientales. Entonces podemos aprender los parámetros (probabilidades) para este clasificador a partir de ejemplos de condiciones ambientales (atributos) y si se jugó o no (clase); un ejemplo hipotético de un modelo se ilustra en la figura 2, incluyendo algunas de las tablas de probabilidad.

El clasificador bayesiano simple provee un esquema muy eficiente para construir clasificadores, el cual da muy buenos resultados en muchas aplicaciones. Sin, embargo en algunos casos su efectividad disminuye al no satisfacerse las condiciones de independencia. En la siguiente sección planetamos una extensión al CBS que ataca esta limitación, y a la vez mantiene su eficiencia.

# 3 Clasificador Bayesiano Semi-simple

La idea del clasificador bayesiano *semi-simple* (CBSS) es transformar la estructura básica del clasificador bayesiano simple para lidiar con atributos que no son independientes, pero a la vez mantener la misma eficiencia del CBS. Para esto, se propone una metodología que mejora la estructura inicial del CBS, mediante transformaciones locales a dicha estructura. Se consideran dos operaciones bśicas: (i) eliminar una variable (un nodo si lo vemos como un grafo), (ii) unir dos variables en una sola (combinar dos nodos). La idea se ilustra en forma gráfica en la figura 3; y abajo se explican a detalle ambas operaciones.

*Eliminación* de una variable consiste simplemente de eliminar un atributo, $X_i$ del clasificador, lo que puede ser por dos razones: (a) el atributo no es relevante para la clase, (b) el atributo no es independiente de algún otro atributo, $X_j$. La razón para (b) es que si dos atributos son altamente dependientes, dan básicamente la misma infor-

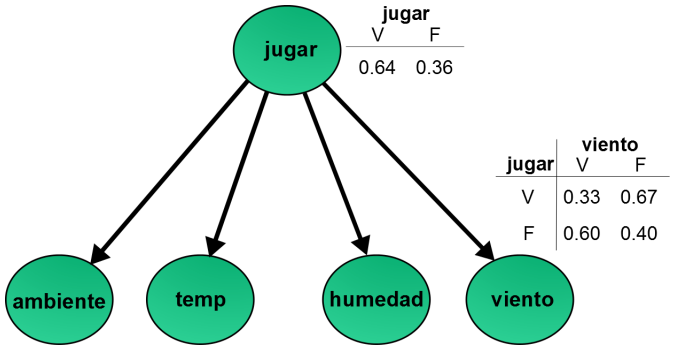


Figura 2: Un CBS para decidir si jugar golf basado en 4 atributos: ambiente, temperatura (temp), humedad y viento. Se muestran dos de las tablas de probabilidad, $P(jugar)$ y $P(viento \mid jugar)$, asumiendo que ambas variables son binarias.

mación, as que uno es redundante y se puede eliminar.

*Combinación* de dos variables consiste en unir dos atributos, $X_i$ y $X_j$,en un nuevo atributo, $X_k$ (asumiendo atributos discretos). Por ejemplo, si $X_i = a, b, c$ y $X_j = 1, 2$, entonces $X_k = a1, a2, b1, b2, c1, c2$. Esta operación se utiliza cuando los dos atributos no son independientes dada la clase, ya que al unirlos ya no importa si no son independientes.

Entonces se tienen dos alternativas cuando se encuentras atributos que no son independientes dada la clase: eliminar uno de ellos o unirlos en un solo atributo. Se selecciona una de las dos en base a cual provee una mejora mayor en la efectividad del clasificador.

Basado en las operaciones anteriores, a continuación se describe un algoritmo para mejora estructural de un CBS:

**Algoritmo de Mejora Estructural**
**Entrada**: CBS y conjunto de datos.
**Salida**: CBSS mejorado

1. Se estima la dependencia entre cada atributo y la clase (usando una medida de información mutua), y se eliminan aquellos atributos que no proveen información a la clase (información mutua debajo de un umbral).

2. Los demás atributos se prueban, por pares, para ver si son independientes dada la clase (usando una medida de información mutua condicional). Si un par de atributos, $X_i$ y $X_j$, no son independientes dada la clase, se pasa al paso (3).

3. El par de atributos, $X_i$ y $X_j$, son considerados para eliminación o para combinación. Se evalan las 3 alternativas, eliminar $X_i$, eliminar $X_j$, o unirlos en un solo atributo, mediante su impacto en la exactitud del clasificador (utilizando datos de validación) y se selecciona la alternativa que de mejor resultado.

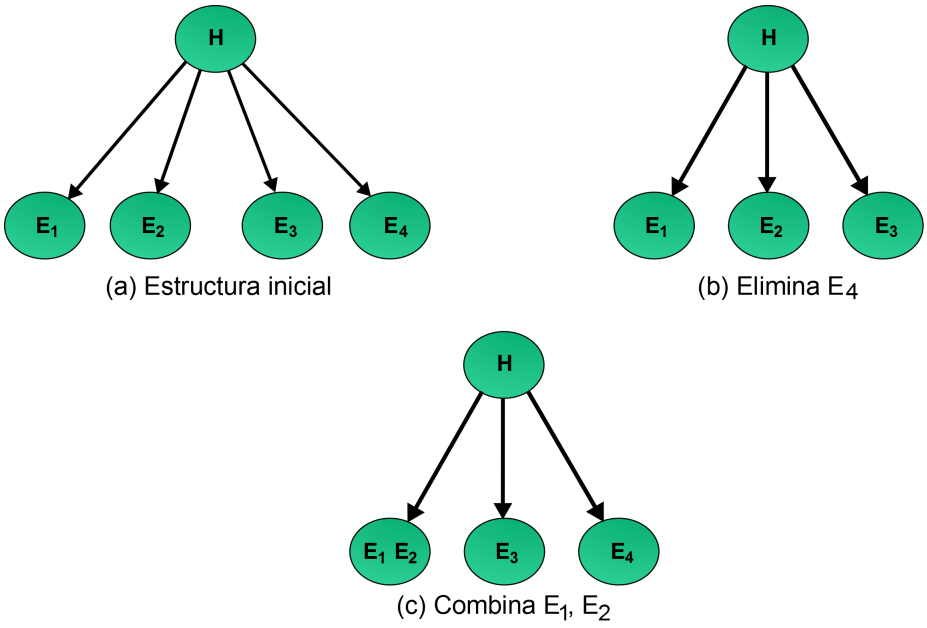


Figura 3: Mejora estructural de un CBS: (a) estructura original, (b) un atributo es eliminado (c) dos atributos son combinados en una variable.

Los pasos (2) y (3) se repiten hasta que ya no existan más atributos que no sean independientes.

El CBSS mantiene una complejidad computacional (en promedio), tanto en espacio como en tiempo de clasificación, similar a la del CBS. En el peor caso, si se llegaran a combinar todos los atributos en uno solo, se volvería exponencial, pero esto es muy raro en la práctica.

A continuación se presenta una aplicación de este algoritmo a clasificación de imágenes.

## 3.1 Clasificación visual de piel

La detecciíon de personas en imágenes tiene muchas aplicaciones actualmente, como en sistemas de seguridad, interacción hombre-máquina, reconocimiento de ademanes, etc. Una forma muy sencilla y rápida de contar con un detector inicial de personas es el clasificar los pixeles de la imagen en $piel$ o $no - piel$ en base a sus atributos de color. Este clasificador puede ser suficiente en algunas aplicaciones, y en otras proveer un procesamiento inicial de la imagen, para luego utilizar otros métodos más sofisticados en las regiones *candidatas*.

Usualmente loas pixeles en una imagen digital se representan como las combinación de 3 colores primarios: Rojo (R), Verde (G) y Azul (B), en lo que se conoce como el modelo $RGB$ (iniciales en inglés). Cada componente de color puede tomar diferentes valores numéricos usualmente en el rango $0...255$. De este forma podemos

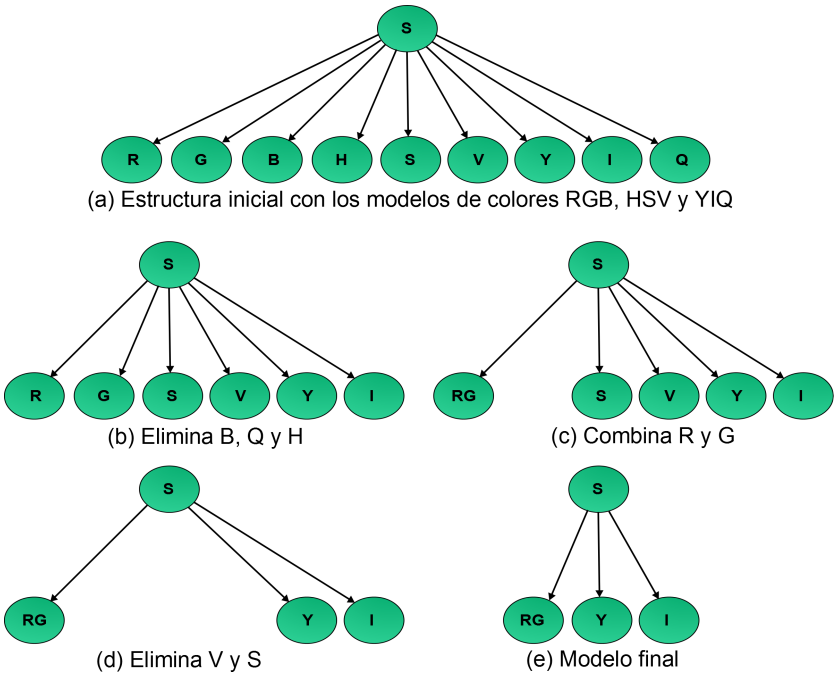


Figura 4: La figura ilustra el proceso de mejora de un CBS para clasificación de pixeles como piel / no-piel, a partir de un modelo inicial con 9 atributos (a) hasta llegar al modelo final con 3 atributos (e).

construir un CBS para piel tomando estos 3 atributos: R, G, B. Sin embargo, existen otras representaciones o modelos de color, como $HSV$, $YIQ$, etc. Así que puede ser que otro modelo de mejores resultados, e incluso un combinación de atributos de diferentes modelos.

Una alternativa es considerar un clasificador bayesiano semi-simple utilizando el algoritmo de mejora estructural, incluyendo inicialmente varios modelos de color. Para ello consideramos 3 modelos de color, RGB, HSV e YIQ, de forma que inicialmente se tienen 9 atributos en total. Este clasificador inicial se entrenó (para estimar las probabilidades) con ejemplos de imágenes de piel y no-piel. Posteriormente el clasificador de optimizó mediante el algoritmo de mejora estructural de la sección anterior. La secuencia de operaciones y la estructura resultante se ilustran mediante sus modelos gráficos en la figura 4. Observamos que inicialmente el método elimina atributos irrelevantes o dependientes, luego combina dos atributos dependientes, y luego elimina otros dos atributos, resultando en una estructura final con 3 atributos : $R-G, Y, I$ (uno es la combinación de dos atributos originales).

Evaluamos experimentalmente tanto al clasificador original como al mejorado con imágenes en que hubiera pixeles de piel y de no-piel (diferentes a las de entrenamiento), resultando en una mejora de una precisión de 94% con los 9 atributos a una precisión del 98% con el CBSS mejorado. En estos rangos de precisión des difícil obtener 4

(a) Imagen original (b) Imagen con los pixeles de piel detectados

Figura 5: Ejemplo de una imagen en que los pixeles han sido clasificados como piel (en negro) y no piel.

puntos de mejora, y además se tiene un sistema incluso más eficiente al reducirse los atributos de 9 a 3. Un ejemplo de una imagen que muestra los pixeles de piel detectados con el clasificador se muestra en la figura 5.

# 4 Clasificadores Multidimensionales

Los clasificadores tradicionales consideran que cada objeto pertenece a una sola clase; sin embargo, hay muchas aplicaciones en que un objeto puede a la vez pertenecer a varias clases. Por ejemplo: clasificación de texto, donde un documento puede puede referirse a varios temas a la vez; clasificación de imágenes, donde un una imagen tiene varios objetos de diferentes clases; clasificación de genes, ya que un gen puede tener varias funciones; entre otras. Esto se conoce como clasificación *multidimensional*. Formalmente, un clasificador multidimensional establece una función $h$ que asigna a cada objeto representado por un vector de $n$ características $\mathbf{X} = (X_1, \ldots, X_n)$ a un vector de $d$ clases $\mathbf{C} = (C_1, \ldots, C_m)$. Dicha función $h$ debe asignar a cada objeto $\mathbf{X}$ la combinación más probable de clases:

$$\arg \max_{c_1,\ldots,c_d} P(C_1 = c_1, \ldots, C_m = c_m | \mathbf{X}) \tag{7}$$

Un conjunto de datos $D$ para un problema multidimensional se compone de $d$ ejemplos, en donde en cada uno se establece el vector de atributos y el vector de clases: $(\mathbf{X_1}, \mathbf{C_1}), ..., (\mathbf{X_d}, \mathbf{C_d})$. La clasificación multi-etiqueta es un caso particular de la clasificación multidimensional en la cual todas las clases son binarias. En este trabajo nos enfocamos al caso de clasificación multi-etiqueta.

Existen dos enfoques básicos para resolver el problema de clasificación multi-etiqueta, conocidos como: *relevancia binaria* y *conjunto potencia* [12]. El primero utiliza $m$ clasificadores binarios tradicionales (unidimensionales), y luego simplemente combina los resultados de estos clasificadores. Su principal limitación es que asume que todas las clases son independientes, lo cual no es necesariamente cierto, y esto puede afectar la precisión del clasificador. El segundo construye un solo clasificador donde la variable clase es el conjunto potencia de todas las clases (si se tienen $m$ clases,

se tendrían $2^m$ valores). El problema es que este enfoque se vuelve muy complejo al aumentar el número de clases.

Bajo un enfoque bayesiano una alternativa son los *clasificadores multidimensionales basados en redes bayesianas* (MBC, por sus siglas en inglés) [3], los cuales si consideran las dependencias tanto entre clases como entre atributos. Un MBC es un caso particular de red bayesiana [8] en la cual se tienen 3 tipos de arcos: entre clases, entre atributos y de clases a atributos. El problema es que este tipo de modelos son muy complejos computacionalmente, en particular cuando existen muchas clases.

Nos otros proponemos otro esquema que si considera las relaciones entre clases, pero es mucho más sencillo y eficiente.

## 4.1 Clasificador Bayesiano en Cadena

Read et al. [9] propusieron originalmente la idea de clasificadores en cadena para el problema de clasificación multidimensional. Un clasificador en cadena consiste de $m$ clasificadores binarios, uno por clase, que están relacionados mediante una cadena, de forma que cada clasificador incorpora como atributos adicionales a las clases de los clasificadores previos en la cadena. Esto es, el vector de características de cada clasificador se aumenta con las etiquetas de los $k$ clasificadores previos en la cadena:
$\mathbf{X} = (X_1, \ldots, X_n, L_1, \ldots, L_k)$.

Para clasificar un nuevo objeto, se empieza por el primer clasificador en la cadena, utilizando su resultado como entrada adicional al siguiente, y así sucesivamente. Como en el enfoque de relevancia binaria, se combinan los resultados de todos los clasificadores. Este enfoque es muy eficiente, similar al de relevancia binaria, pero tiene dos problemas: (i) no hay nada que indique como ordenar las clases en la cadena, y la eficacia del clasificador varía en función de este orden; (ii) si hay muchas clases, el número de atributos puede aumentar demasiado.

Nosotros proponemos un esquema alternativo que no tiene los problemas anteriores y que denominamos *clasificador bayesiano en cadena* (BCC, por sus siglas en inglés). La idea es considerar la relaciones de dependencia e independencia entre las clases para construir la cadena. A continuación se presenta una derivación matemática del BCC.

Si se aplica la regla de la cadena de teoría de probabilidad, se puede escribir la ecuación 7 como:

$$\arg \max_{C_1,\ldots,C_m} P(C_1|C_2, \ldots, C_m, \mathbf{X}) P(C_2|C_3, \ldots, C_m, \mathbf{X}) \ldots P(C_m|\mathbf{X}) \qquad (8)$$

Si se consideran las independencia condicionales entre las clases (como en redes baysianas), se puede simplificar la ecuación 8 eliminando de cada término las clases que sean independientes, de acuerdo al orden de la cadena; es decir dejando sólo sus *padres* si representamos las dependencias como grafo (red bayesiana). De forma que la ecuación 8 se puede escribir como:

$$\arg \max_{C_1,\ldots,C_m} \prod_{i=1}^{m} P(C_i|\mathbf{pa}(C_i), \mathbf{x}) \qquad (9)$$

donde $\mathbf{Pa}(C_i)$ son los padres de la clase $i$ de acuerdo al grafo de dependencia entre clases.

A continuación se hace una simplificación adicional, considerando que la combinación más probable de clases es simplemente la concatenación de los resultados de cada clasificador. Esto es, se resuelve el siguiente conjunto de ecuaciones como una aproximación a la ecuación 7:

$$\begin{aligned}
&\arg\max_{C_1} P(C_1|\mathbf{pa}(C_1), \mathbf{X}) \\
&\arg\max_{C_2} P(C_2|\mathbf{pa}(C_2), \mathbf{X}) \\
&\quad \cdots\cdots\cdots\cdots\cdots\cdots \\
&\arg\max_{C_m} P(C_m|\mathbf{pa}(C_m), \mathbf{X})
\end{aligned}$$

Esto corresponde a un BCC que hace las dos siguientes suposiciones:

1. La dependencia entre clases se puede representar como un grafo acíclico dirgido (red bayesiana).

2. La combinación más probable de las clases (lo que se conoce como el problema de abducción total) se aproxima como una concatenación de las clases individuales más probables.

El clasificador baysiano en cadena provee un buen compromiso para la clasificación multidimensional, al incorporar las dependencias entre clases y mantener una complejidad similar a relevancia binaria, que es básicamente lineal en cuanto al número de atributos y clases.

Para cada clasificador en la cadena utilizamos un clasificador bayesiano simple, que se aprende de igual manera, simplemente incorporando como atributos adicionales las clases *padres* de acuerdo a la estructura de dependencia de las clases. La idea general para construir un BCC se ilustra en la figura 6.

## 4.2 Selección de Fármacos para VIH

El *Virus de Inmuno-deficiencia Humana* (VIH) es el agente que produce el SIDA, una condición en la cual se tiene una falla progresiva del sistema inmune que permite una serie de infecciones oportunísticas que pueden ocasionar la muerte. Para combatir el VIH se han desarrollado una serie de drogas que se pueden dividir en varias tipos, cada tipo combate el virus afectando pasos específicos de su ciclo reproductor. La terapia contra el virus del VlH consiste generalmente de una combinación de drogas, normalmente tres o cuatro. La selección de drogas depende de la condición de paciente, caracterizada por las mutaciones del virus presentes en el paciente. Por lo tanto, es importante seleccionar la mejor combinación de drogas de acuerdo a las mutaciones del virus presentes.

Seleccionar el mejor conjunto de drogas para un paciente con VIH se puede ver como un problema de clasificación multidimensional; en donde las clases son los diferentes tipos de drogas y los atributos son las mutaciones presentes en el virus. Esto se

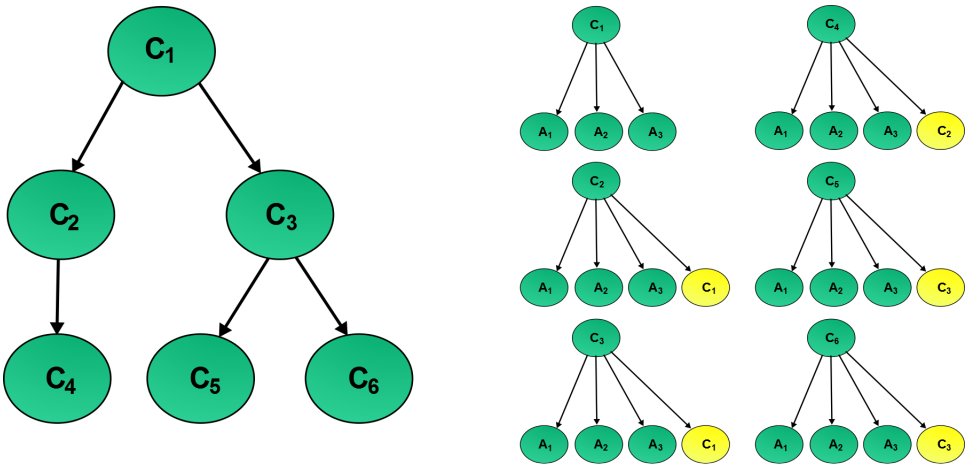


Figura 6: Un ejemplo de un BCC. Izquierda: un grafo (red bayesiana) que representa la estructura de dependencia entre las clases. Derecha: conjunto de CBS, uno por clase. Cada clasificador para la clase $C_i$ incluye como conjunto de atributos a $A_1, A_2, A_3$, además de sus padres en la estructura de clases.

puede modelar como un clasificador bayesiano en cadena, donde cada clase (binaria) indica si se administra o no cierta droga, y las mutaciones presentes en la diferentes posiciones del RNA del virus corresponden a los atributos.

Para poder desarrollar un BCC para seleccionar fármacos para el VIH, se requiere aprender tanto la estructura de dependencias entre clases, como los parámetros del modelo. Para ello utilizamos la base de datos de VIH de la Universidad de Stanford (HIVDB) [10], con datos clínicos de 2373 pacientes. Nos enfocamos inicialmente en un tipo de fármacos, los inhibidores de la *proteasa*, habiendo nueve drogas disponibles actualmente: Amprenavir (APV), Atazanavir (ATV), Darunavir (DRV), Lopinavir (LPV), Indinavir (IDV), Nelfinavir (NFV), Ritonavir (RTV), Tripanavir (TPV) and Saquinavir (SQV); estas corresponden a las clases del BCC. Como atributos seleccionamos a las mutaciones más comunes (estadísticamente) en las base de datos: L63P, I93L, V77I, I62V, L10I, E35D, L90M, M36I, A71V and R41K. Con esto aprendimos un BCC que produjo resultados prometedores para la base de datos de Stanford (utilizando un proceso de valdiación cruzada, donde parte de los datos se utilizan para entrenar el clasificador y parte para probarlo).

La respuesta a los fármacos para el VIH no sólo depende de las mutaciones del virus, sino también de la genética del paciente. Las grandes bases de datos existentes se centran en personas anglo-sajones, por lo que hay una falta de datos y estudios para otros pacientes. En colaboración con el Instituto Nacional de Enfermedades Respiratorias, estamos iniciando actualmente un estudio sobre el VIH en México y Centroamérica, para modelar la respuesta del virus en esta población a los diversos fármacos y ayudar a seleccionar el mejor tratamiento.

# 5 Clasificadores Jerárquicos

La clasifiación jerárquica es una variante de la clasificación multidimensional en la cual las clases están arregladas en una jerárquía. Esta jerarquía puede ser en forma de árbol (cada clase tiene a lo más una super-clase o padre en la jerarquía) o de grafo acíclico dirigido (una clase puede tener más de una super-clase). Hay diversas aplicaciones para las cuales la clasificación jerárquica se ha vuelto importante recientemente, como la clasificación de géneros musciales, del contenido de una página en la *Web*, en bionformática y en visión computacional, entre otras.

Se han propuesto diferentes alternativas para atacar el problema de clasificación jerárquica, que podemos dividir en dos grandes grupos: (i) métodos locales y (ii) métodos globales. Los métodos globales [13] consideran un solo clasificador, donde la variable clase incluye, normalmente, todas las clases en las hojas de la jerarquía. Esto resulta en modelos muy complejos, que no son apropiados cuando se tienen jerarquías muy grandes.

Los métodos locales [11] combinan una serie de clasificadores. Existen 3 principales esquemas para construir clasificadores locales: clasificador por nivel, clasificador por nodo, y clasificador por nodo padre. En el primero se entrena un clasificador por cada nivel de la jerarquía. En el clasificador por nodo, se entrena un clasificador binario por cada nodo en la jerarquía (excepto el nodo raíz). Para el clasificador por nodo padre, se entrena un clasificador multiclase por cada nodo (excepto las hojas), que predice sus nodos hijos. En las 3 variantes, la clasifiación de un nuevo objeto se realiza normalmente de arriba hacia abajo, partiendo de los clasificadores en la parte alta de la jerarquía hasta llegar a los nodos hojas. Una limitación importante de estos enfoques es el problema de inconsistencia, ya que si una predicción es incorrecta a cierto nivel de la jerarquía, este error se propaga a todos sus descendientes.

Nosotros proponemos un método alternativo, que se puede ver como un esquema híbrido (local-global), el cual aminora las limitaciones de los métodos anteriores.

## 5.1 Clasificador Jerárquico Multidimensional

La idea básica del clasificador jerárquico multidimensional (CJM) es aprender una serie de clasificadores locales, y luego combinar sus resultados para obtener la probabilidad de cada *trayectoria* de la raíz a cada hoja en la jerarquía (por ahora suponemos que la jerarquía tiene estructura de árbol), seleccionando aquella trayectoria de mayor probabilidad. A continuación describimos el modelo a detalle.

Asumimos que tenemos como entrada un conjunto $D$ de ejemplos con sus atributos y clase; y una jerarquía de clases, donde $pa(C_i)$ representa el padre de la clase $C_i$ en la jerarquía. El método tiene dos fases, entrenamiento y clasificación.

### 5.1.1 Entrenamiento

En la fase de entrenamiento se aprende a partir de los datos, $D$, un conjunto de clasificadores locales, uno por nodo padre. Esto es, se entrena un clasificador multi-clase $C_i$ por cada nodo no padre en la jerarquía, tal que las clases de cada clasificador comprenden a sus nodos hijos, $c_1, c_2, ...c_k$ de acuerdo a la estructura de la jerarquía. Por

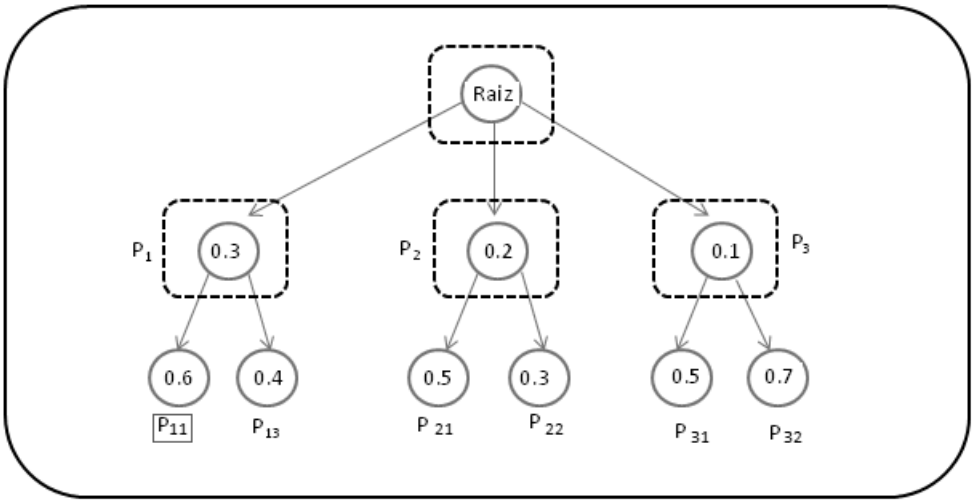


Figura 7: Ejemplo de una jerarquía con estructura de árbol y 10 clases. En la fase de entrenamiento se aprende un clasificador local por cada nodo no-hoja (cuadros punteados). En la fase de clasificación se obtiene la probabilidad de cada nodo hijo de los clasificadores locales (números adentro de los nodos); y en base a estas se obtiene la probabilidad de cada trayectoria de la raíz a las hojas, en este ejemplo hay 6 trayectorias. La de mayor probabilidad es seleccionada, en este caso $P_{11}$.

ejemplo, en la jerarquía de la figura 7, se aprende un clasificador local para cada nodo que tiene un cuadrado punteado alrededor. Los clasificadores locales pueden ser CBS o cualquier otro tipo de clasificador multi-clase.

### 5.1.2 Clasificación

La fase de clasificación consiste en calcular para un objeto representado por su vector de características, $\mathbf{X}$, la probabilidad para cada uno de los clasificadores locales en el CJM. Posteriormente se obtiene la probabilidad de cada trayectoria de la raíz a las hojas mediante una regla de combinación de las probabilidades locales. De varias posibles formas de combinación, una razonable es el producto de probabilidades. Para ellos simplemente se multiplican las probabilidades de cada clasificador local en la trayectoria, y se divide entre el número de clasificadores (considerando que puede haber trayectorias de diferente longitud). La siguiente ecuación muestra la regla de combinación:

$$R_j = \frac{\prod_{i=1}^{n} P_i}{n} \tag{10}$$

donde $n$ es el número de nodos en la trayectoria $j$, $P_i$ es la probabilidad que obtuvo cada nodo en la trayectoria, y $R_j$ es el resultado para la trayectoria $j$[1]. La figura 7 ilustra el proceso de clasificación, mostrando la probabilidad de cada clasificador local en el nodo correspondiente. La trayectoria de mayor probabilidad se indica enmarcada debajo del nodo hoja, la cual corresponde al mayor producto de las probabilidades locales.

Esta regla asume implícitamente que las clases (nodos) de la trayectoria son independientes, lo cual no se cumple necesariamente. Una forma de tomar en cuenta las dependencias, es incluyendo la clase padre (o los ascendientes) como atributos adicionales en los clasificadores locales, en forma análoga al clasificador bayesiano en cadena. Esto lo dejamos como trabajo futuro.

Si utilizamos CBS como clasificador lineal, la complejidad del CJM es básicamente lineal por classificador local, y además se tienen $nl$ multiplicaciones por cada trayectoria, donde $nl$ es el número de niveles en las jerarquía. Si asumimos una jerarquía binaria *balanceada* se tienen del orden de $C/2$ trayectorias, donde $C$ es el número de clases. Entonces habría del orden de $C/2 \times nl$ multiplicaciones, además del orden de $C/2$ CBS; lo que sigue siendo muy eficiente.

## 5.2 Clasificación de Galaxias

La clasificación automática de galaxias es importante por varias razones. Por un lado, ayuda a producir grandes catálagos a partir de las observaciones del cielo. Por otro lado, permite el descubrir la física subyacente. Hay dos formas principales para clasificar galaxias, morfológica y espectral. La morfológica parte de imágenes de las galaxias, y la clasifica en base a su forma. La espectral parte del espectro y realiza la clasificación en base a su composición estelar. En este trabajo nos enfocamos a la clasificación morfológica.

Se han propuesto diversos métodos para clasificación de galaxias [2, 4]. Sin embargo, dos problemas principales permanecen. Uno proviene del desbalance entre clases, ya que en la naturaleza tiende a haber mucho más galaxias de ciertos tipos que de otros, y esto se refleja en el número de ejemplos en los datos de entrenamiento. Por otro lado, dado que ciertas clases de galaxias son difíciles de distinguir, la efectividad de los clasificadores tiende a disminuir al aumentar el número de galaxias. Un aspecto importante, no considerado en los trabajos previos, es que las clases de galaxias forman una jerarquía. Por lo que utilizar un clasificador jerárquico puede ayudar a reducir los problemas anteriores, en particular el segundo.

Hay diferentes clases de galaxias de acuerdo a su morfología. En 1926, Edwin Hubble [5] creo una taxonomía de galaxias de acuerdo a su forma, que se ilustra en la figura 8. De acuerdo a la clasificación de Hubble existen 3 tipos principales de galaxias: espirales, elípticas y lenticulares; así como varios subtipos, como podemos observar en la figura 8. A partir del diagrama de Hubble, incorporando el tipo de galaxias irregulares y sin considerar los subtipos de galaxias elípticas, obtenemos la jerarquía de galaxias que se ilustra en la figura 9, la cual utilizamos en nuestros experimentos.

El clasificador jerárquico multidimensional se aplicó a la clasificación de galaxias

[1] Notar que el resultado no se formalmente una probabilidad, ya que no está normalizado.

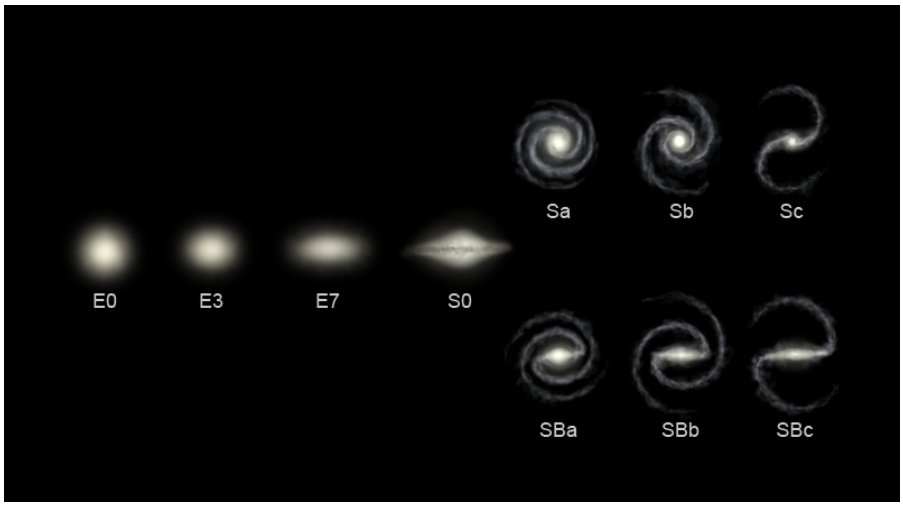


Figura 8: Diagrama de diapasón de Hubble. La figura muestra los 3 tipos principales de galaxias y sus correspondientes subtipos. E0-E7 son galaxias elípticas. Sa, Sb, and Sc son galaxias espirales normales. SBa, SBb, and SBc son galaxias espirales barradas. Finalmente, S0 son galaxias lenticulares.

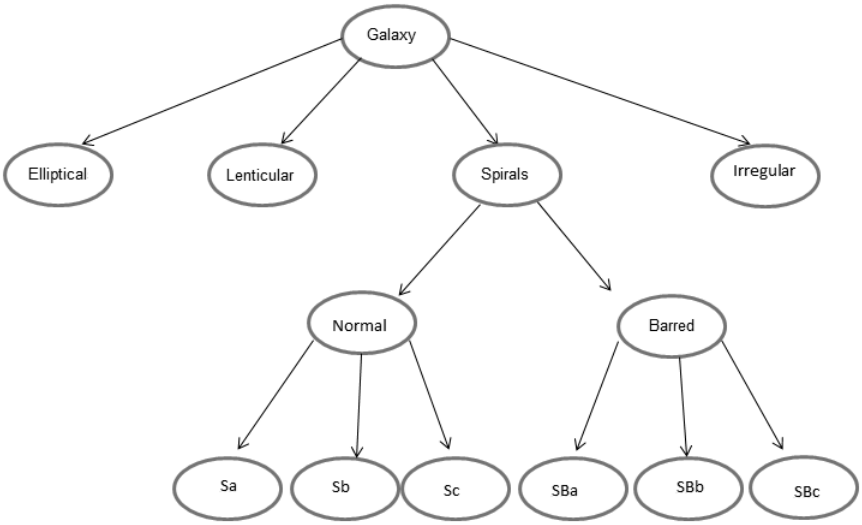


Figura 9: Jerŕquica de clases de galaxias utilizada en los experimentos (nombres de las clases en inglés).

en las placas astronómicas del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE). El INAOE tiene una gran colección de placas tomadas por una cámara Schmidt que cubren un periodo de 50 años de observaciones. La clasificiación de galaxias en estas placas plantea un reto importante ya que en cada placa hay muchas galaxias, algunas a muy baja resolución y hay ruido y artefactos en las placas. Para los experimentos se utilzaron 24 placas que contienen 152 galaxias, entre 6 y 32 de cada clase,

Antes de aplicar el clasificador, se realizó un procesamiento de las imágenes y extracción de características. Se realiza, en primer lugar, una segmentación de los objetos en las placas, que son principalmente estrellas y galaxias. Después se separan las galaxias de las estrellas, utilizando un clasificador basado en los *momentos geométricos* de cada región. Finalmente se extraen una serie de características de las regiones de galaxias basadas en las propuestas por [1]; éstas constituyen los atributos para el clasificador. Adicionalmente, para reducir el problema de desbalance en las clases, se generaron ejemplos adicionales utilizando 2 estrategias: datos repetidos por sobre-muestreo y generación de ejemplos artificiales mediante transformaciones geométricas de las imágenes.

En las pruebas se consideraron 9 clases de galaxias dentro de la jerarquía de la figura 9, así como el agregar ejemplos adicionales mediante sobre-muestreo y datos artificiales. Se comparan los resultados de usar un clasificador plano versus el clasificador jerárquico. Los resultados de las pruebas se resumen en las tabla 1.

Tabla 1: Resultados para la clasificación de 9 tipos de galaxias. Se contrasta el clasificador plano y el jerárquico utilizando sólo los datos originales, y aumentando datos (para las clases con pocos ejemplos) mediante sobre-muestreo y generación de ejemplos artificiales. La tabla muestra la precisión en cada caso en porcentaje.

| | **Clasificador plano** | **Clasificador Jerárquico** |
|---|---|---|
| Solo datos | 22.09 | 42.85 |
| Datos + sobre-muestreo | 29.99 | 42.86 |
| Datos + ejemplos artificiales + sobre-muestreo | 39.44 | **53.57** |

Observamos a partir de estos resultados que el clasificador jerárquico siempre es mejor al plano, y que el uso de sobre-muestreo y ejemplos artificiales mejoran la precisión de ambos clasificadores. Para el último caso (datos + ejemplos artificiales + sobre-muestreo), se tiene una mejora de 14 puntos con el clasificador jerárquico respecto al plano, una diferencia significativa.

# 6 Conclusiones

Un clasificador transforma datos en conceptos. El desarrollar clasificadores efectivos y eficientes es prioritario en ingeniería computacional ya que son importantes para muchas aplicaciones. Esto se ha vuelta cada vez más relevante al contar con muchos datos, por ejemplo en Internet, que hay que abstraer y transformar en conocimiento.

En este trabajo hemos desarrollado una familia de clasificadores basados en el paradigma bayesiano. El clasificador bayesiano semi-simple permite mejorar al CBS cuando se tiene atributos irrelevantes o dependientes, manteniendo una complejidad similar a la del CBS. Los clasificadores bayesianos en cadena resuelven problemas de clasificación multidimensional incorporando las relaciones entre clases y con una eficiencia parecida al del CBS. El clasificador jerárquico multidimensional resuelve problemas de clasificación cuando las clases se estructuran en una jerarquía, evitando el problema de inconsistencia. Una propiedad importante de todos estos clasificadores es que son eficientes, tanto en espacio como en tiempo de cómputo.

Existen varias avenidas de trabajo futuro. Una es extender el CBSS para incorporar atributos continuos. Otra es analizar que tanto afecta al clasificador bayesiano en cadena el que se seleccionen las clases por separado y no en conjunto; y ver si hay alguna forma de aproximar el problema de la abducción total en forma eficiente. En cuanto al clasificador jerárquico, estamos actualmente trabajando en extenderlo a jerarquías en forma de grafo acíclico dirigido e incorporando dependencias entre clases. En un sentido más amplio, un problema es como seleccionar los atributos o características que se alimentan a los clasificadores. En este sentido, planeamos explorar las técnicas de aprendizaje *profundo* para aprender estas características directamente de las señales de entrada.

## Agradecimientos

Agradezco a mis colaboradores por sus contribuciones en estos desarrollos, en particular a Eduardo Morales, Felipe Orihuela, Jesús González y Alma Ríos (INAOE), Concha Bielza y Pedro Larrañaga (UPM) y Duncan Gillies (IC). Un agradecimiento especial a mis estudiantes quienes me ayudaron mejorar y aterrizar estas ideas: Miriam Martínez, Julio Zaragoza, Julio Hernández, Pablo Hernández y Maribel Marín. Reconozco a las instituciones que han financiado los proyectos que llevaron a estos desarrollos: INAOE, CONACYT y la Unión Europea. Finalmente agradezco a mi familia, mi esposa Doris y mis hijos Edgar y Diana, por su apoyo y comprensión. Dedico este trabajo a mi padre, Fuhed Sucar (QEPD), miembro de la Academia de Ingeniería, quien fue siempre para mi ejemplo y fuente de inspiración.

# Referencias

[1] D. Bazell. Feature relevance in morphological galaxy classification. In *Mon.Not.R. Astron. Soc.*, pages 519–528, 2000.

[2] D. Bazell and David W. Aha. Ensembles of classifiers for morphological galaxy classification. *The Astrophysical Journal*, 548(1):219, 2001.

[3] C. Bielza, G. Li, and P. Larrañaga. Multi-dimensional classification with bayesian networks. *International Journal of Approximate Reasoning*, 2011.

[4] Jorge De la Calleja, Gladis Huerta, Olac Fuentes, Antonio Benitez, Eduardo López Domínguez, and Ma. Auxilio Medina. The imbalanced problem

in morphological galaxy classification. In *Proceedings of the 15th Iberoamerican congress conference on Progress in pattern recognition, image analysis, computer vision, and applications*, CIARP'10, pages 533–540, Berlin, Heidelberg, 2010. Springer-Verlag.

[5] H. Karttunen, P. Kröger, H. Oja, M. Poutanen, and K.J. Donner. *Fundamental Astronomy*. Springer-Verlag Berlin Heidelberg, 2007.

[6] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[7] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Howard, 1994.

[8] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan-Kaufmann, 1988.

[9] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, pages 254–269, Berlin, Heidelberg, 2009. Springer-Verlag.

[10] R. Shafer. Rationale and uses of a public hiv drug-resistance databases. *Journal of Infectious Diseases*, 194:S51–S58, 2006.

[11] Carlos Nascimento Silla-Jr. and Alex Alves Freitas. A survey of hierarchical classification across different application domains. *Data Min. Knowl. Discov*, 22(1-2):31–72, 2011.

[12] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

[13] Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. Decision trees for hierarchical multi-label classification. *Mach. Learn.*, 73(2):185–214, November 2008.

## Vita

El Dr. Sucar obtuvo el grado de ingeniero en electrónica y comunicaciones del ITESM, en Monterrey, en 1980: el de maestro en ciencias en ingeniería eléctrica de la Universidad de Stanford, EUA, en 1982; y el de doctor en computación por el Imperial College, Londres, G.B., en 1992. Fue investigador post-doctoral en el Departamento de Computación en el Imperial College en 1992, y ha realizado estancias de investigación en el mismo Imperial College en 1995, en la Universidad de British Columbia, Canadá en 2004, y en el INRIA, Francia en 2008.

El Dr. Sucar es Miembro del Sistema Nacional de Investigadores desde 1984, actualmente Nivel III, ha sido presidente de la Sociedad Mexicana de Inteligencia Artificial (2000-2002), es Miembro de la Academia Mexicana de Ciencias desde 2006

(México), Miembro de la Academia de Ciencias de Morelos desde 2006 (México) y *Senior Member* del IEEE (EUA) desde 2009.

Ingresó al Instituto de Investigaciones Eléctricas en 1982, donde trabajó como investigador y jefe de proyecto en el Departamento de Electrónica hasta 1993. De 1994 a 2005 fue Profesor Titular en el Departamento de Computación en el ITESM Campus Cuernavaca, habiendo sido Director del Departamento de Computación (1998-2003) y Director del Postgrado en Computación (2004-2006). Desde enero de 2006 es Investigador Titular en la Coordinación de Ciencias Computacionales del INAOE, siendo Director de Investigación de 2011 a 2013.