Universidad Nacional Autónoma de México

Facultad de Ciencias

Comparativo entre clasificadores bayesianos y redes neuronales

T E S I S

que para obtener el título de:

actuario

presenta:

américa andrea sandoval zárate

director de tesis:

dr. ramsés humberto mena chávez

A mi familia.

A mis amigos.

II

Agradecimientos

Agradezco la colaboración del Dr. Ramsés Mena Chávez, quien tuvo la paciencia necesaria para ayudarme a llevar a buen termino este trabajo. También quiero agradecer a los sinodales y en particular, al Dr. Martín Romero Martínez, quien colaboraró en la edición y revisión del presente.

A mis amigos,Vanessa, Veronica, Jacob y Brenda que observaron y colaboraron en las distintas etapas de elaboración de esta tesis. A Rafael, que ha sido parte importante de mi vida en los últimos años, gracias por el apoyo, la paciencia y la comprensión.

A mis alumnos, que de especial manera cuestionaron los avances de este trabajo.

Y de manera especial, a mi madre, que gracias a sus esmero y apoyo he podido realizar todo lo que me he propuesto.

América

Contenido

Introducción

En esta tesis se exploran dos herramientas para el modelado de datos, las redes neuronales y las redes bayesianas. Con el fin de tener los conceptos necesarios para comparar estos modelos se estudian las definiciones y algoritmos usados en redes neuronales, así como algunos conceptos de estadística bayesiana y teoría de gráficas.

En Estadística bayesiana, a partir de información *a priori* del problema se define una distribución inicial para el parámetro de interés a la que se puede incorporar infomación mediante el teorema de Bayes resultando en la distribución posterior del parámetro.

Las redes bayesianas son la representación gráfica de un modelo jerárquico que describe las relaciones de independencia condicional entre variables, se define una distribución inicial para esta gráfica y un procedimiento de aprendizaje del que se obtiene una distribución posterior. Una vez obtenida la distribución posterior se tiene la posibilidad de utilizar el modelo para la exploración de escenarios, como una red de clasificación o para tareas de predicción.

Para el propósito de este documento, interesa explorar la capacidad de clasificación de las redes neuronales y las redes bayesianas, por ello los temas se desarrollan de la siguiente manera:

En el Capítulo 1, se define la unidad básica de procesamiento de una red neuronal, la neurona, y algunas estructuras que pueden formarse a partir de ella, llegando a formar modelos tan complejos como el *perceptrón multicapa*. Se explican también los algoritmos de aprendizaje utilizados por los modelos de redes neuronales usuales.

En el Capítulo 2, se aborda el paradigma bayesiano, definiendo conceptos como distribución inicial y distribución posterior, así como el principio de actualización implícito en el teorema de Bayes.

Además, se describen algunos algoritmos usados en la generación de números aleatorios que son usados en la integración numérica de las expresiones correspondientes a los estimadores a tratar. Estos métodos son utilizados en los algoritmos de aprendizaje para redes bayesianas.

Dentro del Capítulo 3, se define el concepto de red bayesiana y se explica la parametrización de estas estructuras a través de teoría de gráficas. También, se incorpora el concepto de independencia condicional necesario para definir la *d-separación*, (Pearl,1985).

Al definirse las redes bayesianas como gráficas acíclicas dirigidas (DAG), el proceso de aprendizaje es un procedimiento jerárquico que se lleva a cabo recorriendo la gráfica de nodos padre a nodos hijo. Como resultado de este proceso se obtienen las distribuciones posteriores de cada nodo.

El modelo más simple de red bayesiana se conoce como *Naive-Bayes*, que considera los atributos de la clase como variables independientes entre sí. Otros modelos permiten formar un árbol que exprese las relaciones entre los atributos de la clase, *Naive-Bayes aumentados* (*TAN*), (Cheng,2001).

Con el fin de comparar el desempeño de los modelos descritos, se utiliza un

conjunto de datos sobre el tratamiento de rinitis alérgica para ejemplificar cada uno de los modelos. En el Capítulo 4 se describen los resultados obtenidos al usar una red neuronal, un *Naive-Bayes* y una red bayesiana general (BAN).

Los resultados que se muestran son obtenidos usando paquetes para minería de datos, del software estadístico R, la librería rattle. Para gran parte de los modelos presentados es utilizado el software Weka (Waikato Enviromental for Knowledge Analysis), desarrollado por la Universidad de Waikato, Nueva Zelanda.

Capítulo 1

Redes neuronales

Las redes neuronales son modelos matemáticos que se componen de un conjunto de funciones llamadas *neuronas*. Para facilitar su comprensión, las redes neuronales poseen una representación gráfica. Por ejemplo, la red neuronal más sencilla se compone de datos de entrada x y una neurona, que procesa los datos como $y = f(x)$.

A continuación se presentan los modelos usuales de redes neuronales. La presente exposición está basada en *An Introduction to Neural computing* (Alexander,1990), *Redes de neuronas artificiales* (Isasi Viñuela,2004), *Fundamentals of neural networks* (Faussett,1994) y *An Introduction to neural networks* (Anderson,1995).

1.1. Neuronas

En esta sección se presenta la idea general de redes neuronales como un modelo matemático basado en el procesamiento de estímulos de las neuronas biológicas.

El sistema nervioso se compone de células especializadas en el procesamiento de

estímulos denominadas neuronas que varían de forma y tamaño dependiendo del tipo de estímulos que les corresponda procesar. Las neuronas se componen por *dendritas*, que son ramificaciones de la neurona dedicadas a la recepción de estímulos. El cuerpo de la neurona se denomina *soma*, el cuál tiene una prolongación delgada llamada *axon*, cuya función es ser la línea de transmisión entre una neurona y otra.

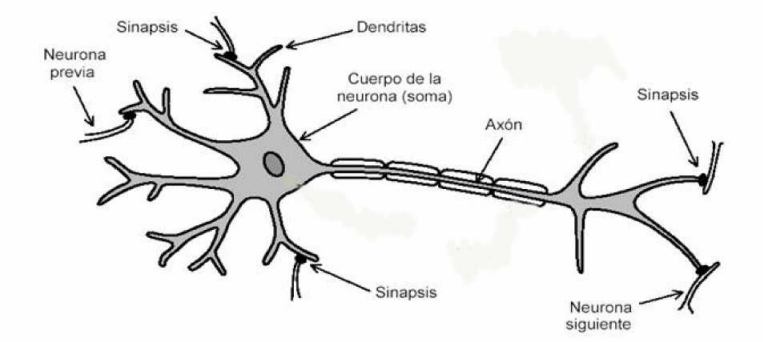

Figura 1.1: Representación de una neurona biológica.

Las neuronas son las células de mayor longitud en el cuerpo humano, cuando los axones llegan a un punto de anclaje desarrollan una arborescencia en cuyas terminales se lleva a cabo la sinapsis, que consiste en un proceso que permite a una célula afectar la actividad de las otras.

En el modelo biológico que se tiene sobre el funcionamiento de las neuronas, la recepción de estímulos se lleva a cabo por parte de las dendritas, estos estímulos provienen de otras neuronas. El soma y las dendritas procesan los estímulos recibidos y, el resultado es transmitido a lo largo del axon, hasta llegar a las ramificaciones que pasarán la información a otras neuronas mediante la sinapsis.

Las neuronas están delimitadas por una delgada membrana formada por lípidos

y proteínas cuya función es separar el interior de la neurona con el exterior. Cada neurona tiene su propia membrana que la separa de las otras.

Las estructuras formadas por las conexiones neuronales, se denominan *redes neuronales* y definen estructuras con comportamientos de alta complejidad. Tratando de imitar este comportamiento surgieron modelos llamados redes neuronales que consisten en la composición de funciones que representan relaciones secuenciales y jerárquicas.

El modelo más sencillo de red neuronal se compone de una sola neurona, este modelo se conoce como la neurona de McCulloch-Pitts.

1.2. La neurona de McCulloch-Pitts

En 1943, Warren McCulloh y Walter Pitts presentaron un modelo del funcionamiento de una neurona en su artículo *A logical calculus of the ideas immanent in nervous activity*, véase [13]. McCulloh y Pitts proponen una neurona de estados binario, cuyo estado puede modificarse de acuerdo a la intensidad de las señales provenientes de las salidas de otras neuronas. La neurona tiene definida una función que le permite cambiar de estado cuando se sobrepasa un umbral θ, esta función se denomina *función de activación*. La función de activación está definida con base a la suma de los productos de los valores de salida de las otras neuronas y sus pesos sinápticos,

$$y = \sum_{i=1}^{n} w_i \cdot x_i \tag{1.1}$$

Así, la salida de la red queda establecida por,

$$s = \begin{cases} 1 & \text{si } y > \theta \\ 0 & \text{e.o.c.} \end{cases}$$

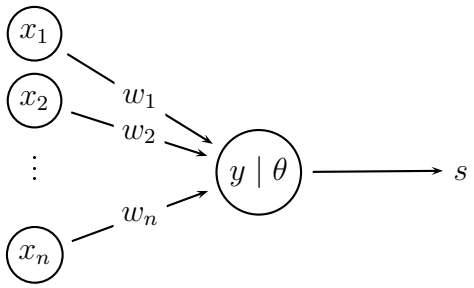

Figura 1.2: Neurona de McCulloh-Pitts.

donde $x_1, x_2, \ldots, x_n$ representan las entradas correspondientes a las señales de la sinapsis de una neurona biólogica. Cada señal se pondera considerando el peso asociado $w_i, i = 1, \ldots, n$. Tanto los valores de entrada como los pesos toman valores reales, si los valores de entrada son normalizados, entonces $0 < x_i < 1$ y $0 < \theta < 1$. La modificación de los valores de los pesos se determinan mediante una serie de reglas que llamaremos algoritmo de aprendizaje.

1.3. Algoritmos de aprendizaje

La adaptabilidad permite a los organismos reconocer patrones del entorno que tienen relevancia para su sobrevivencia. Por ello, es de interés estudiar cómo se lleva a cabo este proceso de reconocimiento, tras años de estudio se han propuesto varias reglas que en funcionamiento lo semejan. Estas reglas consideran el apendizaje para una sola neurona, aunque se extienden para toda una red, adaptando cada uno de los pesos sinápticos hasta obtener la respuesta deseada, esto se conoce como *proceso de entrenamiento*. Algunos de los procedimientos más conocidos son:

1. Regla de Hebb

2. Perceptrón

3. Regla de Widrow-Hoff

En adelante, considérese un conjunto de p vectores, que describen p objetos con ciertas características, seleccionaremos k vectores que sirvan como vectores de entrenamiento para la red; luego, los $p - k$ vectores restantes servirán como vectores de validación, es decir, ayudarán para decir qué tan bueno fue el entrenamiento de la red, mediante la comparación entre y y $\eta(x)$, donde $\eta(x)$ es la predicción de y a partir de x. El aprendizaje de una red neuronal consiste en determinar los valores de los pesos a partir del conjunto de entrenamiento. En las secciones siguientes se describen los modelos usuales de redes neuronales así como el algoritmo de aprendizaje usado para cada modelo.

1.3.1. Regla de Hebb

Se distingue por ser la regla más sencilla de aprendizaje, fue propuesta por Donald Hebb, en su libro *Organization of Behavior* (1949). Esta regla consiste en la modificación de los pesos sinápticos de manera tal que si dos neuronas están conectadas y activas, el peso entre ellas será incrementado en proporción a esta actividad.

El algoritmo de aprendizaje se lleva a cabo como sigue:

El nivel de activación de la neurona de procesamiento está dado por b_i, al inicio vale 1, y se considera como otro peso dentro de la red por lo que se toma en cuenta para el calculo de y.

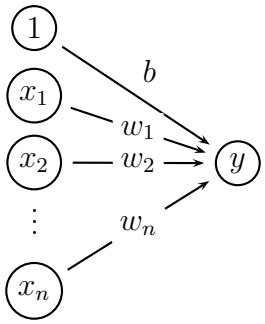

Figura 1.3: Ilustración del aprendizaje mediante la regla de Hebb.

Inicializar: los pesos, $w_i = 0$

Para todo vector de entrenamiento t y cada vector de salida s

Activar las neuronas de entrada, $x_i = t_i$

Calcular los valores para las neuronas de salida, y, como en (1.1)

Si $y > \theta$ **Entonces**

Ajustar los pesos, $w_j(x_i) = w_{j-1}(x_i) + x_i\, y$

Ajustar $b_i = b_{i-1} + y$

Fin Si

Fin Para

1.3.2. Perceptrón

Durante los 60'*s* el concepto de perceptrón fue introducido por Frank Rosenblatt (1962), Minsky y Papert. El perceptrón es una generalización de la neurona de McCulloch-Pitts, la versión original del perceptrón consta de tres capas:

1. Retina: es la capa de entrada de los datos (x).

2. Capa oculta: es la capa que procesa la información (ε).

3. Capa de salida: es la salida de la red (s).

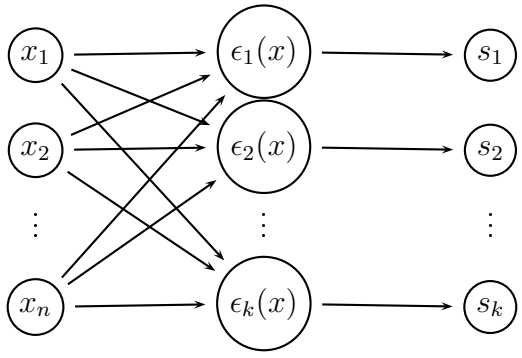

Figura 1.4: Gráfica de un perceptrón.

El perceptrón extrae información específica en cada una de sus capas, por lo que es utilizado como una herramienta para el reconocimiento de patrones. El perceptrón es utilizado en problemas linealmente separables, es decir, para distinguir entre dos posibles clases se puede trazar una línea o plano que las separe.

La función de activación para cada neurona es la función escalón con un umbral arbitrario θ. Las señales enviadas entre la capa oculta y la capa de salida son de

naturaleza binaria. La salida de las neuronas se define como una función aplicada a la contribución definida por (1.1), es decir, $s_i = f(\varepsilon)$, la función de activación queda como sigue:

$$s_i(\varepsilon) = \begin{cases} 1 & \text{si } \varepsilon > \theta \\ 0 & \text{si } -\theta \leq \varepsilon \leq \theta \\ -1 & \text{si } \varepsilon < -\theta \end{cases} \tag{1.2}$$

En el perceptrón, los pesos w_{ij} de la capa de procesamiento hacia la capa de salida son ajustados de acuerdo a una regla de aprendizaje. Para cada entrada durante el entrenamiento, la red deberá calcular una salida, esto nos ayudará a obtener el grado de error. Los pesos de la red deberán ajustarse de acuerdo a lo siguiente:

$$w_j(x_i) = w_{j-1}(x_i) + \alpha\, t\, x_i,$$

donde $t = \{-1, 1\}$ y α la tasa de aprendizaje. El término t es usado como correción de los pesos en caso de haber un error de clasificación.

El entrenamiento de la red continuará hasta que el error exhibido sea mínimo:

Cuando existe más de una capa oculta el algoritmo de entrenamiento utiliza la regla de Widrow-Hoff para la actualización de los pesos w_{ij}.

1.3.3. Regla de Widrow-Hoff

En 1960, Widrow y Hoff propusieron un sistema denominado *Adaptative Linear Neuron (Adaline)*, que consta de una sola neurona que recibe entradas de las otras n células. Consideremos el valor inicial de activación de la neurona de salida, como 1, sus salidas serán $\{-1, 1\}$.

El proceso de aprendizaje, conocido como regla delta o regla de Widrow-Hoff, consiste en encontrar los pesos que minimizan la diferencia entre el valor de salida

---

Inicializar: pesos y el valor de activación de cada neurona, de manera aleatoria, y $0 \leq \alpha \leq 1$

Mientras la condición de paro sea falsa

Para todo vector de entrenamiento

Establecer el valor del conjunto de entrada, x_i

Calcular $\varepsilon = \sum_{i=1}^{n} w_i \cdot x_i$

Aplicando la función descrita en (1.2) a cada neurona.

Actualizamos los pesos:

Si $s \neq t$ **Entonces**

$w_j(x_i) = w_{j-1}(x_i) + \alpha\, t\, x_i$

En caso contrario

$w_j(x_i) = w_{j-1}(x_i)$

Fin Si

Fin Para

Evaluar la condición de paro.

Fin Mientras

---

del patrón ingresado y el valor de salida esperado:

$$Error = \frac{1}{2}\sum_{n=1}^{N}(\eta - s(\varepsilon))^2 \tag{1.3}$$

de donde, η es la salida esperada del vector de entrenamiento, y $s(\varepsilon)$ representa la salida producida por la red.

El algoritmo de entrenamiento queda como sigue:

Inicializar: los pesos de forma aleatoria, y establecer la tasa de aprendizaje α.

Mientras la condición de paro sea falsa

Para todo vector de entrenamiento x

Establecer las activaciones de la capa de entrada, x_i

Evaluar las entradas de la red en las neuronas de salida

$\varepsilon(x) = \sum_{i=1}^{n} x_i \cdot w_i$

Actualizar pesos y el nivel de activación de la salida,

$w_j(x_i) = w_{j-1}(x_i) + \alpha\,(t - s(\varepsilon))x_i$

Fin Para

Evaluar condición de paro.

Fin Mientras

La manera en cómo se minimiza el error es utilizando un proceso iterativo en el que los patrones van pasando uno a uno por la red, y el cambio en los pesos viene dado por la regla de descenso del gradiente.

1.3.4. Perceptrón multicapa

En 1969, Minsky y Papert mostraron que la combinación de varios perceptrones simples podía ser una solución adecuada al tratamiento de problemas no lineales.

Sin embargo, no presentaron una solución a la estimación de los pesos de la capa de entrada a la capa oculta. Posteriormente, en 1986, Rumelhart, Hinton y Willian, presentaron una manera de ajustar los pesos a través de las neuronas ocultas para reducir el error de predicción, este procedimiento se conoce como regla delta generalizada.

El diseño del perceptrón multicapa consta de más de una capa oculta, donde se realizará un procesamiento no lineal de los vectores recibidos. Otra característica es que presenta conectividad total, lo que significa que todas la neuronas de la capa anterior están conectadas con todas las neuronas de la siguiente capa.

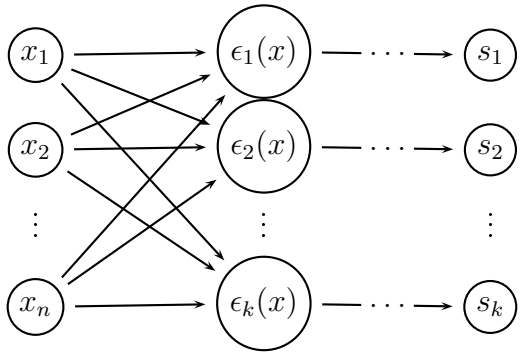

Figura 1.5: Vista del modelo perceptrón multicapa.

Sea un perceptrón multicapa con C capas, considerando la capa de entrada y la capa de salida, nos quedan $c-2$ capas ocultas, y sea n_c el número de neuronas en la capa c, $c = 1, 2, \ldots, C$. La matriz W^c, es la matriz de pesos de la capa c a la capa $c+1$; U^c, el vector de umbrales de las neuronas de la capa c, y y_i^c la activación de la neurona i en la capa c.

Para la capa de entrada, $y_i^1 = x_i$, para $i = 1, 2, \ldots, n_1$ y cada x_i una entrada

del vector de entrenamiento. Las neurona ocultas de la red procesan información aplicando la función de activación a la suma de los productos de las activaciones por sus pesos:

$$y_i^c = f\left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} y_j^{c-1} + u_i^c\right), \tag{1.4}$$

donde w_{ji}^{c-1} es el peso de la neurona j de la capa $c-1$ a la neurona i de la capa c.

Para el perceptrón multicapa, la función de activación debe cumplir con ciertas características: continua, diferenciable y monótamente decreciente, y para efectos de cómputo, su derivada debe ser fácil de calcular. Las funciones más utilizadas son la función sigmoidal, tangente hiperbólica, gausiana, entre otras.

Algoritmo de retropropagación Para el perceptrón multicapa el algoritmo de aprendizaje utilizado se formula como un problema de minimización, el de minimizar el error total de la red,

$$E = \frac{1}{N}\sum_{n=1}^{N} e(n),$$

tal que N es el número de patrones y $e(n)$ el error cometido por la red para el patrón n.

$$e(n) = \frac{1}{2}\sum_{i=1}^{n_c}(\eta_i(n) - y_i(n))^2,$$

con $\eta_i(n)$ las salidas esperadas y $y_i(n)$ las salidas dadas por la red.

Como el aprendizaje de la red debe realizarse para minimizar el error total, el método más usado, consiste en una sucesiva minimización de los errores para cada vector, $e(n)$, en lugar de minimizar el error total E. Cada parámetro de la red se modifica para cada vector de entrada n:

$$w(n) = w(n-1) - \alpha\frac{\partial e(n)}{\partial w}. \tag{1.5}$$

El término de retropropagación se utiliza debido a la manera en que se implementa el método del descenso del gradiente, pues el error cometido a la salida de la red es propagado hacia atrás, haciéndolo un error para cada una de las neuronas ocultas de la red.

1.3.5. Regla delta generalizada

La regla delta tiene como propósito evaluar la derivada parcial en (1.5). Sea un perceptrón multicapa con C capas, para describir el proceso de actualización de pesos y umbrales es necesario distinguir dos casos,

i) para los pesos y umbrales de la capa oculta $C-1$ a la capa de salida,

ii) para los demás pesos y umbrales, de la capa 1 a la capa $C-1$.

Caso i)

Sea w_{ji}^{c-1}, el peso de la neurona j de la capa oculta $c-1$ a la neurona i de la capa de salida. Usando (1.5), el cambio en el peso de la capa $c-1$ queda:

$$w_{ji}^{c-1}(n) \;=\; w_{ji}^{c-1}(n-1) \;-\; \alpha \frac{\partial e(n)}{\partial w_{ji}^{c-1}},$$

considerando que el peso w_{ji}^{c-1} sólo afecta a la neurona i, y dada la definición de $e(n)$ se obtiene,

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} \;=\; -(x_i(n) \;-\; y_i(n)) \, \frac{\partial y_i(n)}{\partial w_{ji}^{c-1}}.$$

Entonces, por (1.4),

$$\frac{\partial y_i(n)}{\partial w_{ji}^{c-1}} \;=\; f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} \, y_j^{c-1}(n) \;+\; u_i^c \right) \, y_j^{c-1}(n).$$

Ahora se define $\delta_i(n)$ como el término asociado a la neurona i de la capa de salida y al patrón n,

$$\delta_i(n) \;=\; -(x_i(n) \;-\; y_i(n))\, f'\left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1}\, y_j^{c-1}(n) \;+\; u_i^c\right).$$

Por lo anterior, $w_{ji}^{c-1}(n)$ queda como sigue,

$$w_{ji}^{c-1}(n) \;=\; w_{ji}^{c-1}(n-1) \;-\; \alpha\delta_i^c(n)\, y_j^{c-1}(n).$$

Los umbrales también se modifican, de acuerdo a la siguiente expresión

$$u_i^c(n) \;=\; u_i^c(n-1) \;+\; \alpha\delta_i^c(n).$$

Caso ii)

Considérense los pesos de la capa $c-2$ a la capa $c-1$, por (1.5) w_{kj}^{c-2} se define como

$$w_{kj}^{c-2}(n) \;=\; w_{kj}^{c-2}(n-1) \;-\; \alpha\frac{\partial e(n)}{\partial w_{kj}^{c-2}}.$$

A diferencia de w_{kj}^{c-1}, w_{kj}^{c-2} afecta a las neuronas de las capas siguientes, por tanto el cambio en los pesos de la capa $c-2$ se tomará en cuenta en la salida de la red,

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} \;=\; -\sum_{i=1}^{n_c}(x_i(n) \;-\; y_i(n))\frac{\partial y_i(n)}{\partial w_{kj}^{c-2}}.$$

Hay que tomar en cuenta que $\frac{\partial y_i(n)}{\partial w_{kj}^{c-2}}$ influye en la activación de la neurona j de la capa $c-1$ y que el reto de las activaciones de las neuronas en esta capa no dependen de éste. Por tanto,

$$\frac{\partial y_i(n)}{\partial w_{kj}^{c-2}} \;=\; f'\left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1}\, y_j^{c-1}(n) \;+\; u_i^c\right)\; w_{ji}^{c-1}\,\frac{\partial y_j^{c-1}(n)}{\partial w_{kj}^{c-2}},$$

definiendo $\delta_i^c(n)$ como en i),

$$\delta_i^c(n) \;=\; -(x_i(n) \;-\; y_i(n))\, f'\left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1}\, y_j^{c-1}(n) \;+\; u_i^c\right),$$

entonces,

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} \;=\; -\sum_{i=1}^{n_c} \delta_i^c(n) \frac{\partial y_j^{c-1}(n)}{\partial w_{kj}^{c-2}}.$$

Ahora,

$$\frac{\partial y_j^{c-1}(n)}{\partial w_{kj}^{c-2}} \;=\; f'\left(\sum_{j=1}^{n_{c-2}} w_{kj}^{c-2}\, y_k^{c-2}(n) \;+\; u_i^{c-1}\right) y_k^{c-2}(n),$$

y sea

$$\delta_j^{c-2}(n) \;=\; f'\left(\sum_{j=1}^{n_{c-2}} w_{kj}^{c-2}\, y_k^{c-2}(n) \;+\; u_i^{c-1}\right) \sum_{i=1}^{n_c} \delta_i^c(n)\, w_{ji}^{c-1}.$$

La regla de modificación de pesos se define,

$$w_{kj}^{c-2}(n) \;=\; w_{kj}^{c-2}(n-1) \;-\; \alpha \delta_j^{c-2}(n) y_k^{c-2}(n),$$

con $k \;=\; 1,\, 2, \ldots,\, n_{c-2}$ y $j \;=\; 1,\, 2, \ldots,\, n_{c-1}$. Por su parte los umbrales se adaptarán de acuerdo a lo siguiente,

$$u_j^{c+1}(n) \;=\; u_j^{c+1}(n-1) \;+\; \alpha \delta_j^{c+1}(n),$$

con $j \;=\; 1,\, 2, \ldots,\, n_{c+1}$ y $c \;=\; 1,\, 2, \ldots,\, c-2$.

1.3.6. Redes neuronales probabilistas

Las redes neuronales probabilistas son modelos utilizados principalmente para la clasificación de patrones y se construyen usando probabilidad clásica. Por ejemplo, supongamos que el problema es clasificar los vectores $\{x_1, x_2, \ldots, x_n\}$ en k clases

usando una función que minimice el costo del error de clasificación. Una regla de decisión para clasificar un vector de entrada en la clase A es la siguiente:

$$h_A c_A f_A > h_B c_B f_B,$$

donde h_A es la distribución inicial de la clase A, c_A es la función de pérdida asociada a clasificar erróneamente un vector de la clase A como un elemento de la clase B, y f_A es la función de densidad asociada a la clase A.

Para estimar f_A y f_B a partir de los vectores de entrenamiento, se usa:

$$f_A(x) = \frac{1}{(2\pi)^{n/2}\sigma^n}\frac{1}{m_A}\sum_{i=1}^{m_A}\exp\left\{-\frac{(X - X_{A_i})^T(X - X_{A_i})}{2\sigma^2}\right\},$$

de donde X_{A_i} es el i-ésimo vector de entrenamiento de la clase A, n es la dimensión de los vectores, y m_A es el número de vectores de entrenamiento.

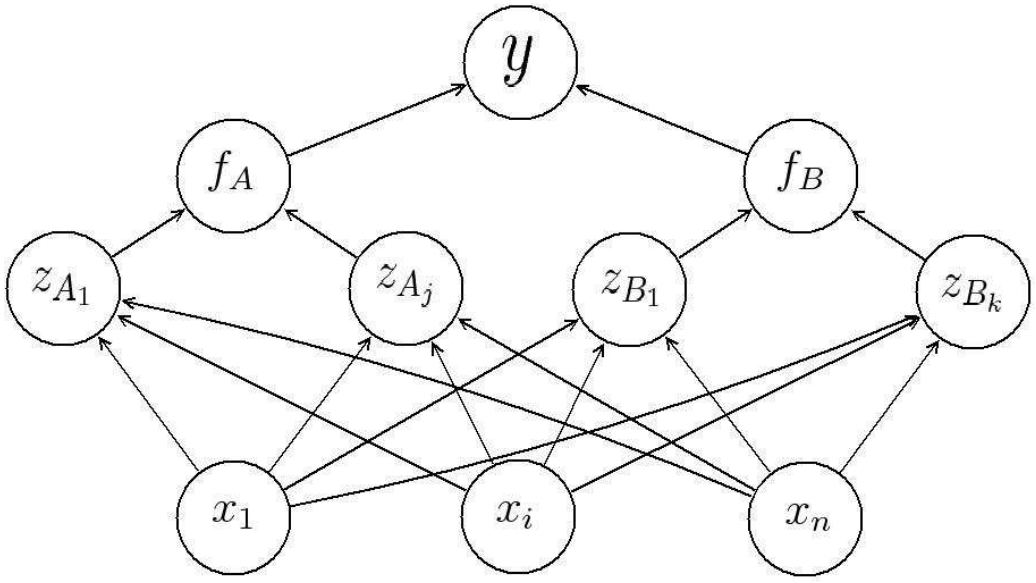

Figura 1.6: Representación de una red neuronal probabilista.

Estas redes están compuestas por cuatro capas, la de entrada (x_i), la de patrones (z_{A_j},z_{B_k}), la de unidades de procesameinto (resumen de clasificación, f_A y f_B) y la de salida (y). A través de la capa de entrada se introducen los vectores los vectores de entrenamiento; en la capa de patrones, hay una neurona por cada atributo de cada clase cuya salida se identifica con alguna unidad de procesamiento (f_A y f_B en la gráfica). Finalmente, para ubicar al vector como perteneciente a una clase se observa la salida de la última capa, y, que indica a qué clase pertenece el vector ingresado.

1.4. Aplicaciones

Las aplicaciones de las redes neuronales son tan diversas debido a que en comportamiento tratan de imitar el funcionamiento de los sistemas biológicos, y éstos realizan tareas sumamente diversas y complejas. Por ejemplo, mediante un proceso de aprendizaje un sistema biológico puede reconocer e imitar sonidos, al tiempo en que les asocia un significado; realiza la identificación de imágenes y las relaciona con conceptos.

El aprendizaje realizado por cualquier organismo le permite identificar patrones que le serán de utilidad para su adecuado desenvolvimiento en el medio. Como menciona Masumi Ishikawa,

> *La extracción de las reglas que determinan el comportamiento de los datos es importante porque es la llave a la solución de la adquisición del conocimiento.*

Las redes neuronales han incursionado en campos como la Estadística, en la estimación de series de tiempo, principalmente índices financieros, y en la clasificación

de patrones; el reconociento de secuencias de ADN, ha resultado de utilidad en Genética; la representación de modelos climáticos, ha ayudado a los meteorólogos a hacer predicciones de los cambios climáticos que podrían darse en los próximos años.

Las redes neuronales son usadas en la mejora en la identificación y clasificación de patrones, se habla de validación cruzada como una herramienta estadística para la validación de un modelo. La contribución de los patrones de entrenamiento mediante el algoritmo de retropropagación es estimada y esa información es usada para seleccionar los patrones que serán de utilidad para la validación cruzada (Friendrich Leisch, Kurt Hornik & Lakhmi C. Jain, *NNclassifiers: Reducing the computational cost of cross-validation by active pattern selection*).

El manejo de información es un tema que ha cobrado relevancia en los últimos años, sobretodo con el desarrollo de los sistemas de bases de datos (DBMS). Frecuentemente, se establecen relaciones entre la información, y para ello ésta debe ser precisa; la idea de Janusz Kacprzyk (*Fuzzy Logic in DBMS's and querying*), es incorporar información imprecisa a las relaciones, en principio, Zvielv y Chen (1986) modificaron el modelo entidad-relación de manera tal, que en las consultas se advierte una mayor fexibilidad, como la distinción cualitativa entre respuestas y la posibilidad de usar condiciones difusas en éstas.

Para más ejemplos del uso de redes neuronales y redes neuronales probabilísticas se puede consultar [16].

Capítulo 2

Distribuciones iniciales

En este capítulo se abordan los conceptos que enmarcan el paradigma bayesiano. El paradigma bayesiano consta de 3 etapas, la primera consiste en la colecta de una muestra X en una población; la segunda, se basa en la modelación de una distribución de probabilidad dada por la información de la muestra; finalmente, se lleva a cabo un proceso de actualización de la distribución de probabilidad mediante el Teorema de Bayes. Para mayor detalle ver *The Bayesian Choice* (Robert,1994).

En este capítulo también se presentan algoritmos usados frecuentemente en estadística bayesiana que basan su funcionamiento en la definición de cadenas de Markov. Para mayor profundidad en el tema veáse *Monte Carlo Statistical Methods* (Robert,2004), *Markov Chain Monte Carlo in Practice* (Wilks,1996).

2.1. Teorema de Bayes

Mediante la definición de probabilidad condicional y el teorema de Bayes podemos comprender la relación existente entre las observaciones y los parámetros. Si A y E son eventos, el teorema de Bayes se considera como el principio de actualización de $p(A)$ a $p(A \mid E)$, donde E es el resultado de un experimento que ha sido observado.

Teorema 2.1 ***Teorema de Bayes*** *Sea* $\{E_1, E_2, \ldots, E_n\}$ *una colección de eventos mutuamente excluyentes y exhaustivos, cada uno de ellos con probabilidad distinta de cero. Sea* B *un evento de manera que* $p(B \mid E_i)$ *es conocida. Entonces* $p(E_i \mid B)$ *queda dada por*

$$p(E_i \mid B) = \frac{p(B \mid E_i)p(E_i)}{\sum_{i=1}^{n} p(B \mid E_i)p(E_i)}.$$

Bayes y Laplace consideraron que la incertidumbre sobre el parámetro θ puede ser modelada mediante una distribución de probabilidad π, llamada distribución *a priori* o distribución inicial. El Teorema de Bayes es utilizado para definir la distribución que condicionada a la información aportada por la muestra X servirá para realizar inferencia. Esta distribución se denomina distribución *a posteriori* o posterior, $\pi(\theta \mid x)$, y se define de la siguiente manera

$$\pi(\theta \mid x) = \frac{f(x \mid \theta)\pi(\theta)}{\int f(x \mid \theta)\pi(\theta)d\theta}. \tag{2.1}$$

Es decir, la distribución posterior es proporcional a la verosimilitud dada una distribución inicial, esto se interpreta como la actualización de la distribución inicial utilizando la información disponible en la muestra X sobre el parámetro θ.

2.2. Distribuciones iniciales

Para la aplicación del teorema de Bayes es necesario definir una distribución inicial. La distribución inical π es una función que resume toda la información inicial disponible sobre el parámetro. Dos técnicas comúnmente utilizadas son las distribuciones iniciales conjugadas y las distribuciones no informativas.

2.2.1. Distribuciones conjugadas

Una técnica usada frecuentemente para la construcción de distribuciones iniciales continuas consiste en la elección arbitraria de una π dentro de las familias de densidades conocidas en donde los parámetros pueden ser determinados por información previa.

Un caso particular de distribuciones iniciales son las distribuciones conjugadas que se caracterizan porque la distribución incial y la distribución posterior pertenecen a la misma familia paramétrica.

Definición 2.2 *Una familia $\mathcal{F}$ de distribuciones de probabilidad sobre Θ se dice conjugada para una función de la muestra $f(x \mid \theta)$ si, para cada $\pi \in \mathcal{F}$, la distribución posterior $\pi(\theta \mid x)$ también pertenece a $\mathcal{F}$.*

Si la distribución inicial pertenece a $\mathcal{F}$, entonces, para cualquier tamaño de muestra n y cualquier valor de las observaciones, la distribución posterior deberá pertenecer a la misma familia. Por ejemplo, consideremos que la distribución inicial $\pi(\theta)$ es una distribución Gamma $\mathcal{G}(\alpha, \beta)$ y que la distribución muestral $f(x \mid \theta)$ es una distribución Poisson $\theta^{\sum_i x_i} \exp^{-n\theta}$. Luego, la distribución posterior es una distribución Gamma $\mathcal{G}(\alpha + n\bar{x}, \beta + n)$:

$$\begin{aligned}\pi(\theta \mid x) \propto & \quad \frac{\beta}{\Gamma(\alpha)}(\beta\theta)^{\alpha-1}\exp^{-\beta\theta}\theta^{\sum_i x_i}\exp^{-n\theta} \\ \propto & \quad \theta^{\alpha+\sum_i x_i-1}\exp^{-(\beta+n)\theta}\end{aligned}$$

Entonces, la distribución inicial Gamma es conjugada con la distribución muestral Poisson.

2.2.2. Distribuciones iniciales con máxima entropía

Una de las maneras para definir una distribución inicial es mediante el concepto de entropía desarrollado por Jaynes y Shannon. La entropía se define como una medida de información contenida en un conjunto de elementos. Para una distribución de probabilidad discreta la entropía se define como

$$\varepsilon(\pi) = -\sum_i \pi(\theta_i)\log(\pi(\theta_i)). \tag{2.2}$$

donde π es la distribución inicial para el parámetro θ. Esta cantidad había sido introducida por Shanon(1945) como una medida de la información de la distribución, usalmente utilizada en teoría de la información y procesamiento de imágenes.

Se dice que π es una distribución inicial con máxima entropía si de las distribuciones propuestas es la que tiene mayor entropía, es decir, la que aporta menor información sobre la distribución de los elementos. Sea π_0 un modelo alternativo a π, la diferencia de entropías se define como un valor esperado con respecto a la distribución de referencia π_0

$$\varepsilon(\pi) = E_{\pi_0}\left[log\left(\frac{\pi(\theta)}{\pi_0(\theta)}\right)\right].$$

Esta cantidad se interpreta como una medida de discrepancia entre la distribución π y la distribución π_0.

2.2.3. Distribuciones no informativas

Otra forma para especificar distribuciones iniciales es utilizar una distribución no informativa, llamadas así debido a la falta de información inicial. Las distribuciones no informativas tienen la restricción de que la distribución propuesta no debe asignar mayor probabilidad a ninguno de los posibles valores del parámetro θ.

Inicial de Laplace Dada la falta de información, se caracterizan los valores del parámetro mediante una distribución inicial uniforme. Esta distribución asigna igual probabilidad a cada uno de los valores en el espacio paramétrico, al no haber información sobre el parámetro no hay razones para asignar una mayor probabilidad a un valor del parámetro.

Inicial de Jeffreys Si no se conoce mayor información sobre el parámetro θ, entonces el conocimiento de θ dada una muestra X debe ser el mismo para alguna función de θ. En este caso, la distribución propuesta por Jeffreys se basa en la función de información de Fisher,

$$I(\theta) = E_{\Theta}\left[\left(\frac{\partial \log f(x \mid \theta)}{\partial \theta}\right)^2\right].$$

La distribución de Jeffreys queda como sigue:

$$\pi^*(\theta) \propto [\det(I(\theta))]^{\frac{1}{2}}.$$

Esta función de información se ve incluída en la definición de otras funciones de información como el *criterio de información bayesiana*, BIC o el *criterio de información de Akaike*, AIC.

Las distribuciones no informativas aportan la mínima información sobre el parámetro por lo que son distribuciones con máxima entropía.

2.3. Estimación bayesiana

Cuando se cuenta con la distribución inicial $\pi(\theta)$ la inferencia sobre θ queda dada por la distribución posterior $\pi(\theta \mid x)$. Cuando se quiere estimar mediante una cantidad de interés $h(\theta)$ un estimador de $h(\theta)$ es la media posterior $E_\pi[h(\theta) \mid x]$. En el caso general, se cuantifica el error de estimar θ mediante una función de pérdida, $L(\theta, \delta)$, véase [7], donde δ es la regla de decisión y θ el parámetro de interés, dada una distribución inicial π el estimador es solución de

$$\min_{\delta} E_\pi[L(\theta, \delta) \mid x]. \tag{2.3}$$

Otros métodos usados para la estimación son máxima verosimilitud (ML) y máxima distribución posterior (MAP)[14]. El método MAP busca un valor para θ que maximice la distribución posteior $\pi(\theta \mid x)$. Este método permite incorporar evidencia a la estimación. En la mayoría de los casos las expresiones obtenidas por ambos métodos de estimación no son tratables analíticamente por lo que la búsqueda de soluciones llama al uso de métodos numéricos.

Cuando se tiene la distribución posterior $\pi(\theta \mid x)$ es posible evaluar la precisión de la estimación, como el error cuadrático posterior

$$E_\pi[(\delta_\pi(x) - h(\theta))^2 \mid x],$$

donde $\delta_\pi(x)$ es el etimador de $h(\theta)$.

Otro de los usos de la inferencia bayesiana es en problemas de predicción. Si x se distribuye $\pi(x \mid \theta)$ y z es una observación, se puede evaluar la probabilidad de que la observación z provenga de $\pi(x \mid \theta)$ como $\pi(z \mid \theta, x)$.

Una vez establecidos los conceptos básicos para la estimación bayesiana se describen técnicas usuales empleadas en la aplicación de estadística bayesiana.

2.4. Integración Monte Carlo

En muchos casos la obtención de los estimadores bayesianos requiere de la integración de expresiones altamente complejas, por lo que el uso de métodos numéricos se hace necesario. En general, el estimador bayesiano bajo una función $h(\theta)$ y una distribución inicial π es la solución de (2.3), véase [21]:

$$E_\pi[h(\theta)] = \int_\Theta h(\theta)\pi(\theta \mid x)d\theta \tag{2.4}$$

donde el problema a resolver es la evaluación de la integral, para ello se utiliza una muestra $\{\theta_1, \theta_2, \ldots, \theta_m\}$ generada por la distribución f para aproximar a (2.4) mediante un promedio ergódico,

$$\bar{h}_m = \frac{1}{m}\sum_{i=1}^{m} h(\theta_i), \tag{2.5}$$

$\bar{h}_m$ converge casi seguramente a $E_f[h(\theta)]$ por la Ley Fuerte de los Grandes Números. Más aun, cuando h^2 tiene esperanza finita bajo f. Otro método para evaluar la integral es el muestreo por importancia.

Definición 2.3 *El método de muestreo por importancia es una evaluación de (2.4) basado en la generación de una muestra* $\{\theta_1, \theta_2, \ldots, \theta_n\}$ *dada una distribución ins-*

trumental g y aproximando,

$$E_f[h(\theta)] \approx \frac{1}{m}\sum_{j=1}^{m}\frac{f(\theta_j)}{g(\theta_j)}h(\theta_j). \tag{2.6}$$

Este método se basa en una representación alternativa de (2.4):

$$E_f[h(\theta)] = \int_{\Theta} h(\theta)\frac{f(\theta)}{g(\theta)}g(\theta)d\theta$$

La función instrumental g tiene pocas restricciones y puede ser elegida como una distribución fácil de simular. La muestra $\{\theta_1, \theta_2, \ldots, \theta_n\}$ puede ser usada repetidamente con diferentes funciones y para distintas distribuciones de f. Si la distribución g tiene colas más ligeras que f, no se considera una distribución adecuada debido a que la varianza para algunas funciones h no será finita.

La evaluación de integrales se basa en la generación de una muestra $\{\theta_1, \theta_2, \ldots, \theta_n\}$ proveniente de una distribución f. Para la generación de la muestra se utilizan métodos Monte Carlo vía cadenas de Markov.

2.5. Métodos de Monte Carlo vía Cadenas de Markov

Tanto el enfoque bayesiano como el frecuentista requieren de la integración numérica de los estimadores para realizar inferencia sobre los parámetros del modelo, uno de los métodos utilizados consiste en integrar una función mediante el método de Monte Carlo usando cadenas de Markov, es decir, usar una muestra $\{\theta_1, \theta_2, \ldots, \theta_n\}$ proveniente de una función f. Existen varias maneras para construir estas muestras, pero la mayoría son casos particulares de Metropolis y Hastings [22].

Para la evaluación de (2.4) se utilizan métodos numéricos, los métodos más usados se explican a continuación.

2.5.1. Métodos de Monte Carlo vía cadenas de Markov

El método de Monte Carlo evalúa (2.4), mediante muestras $\{x_i : i = 1, 2, \ldots, n\}$ de la distribución $\pi(\cdot)$, por medio de (2.5).

Hay que notar que $\{x_i\}$ no necesariamente independientes. Las muestras $\{x_i\}$ pueden ser generadas por algún proceso que arroje muestras con soporte en $\pi(\cdot)$ una cadena de Markov hasta M de manera que $\pi(\cdot)$ se considera una distribución estacionaria.

Definición 2.4 *Un método de Monte Carlo vía cadenas de Markov (MCMC) para la simulación de una distribución f es un método que produce una cadena de Markov ergódica $\{X_n\}$ cuya función estacionaria es f.*

Para un valor inicial arbitrario x_0, una cadena $\{X_n\}$ es generada usando la función de transición con función estacionaria f. El uso de una cadena $\{X_n\}$ producida por un MCMC es idéntico a usar una muestra independiente e idénticamente distribuida generada por f.

2.5.2. Cadenas de Markov

Supongamos que se genera una secuencia de variables aleatorias $\{X_0, X_1, \ldots\}$, tal que para cada $t \in \{0, 1, 2, \ldots\}$, el siguiente estado X_{t+1} depende únicamente del estado anterior X_t y no de toda la cadena, de forma que $p(X_{t+1} \mid X_t)$ sólo dependerá del estado X_t. Esta secuencia se denomina cadena de Markov, y $p(\cdot \mid \cdot)$ se llama función

de transición, que indica la probabilidad de transición del estado X_t al estado X_{t+1}. Se dice que una cadena es homogénea en el tiempo si $p(\cdot \mid \cdot)$ no depende de t.

Bajo ciertas condiciones de regularidad, la cadena olvida su estado inicial y $p^m(\cdot \mid X_0)$ eventualmente convergerá a una distribución estacionaria o invariante $(\phi(\cdot))$ que no dependerá ni del valor inicial ni del tiempo. *Veáse (Apéndice 1).*

2.5.3. Metropolis-Hastings

Este algoritmo comienza definiendo la distribución objetivo f y una densidad instrumental $q(y \mid x)$ definida con respecto a la medida dominante.

Este algoritmo produce una cadena de Markov $\{X_n\}$ donde el estado X_{t+1} es elegido por un primer muestreo a un punto Y proveniente de la distribución instrumental $q(\cdot \mid X_t)$, nótese que la distribución propuesta sólo depende de X_t. El punto Y es aceptado con probabilidad $p(X_t, Y)$,

$$p(X, Y) = \text{mín}\left(1, \frac{\pi(Y)q(X \mid Y)}{\pi(X)q(Y \mid X)}\right). \tag{2.7}$$

Si el punto Y es aceptado, el siguiente estado X_{t+1} toma el valor de Y, y en caso de que sea rechazado, la cadena no se mueve, es decir, $X_{t+1} = X_t$.

Cuando se calcula la media como en (2.5), los valores Y_t pueden ser asociados con pesos del estilo m_t/T, donde m_t es el número de veces que los valores fueron rechazados antes de aceptar Y_t.

Metropolis-Hastings es un algoritmo genérico definido para todas las f y g, sin embargo es necesario pedirles mínimas condiciones de regularidad. El soporte de f debe estar incluido en el soporte de q.

Teorema 2.5 *Sea $\{X_n\}$ la cadena producida por este algoritmo. Cada distribución condicional q cuyo soporte incluye al de f,*

1. *la función de transición de la cadena satisface $f(x)q(x \mid y) = f(y)q(y \mid x)$.*

2. *f es la distribución estacionaria de la cadena.*

Por construcción, la cadena tiene una distribución invariante, f, si también es una cadena aperiódica y Harris recurrente, entonces el teorema ergódico aplica para establecer el resultado de convergencia para (2.5).

A continuación se describen casos particulares del algoritmo Metropolis-Hastings.

2.5.4. Algoritmo Metropolis

Este algoritmo sólo considera distribuciones simétricas, de forma que $q(X \mid Y) = q(Y \mid X)$ para toda X y Y. Resulta conveniente elegir distribuciones que generen cada componente de Y condicionalmente independiente dado X_t. Por lo anterior, (2.7) queda

$$p(X, Y) = \text{mín}\left\{1, \frac{\pi(Y)}{\pi(X)}\right\},$$

un caso especial del algoritmo son las caminatas aleatorias, para las cuales $q(Y \mid X) = q(|X - Y|)$.

2.5.5. Muestreo independiente

Es un algoritmo Metropolis-Hastings que propone $q(Y \mid X) = q(Y)$, es decir, la función instrumental q no depende de X_t, para esta versión, la probabilidad de acep-

tación se escribe como

$$p(X, Y) = \text{mín}\left\{1, \frac{\pi(Y)/q(Y)}{\pi(X)/q(X)}\right\}.$$

Para que este algoritmo funcione adecuadamente, $q(\cdot)$ debe ser una buena aproximación a $\pi(\cdot)$, pero es más seguro cuando $q(\cdot)$ tiene colas más pesadas.

2.5.6. Muestreo de Gibbs

Es un caso particular del de componentes individuales de Metropolis-Hastings. La principal diferencia radica en la propuesta de distribución a utilizar para simular $Y_{\cdot i}$:

$$q_i(Y_{\cdot i} \mid X_{\cdot i}, X_{\cdot -i}) = \pi(Y_{\cdot i} \mid X_{\cdot -i}).$$

Los puntos $Y_{\cdot i}$ propuestos siempre son aceptados pues $p(X_{\cdot -i}, X_{\cdot i}, Y_{\cdot i})$ será 1.

Los métodos presentados tienen la finalidad de proveer de una visión general de los métodos utilizados usualmente para la generación de variables aleatorias, todo con la finalidad de acercarse a la distribución de un conjunto de datos dado mediante distribuciones propuestas.

2.6. Optimización Monte Carlo

Como se observa, el problema consiste en resolver (2.4), para ello se utilizan métodos para generar variables aleatorias. El problema expuesto en (2.4) se resume en la siguiente expresión:

$$\max_{\theta \in \Theta} h(\theta). \tag{2.8}$$

Se distinguen dos usos para el proceso de generación de variables aleatorias. El primero, es en las técnicas de exploración estocástica que consisten en la búsqueda

del máximo (mínimo) de una función usando descenso del gradiente. El segundo uso se basa en la aproximación probabilista a la función objetivo h, uno de los algoritmos más conocidos es el EM (*expectation-maximization algorithm*), véase [21].

Exploración estocástica

Una primera forma de resolver (2.8) es simular una distribución uniforme sobre Θ, y usar la aproximación $h_m^* = \text{máx}(h(u_1), \ldots, h(u_m))$, donde $u_i \sim U_\Theta$. Este método converge cuando m tiende a infinito, aunque no toma en cuenta alguna propiedad específica de h. En este método cuando el calculo del estimador $h(\cdot)$ es costoso el número de evaluaciones de la función h se mantiene al mínimo. Este hecho lleva a reconocer ciertas propiedades de h, si es una función positiva y si $\int_\Theta h(\theta)d\theta < +\infty$, la solución de (2.8) se reduce a encontrar las modas de la distribución h.

Método del gradiente

Es un método numérico que produce una secuencia $\{\theta_j\}$ que converge a la solución exacta de (2.8), θ^*, cuando el dominio $\Theta \subset \mathbf{R}^d$ y la función es convexa. La secuencia θ_j se construye de manera recursiva mediante la siguiente expresión,

$$\theta_{j+1} = \theta_j + p_j \bigtriangledown h(\theta_j), p_j > 0, \tag{2.9}$$

donde $\bigtriangledown_h$ es el gradiente de h.

Simulated Annealing

Este algoritmo fue introducido por Metropolis en 1953 al minimizar una función en un conjunto finito, pero también se aplica en el proceso de optimización sobre un

conjunto continuo.

La idea fundamental en este método es que un cambio de escala, llamado temperatura, permite movimientos más rápidos sobre la superficie de la función h a maximizar, que es llamada energía. Reescalando parcialmente permite la atracción a un máximo local, dada una temperatura $T > 0$, una muestra $\theta_1^T, \theta_2^T, \ldots$ generada de la distribución $\pi(\theta)$. Como T decrece hasta casi ser cero, los valores simulados de la distribución se encuentran concentrados en una vecindad del máximo local.

Comenzando en θ_0, X es generado de una distribución uniforme sobre una vecindad de θ_0, de manera más general X puede generarse a partir de una distribución $g(|X - \theta_0|)$, y el nuevo valor de θ es generado como sigue,

$$\theta = \begin{cases} X & \text{con probabilidad} p = \exp(\triangle h/T) \wedge 1 \\ \theta_0 & \text{con probabilidad} 1 - p \end{cases}$$

donde $\triangle h = h(X) - h(\theta_0)$, X es aceptado con probabilidad 1, lo que significa que θ_0 siempre cambia su valor. Por otro lado, si $h(X) < h(\theta)$, X es aceptado con probabilidad distinta de cero. Esta propiedad permite al algoritmo escapar de la atracción de θ_0 si θ_0 fuera un máximo local de h, con una probabilidad que depende de la elección de T, comparado con el rango de la densidad g.

Este algoritmo modifica la temperatura con cada iteración:

Inicializar: Simular X de una distribución instrumental con densidad $g(|X - \theta_0|)$,

Aceptar $\theta_{i+1} = X$ con probabilidad $p_i = \exp(\triangle h_i/T_i) \wedge 1$ en otro caso $\theta_{i+1} = \theta_i$

Actualizar $T_i a T_{i+1}$

Una importante característica de este algoritmo es que existen resultados de convergencia para espacios finitos. Las siguientes consideraciones se toman en cuenta

para el descenso de la temperatura en el algoritmo descrito:

Definición 2.6 *Dado un espacio de estados finito ε y una función h a maximizar:*

1. *El estado $e_j \in \varepsilon$ puede ser alcanzado, a una altura $\underline{h}$, desde un estado e_i, si existe una secuencias de estados $(e_i, \ldots, e_n)$ que unen e_i con e_j, tal que $h(e_k) \geq \underline{h}$ para $k = 1, 2, \ldots, n$.*

2. *La altura de un máximo e_i es el valor mayor de d_i tal que existe un estado e_j tal que $h(e_j) > h(e_i)$, el cual puede ser alcanzado a una altuta $h(e_j) + d_i$ desde e_i.*

Los métodos descritos serán utilizados para realizar el proceso de estimación bayesiano que, una vez definida una distribución inicial, consiste en la actualización de la distribución inicial a partir de la información aportada por la muestra.

Capítulo 3

Redes bayesianas

Las redes bayesianas son modelos probabilistas compuestas por unidades llanadas nodos que corresponden a cada una de las variables del problema. Por ejemplo, la figura (3.1) modela la relación entre edad y rinitis alérgica:

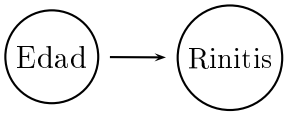

Figura 3.1: Red que representa la relación entre las variables edad y padecer rinitis alérgica.

La flecha en la figura (3.1) indica una relación de dependencia condicional entre la variable edad y el desarrollo de rinitis alérgica, es decir, la distribución de la variable Rinitis depende del valor que tome la variable Edad.

Uno de los usos de las redes bayesiabas es la modelación de relaciones de causalidad ya que, de acuerdo con Wermuth y Lauritzen [15], los supuestos de independencia condicional son el elemento básico para la descripción del conocimiento. El concepto

de causalidad es un concepto complejo que Judea Pearl trata en su libro *Causality*, sin embargo, para propósitos de esta tesis, las redes bayesianas serán utilizadas como modelos de clasificación, es decir, modelos que predicen la pertenencia de un sujeto a una clase.

Algunos de los textos usados para desarrollar este capítulo son *Probabilistic Networks and Expert Systems* (Cowell,1999), *Pattern Recognition and Neural Networks* (Ripley,1996) y *Bayesian Network Classifiers* (Friedman,1997). Otro texto utilizado es *A constrait-propagation to probabilistic reasoning* (Pearl,1985).

3.1. Preliminares

Los problemas de modelación que se enfrentan frecuentemente requieren la modelación de las asociaciones entre causas y consecuencias. Describir cómo se establecen estas relaciones resulta difícil cuando el número de variables a analizar es grande. Para resolver este problema, se han desarrollado modelos que ayudan simplificar la especificación de las relaciones establecidas. Una de estas herramientas son las redes probabilistas, que consisten en la representación gráfica de las proposiciones de independencia condicional.

3.1.1. Independencia condicional

Supongamos una gráfica con tres nodos que representan variables aleatorias, X, Y y Z, de manera que los nodos están comunicados y no forman ciclos, la figura (3.2) muestra cómo puede darse la comunicación entre los nodos.

Cada una de las gráficas representan diferente relaciones de jerárquia que se

traducen en relaciones de dependencia, por ejemplo, en la primera figura, X y Z son nodos hijo de Y, es decir, X y Z dependen de Y. La figura central muestra la dependencia de Z hacia Y y éste a su vez depende de X, por lo que Z depende indirectamente de X a través de Y. En la figura de la derecha se observa a los nodos X y Z con un nodo hijo común Y, o sea, dos variables que tienen un efecto en conjunto. A pesar de la diferente interpretación de las gráficas, la comunicación entre X y Z se lleva a cabo a través de Y, se manera que Z es condicionalmente independiente de X dado Y.

Definición 3.1 *De la figura (3.2) decimos que Z es condicionalmente independiente a X dado Y, $Z \perp X \mid Y$, si para cada pareja (y, x) de (Y, X), tenemos que $p(Z \mid Y = y, X = x) = p(Z \mid Y = y)$.*

De la definición de independencia condicional se derivan propiedades útiles para la parametrización de una gráfica, las propiedades son las siguientes:

1. Simetría, si $X \perp Y \mid Z$ entonces $Y \perp X \mid Z$.

2. Descomposición, supongamos W una variable aleatoria, si $X \perp Y\,W \mid Z$ entonces $X \perp Y \mid Z$.

3. Unión débil, si $X \perp Y\,W \mid Z$ entonces $X \perp Y \mid (Z, W)$.

Figura 3.2: Posibles formas de comunicación entre nodos.

4. Contracción, si $X \perp Y \mid Z$ y $X \perp W \mid (Z, Y)$ entonces $X \perp (Y, W) \mid Z$.

5. Intersección, si $X \perp W \mid (Z, Y)$ y $X \perp Y \mid (Z, W)$ entonces $X \perp (Y, W) \mid Z$.

Estas propiedades tienen interpretación desde el punto de vista de la información que representa cada una de las variables, (Paz,1987), les llaman axiomas sobre gráficas. El axioma de simetría establece que para cualquier estado de Z, Y no proporciona información de X y viceversa. El axioma de descomposición manifiesta que si los valores conjuntos de dos variables son irrelevantes, entonces por separado también lo son. El axioma de unión débil indica que si W y Y contienen información irrelevante para X el proceso de actualización sobre W no hace que la información de Y sea relevante para X. El axioma de contracción manifiesta que si W es irrelevante para X después de haber recibido información irrelevante de Y, entonces W ya era irrelevante para X antes de recibir información de Y.

Consideremos una gráfica no dirigida $\mathcal{G} = (V, E)$ y una distribución de probabilidad p sobre el conjunto de variables V. Las propiedades de independencia condicional se extienden a la distribución p.

3.1.2. Gráficas y probabilidad

Consideremos una gráfica no dirigida $\mathcal{G}$, conformada por una familia de vértices $\mathcal{C}$ de forma que para cada dos vértices hay un arista que los une, para cada $c \in \mathcal{C}$ se tiene una función potencial ϕ_c, de forma que la función de probabilidad asociada a $\mathcal{C}$ que descrita por

$$p(x) = \prod_{c \in \mathcal{C}} \phi_c(x_c), \tag{3.1}$$

Para parametrizar la gráfica $\mathcal{G}$ es necesario tenga la estructura de árbol, en caso de que la gráfica no lo sea se reacomodan los nodos de forma que $\mathcal{G}$ se alcance una estructura de árbol T. Para la gráfica T la distribución p se interpreta como la distribución conjunta determinada por el producto de las densidades de cada nodo. También se parametriza la gráfica usando subgráficas completas de T a las que llamaremos clanes.

Ahora, sea $\mathcal{B} = \{X_v\}$ una colección de subconjuntos del conjunto de vértices V. Para cada $B \in \mathcal{B}$, $\phi_B(x)$ denota la función potencial para $x_B = (x_V)_{V \in \mathcal{B}}$. La factorización de la gráfica se distingue por ser una distribución jerárquica.

Definición 3.2 *(Ditribución jerárquica) Una distribución conjunta de p de X es $\mathcal{B}-$ jerárquica si su densidad p se factoriza de la siguiente manera*

$$p(x) = \prod_{B \in \mathcal{B}} \phi_B(x) \tag{3.2}$$

donde $\phi_B(x)$ es una función potencial para $x_B = (x_V)_{V \in \mathcal{B}}$.

3.2. Redes de Markov

Sea $\mathcal{G} = (V, E)$ una gráfica no dirigida, y sea $\{X_v\}_{v \in V}$ una colección de variables aleatorias. Se define una función potencial ϕ_A sobre X_A para subconjuntos completos de $A \subset \mathcal{G}$, esto indica que los clanes de $\mathcal{G}$ son elementos de $\mathcal{A}$, de forma que la densidad puede factorizarse como en (3.2).

Asociado con la gráfica $\mathcal{G}$ existen algunas propiedades que una medida de probabilidad p sobre X debe cumplir para $\mathcal{G}$ sea definida como red de Markov. Pearl (1985) define un $I-map$ cuando las relaciones de dependencia son representadas mediante

una gráfica no dirigida, tal que para α y β dos vértices están separados por γ, si esta separación indica independencia condicional se dice que $\mathcal{G}$ es *Markov globlal*. Para un par de vértices α y β, no relacionados, podemos encontrar un camino por el que α y β sean condicionalmente independientes dado ese camino, esta característica se conoce como la propiedad de *Markov para parejas de vértices*. Para cada vértice $\alpha \in \mathcal{G}$, éste es condicionalmente independiente a los otros nodos dada la frontera de α. Estas propiedades son equivalentes a las propiedades de independencia condicional.

Sean $\mathcal{C}_i$ y $\mathcal{C}_j$ clanes adyacentes en el árbol T, podemos asociar al arista que los une su separador (3.3), $\mathcal{S} = \mathcal{C}_i \cap \mathcal{C}_j$ de nodos de T. De esta manera si se considera una función ϕ_c para cada $c \in \mathcal{C}$, también tendremos una función ϕ_s para cada separador $s \in \mathcal{S}$, de forma que

$$p(X_v) = \frac{\prod_{c \in \mathcal{C}} \phi_c(x_c)}{\prod_{s \in \mathcal{S}} \phi_s(x_s)}, \tag{3.3}$$

donde $\{\phi_c, c \in \mathcal{C}\}$ y $\{\phi_s, s \in \mathcal{S}\}$ definen las funciones potenciales de p.

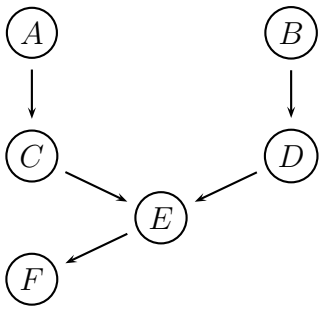

Figura 3.3: En esta gráfica se puede ver los conjuntos $\mathcal{C}_1 = A, B$ y $\mathcal{C}_2 = B, D$, separados por el nodo E.

3.2.1. Propiedades de Markov sobre *DAGs*

Las propiedades de independencia para las gráficas de Markov pueden extenderse sencillamente a gráficas dirigidas, de manera que una gráfica acíclica dirigida $\mathcal{D}$ admite una parametrización si existe una distribución definida por $p(x) = \prod_{v \in V} p(X_v \mid X_{pa(v)})$.

Una propuesta alternativa para esta propiedad está dada por Pearl y Verman (1990). Mencionan que un camino dirigido es una secuencia de vértices que formen un camino Π de α a β en $\mathcal{D}$, un camino se dice bloqueado por $\mathcal{S}$ si éste contiene un vértice $\gamma \in \Pi$ tal que $\gamma \in \mathcal{S}$ y α está $d-$*separado* de β por $\mathcal{S}$, es decir, $\alpha \perp \beta \,|\, \mathcal{S}$. Un camino dirigido que no está bloqueado por $\mathcal{S}$ se dice que está activo. Dos subconjuntos A y B se dicen $d-$*separados* por $\mathcal{S}$ si todos los caminos dirigidos de A a B están bloqueados por $\mathcal{S}$.

3.3. Redes Bayesianas

Las redes bayesianas son gráficas acíclicas dirigidas (DAG), donde cada uno de los vértices representan una variable; y los arcos, las asociaciones entre ellas.

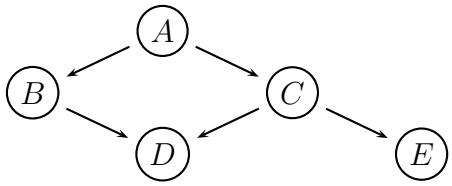

Figura 3.4: Gráfica acíclica dirigida (DAG).

La figura 3.4 representa una red bayesiana donde la dirección de los arcos indica

una relación entre variables, por ejemplo, se puede hablar de una relación de jerarquía entre los nodos $\{B, C\}$ y $\{A\}$, donde $\{A\}$ es el nodo padre y $\{B, C\}$ los nodos hijo de A. El modelo gráfico se define mediante una función conjunta basada en funciones potenciales ϕ_c asociadas a cada conjunto de nodos c.

La parametrización de una gráfica dirigida se define de manera distinta, sea una DAG definida por un conjunto de vértices V, para los que se distringue una relación jerárquica, la función de probabilidad asociada a la (3.1) se especifica mediante una distribución condicional de X_v, $v \in V$, dados sus padres, $X_{pa(v)}$,

$$p(x) = \prod_{v \in V} p(X_v \mid X_{pa(v)}) \tag{3.4}$$

donde $p(X_v \mid X_{pa(v)})$ es la probabilidad del nodo v dados sus predecesores directos. Observemos la función conjunta que describe la parte de la gráfica 3.4 formada por los nodos $\{A, C, E\}$. La relación entre A y C se establece por $p(C \mid A)$, la relación entre los tres nodos se describe por $p(x) = p(E \mid C, A)$; la influencia del nodo A sobre el nodo E se realiza a través de C, entonces $p(x) = p(E \mid C)p(C \mid A)p(A)$.

Se observa que para una gráfica dirigida $\mathcal{G}$ y una función conjunta $p(x)$ que satisfaga las restricciones de independencia, la gráfica no dirigida de $\mathcal{G}$, $\mathcal{G}^m$, también puede representar las relaciones de independencia no bajo la misma parametrización $p(x)$.

Trabajando con la representación general del árbol, los algoritmos que se usan modifican la función potencial mediante una serie de pasos que se interpretan como el paso de información entre nodos.

3.4. Algoritmo de propagación

La figura 3.1 se parametriza por $p(Rinitis \mid Edad)$, se se ha observado Edad=25, entonces esta información puede incorporarse al modelo como $p(Rinitis \mid Edad = 25)$. De modo general, las redes bayesianas permiten observar los efectos de la incorporación de información (evidencia ε) al modelo, es decir, $p(x)$ es actualizada como $p^* = p(x)\mathbf{I}_{X_A = x_A^*}$, donde $X_A = x_A^*$ ha sido observado.

Sea U un conjunto de variables, ya sea un clan o un separador, de manera que la representación marginal de este conjunto de variables se expresa como

$$p^*(x_U) = p(x_U \mid \varepsilon)p(\varepsilon).$$

La constante de normalización para cada U será $p(\varepsilon)$, obteniendo así $p(x_U \mid \varepsilon)$ que permitirá transmitir el efecto de la información a cada clan. En resumen,

$$p(X_v \mid \varepsilon) = \frac{\prod_{c \in \mathcal{C}} p_c(x_c \mid \varepsilon)}{\prod_{s \in \mathcal{S}} p_s(x_s \mid \varepsilon)}.$$

En el proceso de propagación de evidencia se distinguen dos fases, la de recolección y la fase de distribución de evidencia sobre el árbol T. La primera, consiste en la recolección de información pasando información de un clan a otro. En la segunda, el flujo de información regresa por el recorrido de colecta obteniendo las probabilidades posteriores de cada clan.

Pensemos en una gráfica acíclica dirigida con 3 nodos, X, Y y Z, de manera que Y es separador de X y Z.

Cuando $X = x$ se dice que se ha incorporado información al modelo, y es propagada a las otras variables a través de las relaciones representadas en la gráfica, en este caso, se actualiza la distibución de Z por medio de Y, por consiguiente, la

distribución posterior de Z considerando (3.4) queda dada por

$$p(Z \mid X = x) = \sum_{Y} p(Z \mid Y)p(Y \mid X = x).$$

Una vez que la evidencia ha sido propagada, se realiza un proceso de actualización de las distribuciones que consiste en actualizar X dado que se ha observado $X = x$, $p^*(X = x) = p(X = x \mid Z = z)$.

De modo general, para cada vértice X_v sea $p(X_v)$ su distribución marginal:

$$p(X_v) = \sum_{pa(X_v)} p(X_{pa(X_v)})p(X_v = x_v \mid pa(X_v) = x_{pa(X_v)}),$$

y se desea calcular $p^*(X_v) = p(X_v = x_v \mid \varepsilon)$, donde ε es la evidencia. Hay que observar dos eventos, ε_v^- que denota el evento de condicionar sobre los descendientes de v, y ε_v^+ indica que se condiciona sobre las varibales restantes que conforman la cubierta de Markov para v. Usando la regla de Bayes:

$$p^*(X_v) \propto p(\varepsilon_v^- \mid X_v, \varepsilon_v^+)p(X_v \mid \varepsilon_v^+) = p(\varepsilon_v^- \mid X_v)p(X_v \mid \varepsilon_v^+).$$

Cada vértice mantiene una versión de $\ell(x) = \prod_u p(\varepsilon_u^- \mid X_v)$ y $\nu(X_v) = p(X_v = x_v \mid \varepsilon_v^+)$. Luego, $p^*(X_v) \propto \mathbb{I}(X_v \in \mathcal{S}_v)\ell(X_v)\nu(X_v)$. Cuando se tiene información en el vértice, $p(\varepsilon_v^- \mid X_v \in \mathcal{S}_v)$, ésta es usada para actualizar $\ell(pa(X_v))$, $p(\varepsilon_v^- \mid Xpa(X_v))$. Luego, en forma descendiente para actualizar a los hijos de v, $p(X_v \mid X_v \in \mathcal{S}_v, \varepsilon_v^+, \varepsilon_v^-)$.

Figura 3.5: Gráfica acíclica dirigida con tres nodos.

Las redes bayesianes representan las relaciones de independencia condicional entre las variables. A continuación se detallan los conceptos básicos de independencia condicional.

3.5. Redes discretas

Si una gráfica no dirigida es un árbol, cualquier vértice puede considerarse como la raíz. La forma de árbol más simple es una cadena de vértices que es modelada por una cadena de Markov. Los cálculos sobre una cadena de Markov dependen del orden de los vértices. Las cadenas de Markov se definen mediante una función de transición $P(X_t = j \mid X_{t-1} = i)$, mientras que un árbol de Markov se especifica por $P(X_v \mid X_{pa(v)})$. Partiendo de la estructura de árbol, se etiquetan los vértices en orden ascendente desde la raíz, de manera que cada vértice sea precedido por sus nodos padre:

$$P(X_v) = \prod_i P(X_i \mid X_j, j < i) = \prod_i P(X_i \mid X_{pa(i)}) \tag{3.5}$$

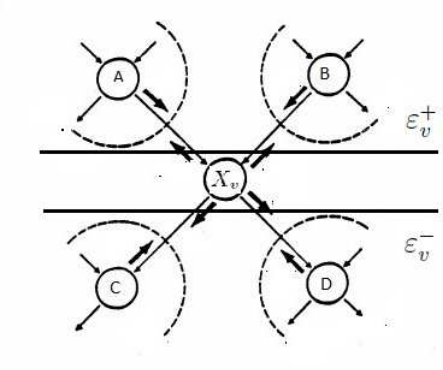

Figura 3.6: Las flechas en la gráfica indican el flujo de la información.

En una red probabilista se manejan varias variables de forma simultánea, por lo que son agrupadas en clanes, y se construye una distribución conjunta sobre el árbol de clanes, en este caso (3.5) se convierte en la representación potencial. Las redes bayesianas son un caso particular de las redes de Markov.

3.6. Clasificadores

Un clasificador es una función que asigna un objeto a una clase a partir de un conjunto de atributos. Uno de los clasificadores más efectivos es el llamado *naive-Bayes* descrito en varios artículos, Duda y Hart(1973), y Langley et al. (1992).

La red que describe a un clasificador naive-Bayes indica que la variable de la clase es el nodo padre de cada atributo, $P(C \mid A_1, A_2, \ldots, A_n)$.

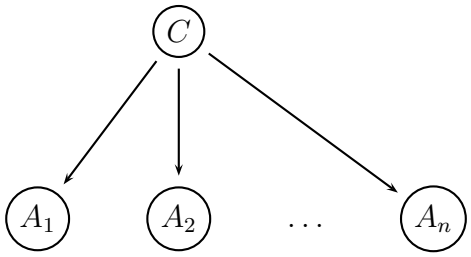

Figura 3.7: Gráfica de un clasificador naive-Bayes.

El procedimiento de clasificación es aplicar la regla de Bayes al cálculo de la probabilidad de C dado el conjunto de atributos $\{A_1, A_2, \ldots, A_n\}$ y luego predecir la clase, tomando la que tenga mayor probabilidad. Se hace un supuesto de independencia condicional entre los atributos dada la clase.

El objetivo es inducir una red que describa mejor la distribución de los datos de

entrenamiento, esto se logra utilando una función que califica a cada red dentro del conjunto de posibles redes.

El modelo *naive Bayes* no toma en cuenta las correlaciones entre atributos, aunque estas correlaciones pueden permitir una mejor descripción de los datos del modelo. El *naive Bayes* aumentado (TAN)(Friedman et al.) plantea una modificación al modelo *naive Bayes* para incorporar las correlaciones.

Por ejemplo, para el modelo *naive Bayes*, si la clase C es la variable que indica estar enfermo o no de rinitis, y, edad y sexo son los atributos de la distribución $p(Edad, Sexo \mid Rinitis)$ es la distribución inicial, que se actualiza como $p(Rinitis \mid Edad, Sexo)$ dados los valores de la variables edad y sexo.

3.6.1. Aprendizaje

Sea $U = \{A_1, A_2, \ldots, A_n\}$, donde $\{A_1, A_2, \ldots, A_n\}$ es el conjunto de atributos que definen a la clase C. La estructura gráfica que se usa define a la clase C como la raíz, y cada atributo tiene como nodo padre al nodo de la clase, $pa(A_i) = C$, la función conjunta que describe esta estructura queda:

$$p(A_1, A_2, \ldots, A_n, C) = p(C) \prod_i p(A_i \mid C)$$

Usando la definición de condicionalidad,

$$p(C \mid A_1, A_2, \ldots, A_n) = \alpha p(C) \prod_i p(A_i \mid C)$$

donde α es la constante de normalización.

El problema de aprendizaje consiste en encontrar la *mejor* red que describa a T, a partir $T = \{u_1, u_2, \ldots, u_k\} \in U$, un conjunto de datos de entrenamiento. Lo

usual es introducir una función que evalue cada red con respecto al conjunto de entrenamiento, esta función es conocida como función *score*.

Las funciones score tienen como propósito proporcionar información acerca de una distribución, de manera que podemos elegir una red $\mathcal{D}$ tal que considerando el tamaño de la red y los datos, $P_{\mathcal{D}}$ sea mínima. Existen diferentes versiones de funciones score cada una incorporando distintos elementos.

Sea $\mathcal{D} = \{\mathcal{G}, \Theta\}$ y $T = \{u_1, u_2, \ldots, u_n\}$ el conjunto de entrenamiento, considérense u_i observaciones independientes. Se define la información aportada por cada observación como $\log(1/p(x_i))$, si hay n observaciones, se puede afirmar que en promedio el valor u_i será observado $N \cdot p(u_i)$ veces. Entonces para n observaciones la información total se define como,

$$I = \sum_{i=1}^{n} n \cdot p(u_i) \log(1/p(u_i)). \tag{3.6}$$

La información promedio [6], I/n, se conoce como entropía $H(p)$. La entropía fue definida por Shannon en 1948 y, tiene propiedades teóricas que permite cuantificar la información descrita por la distribución p (2.2).

Se supone una distribución p para el conjunto de entrenamiento, y se propone una distribución q para el mismo conjunto. El problema consiste en dar una cantidad que mida la discrepancia entre los modelos q y p, para cada u_i esta cantidad se define como $\log(p(u_i)/q(u_i))$, por lo que la comparación de entropía entre dos p y q se expresa como sigue

$$H(p, q) = \sum_{i=1}^{n} p(u_i) \log \frac{p(u_i)}{q(u_i)}.$$

Otra cantidad usual, es la llamada entropía relativa, mejor conocida como infor-

mación de Kullback-Leiber:

$$KL(p,q) = \left(\log \frac{p(u)}{q(u)}\right)_p .$$

Para el caso de gráficas las distribuciones p y q están dadas por las distribuciones de la red $\mathcal{D} = \{\mathcal{G}, \Theta\}$ y la del conjunto de entrenamiento $T = \{u_1, u_2, \ldots, u_n\}$,respectivamente.

La función score MDL (*minimal description length*) de una red $\mathcal{D}$ dado T está dada por:

$$MDL(\mathcal{D} \mid T) = H(\mathcal{D}, T) + \frac{\log n}{2}\mathcal{D} \tag{3.7}$$

donde $H(\mathcal{D}, T)$ indica la entropía, en el segundo término $\mathcal{D}$ indica el número de parámetros en la red. Otra función score que basa su definición en el concepto de entropía, es el criterio de información de Akaike (AIC),

$$AIC(\mathcal{D} \mid T) = H(\mathcal{D}, T) + \mathcal{D} \tag{3.8}$$

La función score favorece gráficas completas, para ello el primer término de (3.7) reduce la complejidad de la red penalizando aquellas que contegan muchos parámetros.

3.6.2. Redes bayesianas como clasificadores

Se puede inducir una red $\mathcal{D}$ que represente $P(A_1, A_2, \ldots, A_n, C)$. Dado un conjunto de entrenamiento, cada $u_i \in T$ es una tupla de la forma $(a_1^i, a_2^i, \ldots, c^i)$, que son los valores de los atributos y el valor de la variable de la clase C. Se puede reescribir la log-verosimilitud respecto a las u_i:

$$LL(\mathcal{D} \mid T) = \sum_{i=1}^{k} \log(P_{\mathcal{D}}(c^i \mid a_1^i, \ldots, a_n^i)) + \sum_{i=1}^{k} \log(P_{\mathcal{D}}(a_1^i, \ldots, a_n^i)) \tag{3.9}$$

El primer término estima la probabilidad de una clase dados los atributos.

La definición de atributos relevantes se basa en la noción de cubierta de Markov de la variable X, es decir, los nodos vecinos de X. Condicionando sobre este conjunto a X, se dice que X es condicionalmente independiente de otras variables en la red.

Los algoritmos de aprendizaje eligen los atributos relevantes para predecir la clase, la elección hecha por el procedimiento refleja el sesgo de la función de score, en particular, la función MDL, que penaliza la adición de atributos a la cubierta de Markov de la variable clase.

La función MDL puede especializarse para la tarea de clasificación, restringiendo la log-verosimilitud en (3.9), de manera que sea la log-verosimilitud condicional de una red $\mathcal{D}$ dado el conjunto T, $CLL(\mathcal{D} \mid T) = \sum_{i=1}^{k} \log(P_{\mathcal{D}}(C^i \mid A_1^i, \ldots, A_n^i))$.

3.6.3. Naive Bayes aumentado

Este modelo permite especificar las asociaciones entre atributos, por lo que gráficamente existe un arco entre ellos. Un arco de A_i a A_j implica que existe una relación de dependencia condicional entre A_i y la clase dado A_j.

En este tipo de redes la variable de la clase no tiene nodos padres y cada atributo tiene al menos un atributo como padre además de la variable clase.

El procedimiento de aprendizaje para estas estructuras está descrito por Chow y Liu (1968) y Pearl (1988).

Una gráfica acíclica dirigida sobre $\{X_1, X_2, \ldots, X_n\}$ es un árbol si $pa(X_i)$ contiene exactamente un padre para cada X_i, a excepción de la raíz. El árbol se describe identificando el padre de cada nodo, la función $t : \{1, 2, \ldots, n\} \mapsto \{0, 1, \ldots, n\}$ define un árbol sobre $\{X_1, X_2, \ldots, X_n\}$ si para exactamente una i se tiene que $t(X_i) = 0$

(raíz) y no existe una secuencia $i_1, \ldots, i_k$ tal que $t(i_j) = i_{j+1}$ para $i \leq j \leq k$ y $t(i_k) = i_1$ (no hay ciclos). La función que define a un árbol es una donde $pa(X_i) = \{X_{pa(X_i)}$ si $t(i) > 0\}$ y $pa(X_i) = \emptyset$ si $t(i) = 0$.

El procedimiento descrito por Chow y Liu [9], reduce el problema a elegir un árbol expandido de peso máximo con máxima verosimilitud, consiste en seleccionar un subconjunto de arcos tal que esta configuración forme un árbol y la suma de los pesos en los arcos sea máxima.

1. Calcular la función de información mutua, $I_{\hat{P}_\mathcal{D}}(X_i, X_j)$, entre cada par $i \neq j$,

$$I_P(X, Y) = \sum_{x,y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)} \tag{3.10}$$

2. Construir una gráfica no dirigida en la que los vértices son las variables X. Los pesos de cada arco están dados por la función de información mutua.

3. Construir un árbol expandido de peso máximo

4. La gráfica obtenida se convierte en una gráfica dirigida al elegir una variable como raíz y fijar las direcciones de los arcos en dirección contraria a ella.

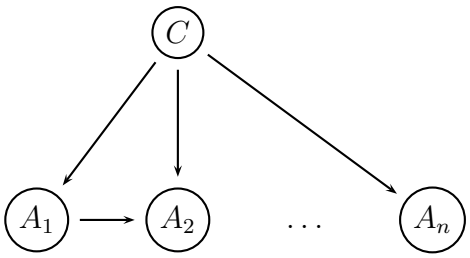

Figura 3.8: Esctructura de un naive-Bayes aumentado (TAN)

El problema consiste en encontrar una función que defina un árbol de manera que la log-verosimilitud se maximice. Este problema de optimización se resuelve usando el procedimiento anterior, y usando la función condicional de información mutua en lugar de (3.10):

$$I_P(X, Y \mid Z) = \sum_{x,y} P(x, y, z) \log \frac{P(x, y \mid z)}{P(x \mid z)P(y \mid z)}$$

El cálculo de la función se realiza para el conjunto de atributos, finalmente, se introduce un vértice C y se agrega un arco de C a cada A_i. De esta manera tenemos una descripción más adecuada de las relaciones existentes entre atributos y la clase, por lo que la gráfica obtenida ya puede utilizarse para predecir la clase de un u_i que no pertenezca al conjunto de entrenamiento.

3.7. Maximización

Para encontrar la máxima configuración se considera el paso de información entre nodos y se busca la configuración tal que maximice su función potencial.

Se tiene una distribución p sobre T, parametrizada por ϕ_c y ϕ_s, si se desea conocer la configuración gráfica más probable de todas las variables, se modifica el algoritmo de propagación sustituyendo la definición de marginalización por la siguiente,

Definición 3.3 *(Max-marginalización) Sea V el conjunto de vértices, y $W \subseteq U \subseteq V$ la expresión $M_{UW}\phi$ denota la marginal máxima de ϕ sobre W definida por*

$$M_{UW}\phi(x) = max_{z \in UW}\phi(z, x).$$

La probabilidad de la configuración de un clan, está dada por $p_c^{max}(X_C)$, que indica la probabilidad que esa configuración de nodos tiene, es decir, es la probabblidad

asociada a $X_C = x_c$, entonces (3.3) se reescribe como

$$p(X_v \mid \varepsilon) = \frac{\prod_{c \in \mathcal{C}} p_c^{max}(x_c)}{\prod_{s \in \mathcal{S}} p_s^{max}(x_s)} \tag{3.11}$$

Sea $U \in C$ y W un vecino de U, $p_W^{max}(X_W)$ pasa de W a U, y U debe encontrar la configuración que maximice su función potencial, la configuración X_U^{max}, pasa como evidencia a sus vecinos. Se construye un *árbol expandido de peso máximo.*

Capítulo 4

Rinitis alérgica

En publicaciones médicas se han utilizado redes de clasificación con el fin de predecir el tratamiento a utilizar, con este fin, en este capítulo se analiza un conjunto de datos provenientes de mediciones de pacientes con rinitis alérgica.

La rinitis alérgica es la primera causa de consulta en Alergología y su prevalencia mundial se estima entre 10 y 25 por ciento (Becerril Ángeles,2010)(*Diágnostico y tratamiento de rinitis alérgica. J30. Guía de práctica médica*).

Por otro lado, las redes bayesianas como clasificadores han tenido participación en numerosas áreas, entre ellas la medicina, han desarrollado un papel de asistencia en el diágnostico de múltiples padecimientos (Bonis,2004) *Sistemas informáticos de soporte a la decisión clínica*).

4.1. Preliminares

Decisiones clínicas

La clasificación es uno de los problemas que requiere de información que permita la estimación de una probabilidad que indique la pertenencia de un sujeto a una clase basándose en la evidencia. Dado un conjunto de n objetos, éstos serán clasificados en K clases , cada objeto tiene asociado un vector de mediciones (atributos), $x \in \mathbb{R}^p$. La proporción de casos de la clase k en la población se define por π_k, mientras que la distribución de la clase k se identifica por $p_k(x)$. El propósito es clasificar un objeto en alguna de las $K+2$ posibles decisiones, $1, \ldots, K, \mathcal{D}, \mathcal{O}$, basados en la observación $X = x$, las decisiones $1, \ldots, K$ indica que el objeto corresponde a la clase k, la decisión $\mathcal{D}$ indica duda de la clase a la que pertenece el objeto, finalmente, la decisión $\mathcal{O}$ indica que el objeto no pertenece a ninguna de las K clases.

Los modelos de apoyo a la desición clínica están diseñados para formalizar el proceso de clasificación, ayudando al personal sanitario a tomar decisiones " preventivas, diagnósticas o terapéuticas". Los primeros modelos surgieron a finales de los cincuenta, utilizadan algoritmos lógicos sobre la información introducida. Otra estrategia incluyo el uso del teorema de Bayes que asumía independencia entre los síntomas o pruebas, relación que en la mayoría de los casos no puede establecerse con claridad.

A finales de los ochenta, se desarrolla un sistema que una vez introducidos los datos era capaz de elaborar la cadena de razonamiento *deductivo basado en el conocimiento de la fisiopatología humana* mediante el cual se obtiene una respuesta. Estos sistemas se encuentran bajo la evaluación de expertos, considerando si la respuesta dada sería la que un médico experto en el área proporcionaría.

Rinitis alérgica

La rinitis alérgica es una enfermedad inflamatoria crónica de la mucosa nasal, cuyos síntomas principales, desencadenados por la exposición a alérgenos, son la rinorrea, obstrucción nasal, prurito nasal y estornudos. Frecuentemente, los pacientes con rinitis alérgica presentan síntomas conjuntivales y de asma. En en caso de los niños que la presentan rinitis alérgica, tienen mayor riesgo de presentar asma si no se trata adecuadamente.

Los antecedentes familiares y la exposición a ambientes adversos como el tabaquismo familiar, ácaros del polvo y epitelios de animales, son factores de riesgo para el desarrollo de la rinitis alérgica [3].

El tratamiento farmacológico adecuado es difícil de determinar debido a las reacciones secundarias que puede desencadenar, el tramatamiento más eficaz para la rinitis alérgica es el uso de esteroides nasales, sin embargo en la práctica clínica son más usados los antihistamínicos.

Los antihistamínicos de primera generación tienen efectos sedantes y anicolinérgicos. Los de segunda generación causan poca sedación, tienen un inicio de acción más rápido y su efecto dura más de 24 horas. En adultos mayores no se recomienda el uso de antihistamínicos de primera generación por su acción sobre el sistema nervioso central.

Los esteroides nasales se consideran la primera línea de tratamiento de la rinitis alérgica persistente, sobre todo en la obstrucción, y se ha demostrado que dosis recomendadas no afectan el crecimiento. Los descongestionantes son muy efectivos para aliviar la obstrucción nasal, pero no se recomienda su uso en menores de un año por su margen estrecho entre dosis terápeuticas y tóxicas. Tampoco es recomendado

su uso en adultos mayores, embarazadas, pacientes con hipertensión, cardiopatía, hipertrofia prostática, glaucoma y los que usan β-bloqueadores e inhibidores de la MAO (monoaminooxidasa).

El tratamiento de la rinitis alérgica se vuelve complejo de determinar. Por eso el mejor tratamiento para cada paciente depende de sus características, por ejemplo, variables como edad y sexo son determinantes para indicar el mejor fármaco para tratamiento.

Con el fin de estudiar la capacidad predictiva de los modelos de redes neuronales y redes bayesianas, se utiliza un conjunto de datos que contiene información de cuatro hospitales sobre el medicamento que se administra a pacientes con rinitis alérgica. En total, se tienen datos de 2200 pacientes. Las variables medidas son:

1. Edad (Age)
2. Sexo (Sex)
3. Presión sanguínea (BP)
4. Nivel de colesterol (Cholesterol)
5. Nivel de sodio (Na)
6. Nivel de potasio (K)

Las mediciones de los niveles de sodio y potasio en la sangre son incluídos debido a que funcionan como elementos mediadores del intercambio celular. Algunos de los medicamentos usados para el tratamiento de la rinitis alérgica aumentan los niveles de actividad de la Na/K adenín trifostasa, cuya modulación es un posible mecanismo de la acción antialérgica.

En este ejemplo, se pretende predecir de 5 medicamentos, que se etiquetan como DrugA, DrugB, DrugC, DrugX y DrugY, el medicamento a usar para el tratamiento de nuevos pacientes basados en las características del paciente.

4.2. Conociendo Weka

Este paquete consiste en una colección de algoritmos útiles en tareas de regresión, asociación y clasificación, todos ellos implementados en Java. Weka es un paquete de minería de datos de distribución libre, desarrollado por la Universidad de Waikato. El nombre de Weka hace referencia a un ave endémica de Nueva Zelanda.

En esta tesis se utiliza la sección de *Classify*, que nos permite implementar algoritmos descritos previamente en este documento. Los modelos usados en la clasificación de patrones que se utilizarán son:

1. Perceptrón multicapa.
2. Clasificador naive-Bayes.
3. Red Bayesiana general.

Luego del periodo de entrenamiento, los modelos fueron evaluados usando un conjunto de entrenamiento y uno de validadción, 60 y 40 por ciento respectivamente. En las secciones siguientes se explican los resultados de los ajustes de los modelos.

4.2.1. Red neuronal

El modelo de red neuronal que se utiliza es un perceptrón multicapa (*MultilayerPerceptron*), donde las variables categóricas son codificadas en sus categorías y todos los

atributos contínuos son normalizados, y las neuronas usan la función sigmoide como función de activación. Se usa el algoritmo de retropropagación explicado en (1.3.4).

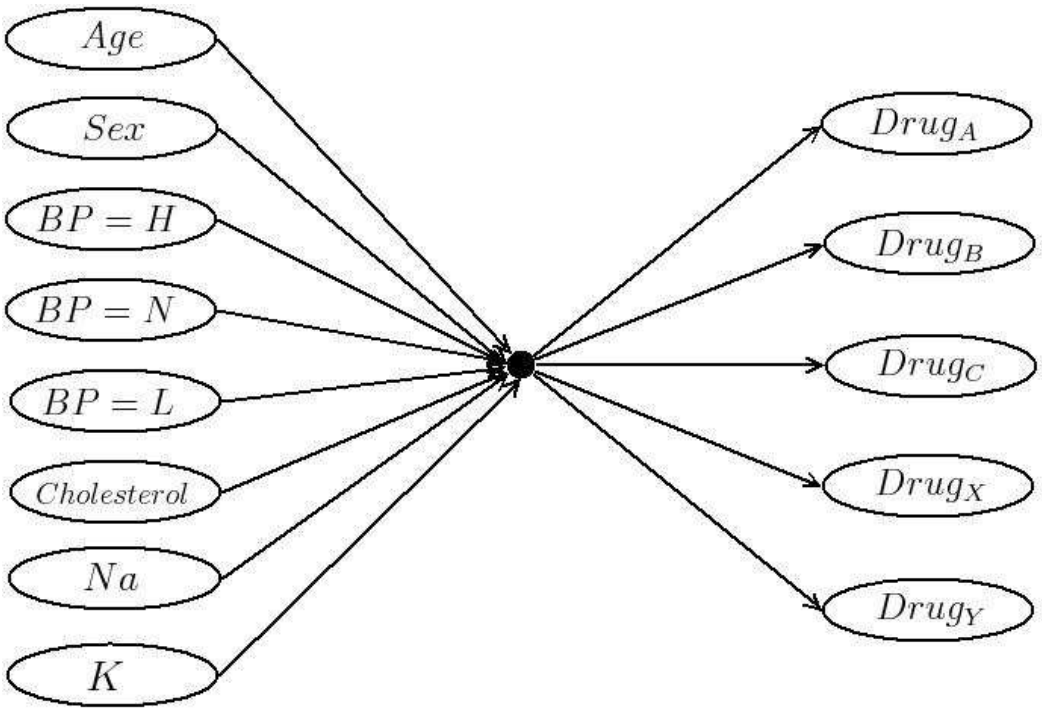

Figura 4.1: Perceptrón con una neurona y una capa oculta.

Se observa mayor precisión con un mayor número de neuronas y de capas ocultas, sin embargo, el modelo se vuelve difícil de interpretar. A continuación se muestra la precisión alcanzada con distinto número de neuronas en dos capas ocultas, para cada modelo se incluye el porcentaje de elementos clasificados correctamente y el estadístico kappa.

Debido a la complejidad los modelos, en adelante nos referiremos al perceptrón con una capa oculta.

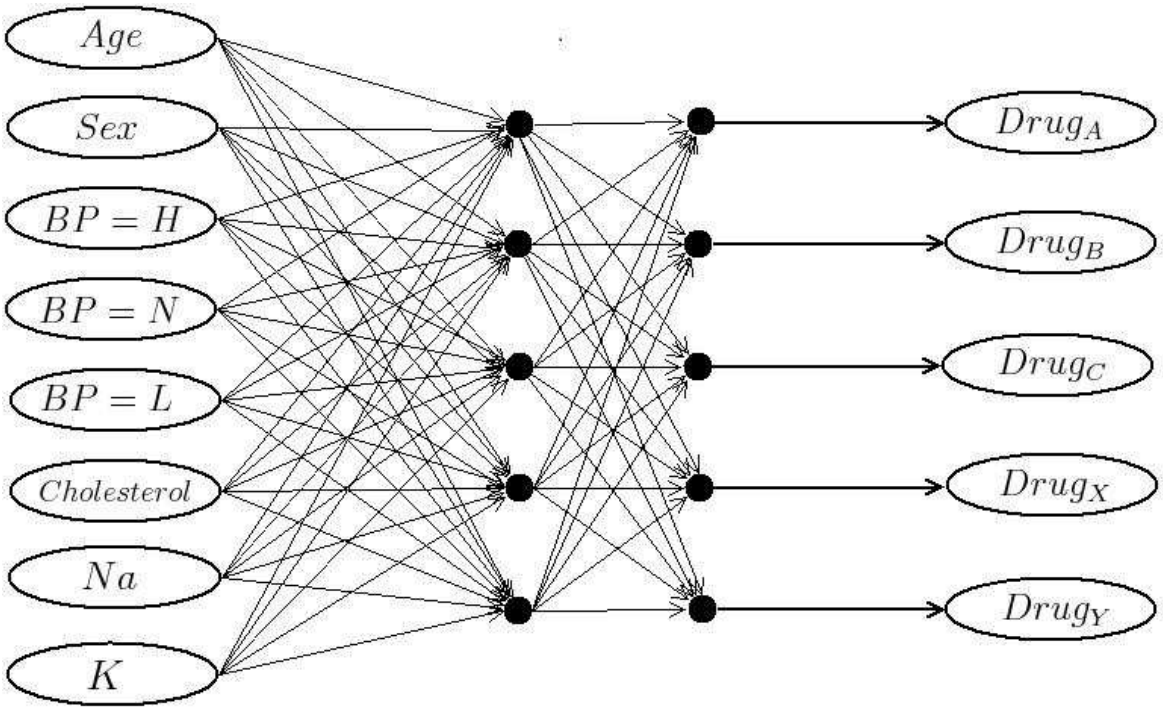

Figura 4.2: Perceptrón con 5 neuronas y dos capas ocultas.

Cuadro 4.1: Precisión obtenida por un perceptrón con una capa oculta.

Número de neuronas	Precisión	Estadístico kappa
1	70.34	0.5426
2	90.34	0.861
3	93.40	0.9053
4	98.97	0.9853
5	98.86	0.9837

Cuadro 4.2: Precisión_A corresponde al modelo sin estimar y Precisión_B con estimación de distribución.

Fármaco	Precisión_A	Precisión_B
DrugA	0.918	0.929
DrugB	0.928	0.952
DrugC	0.959	0.958
DrugX	0.942	0.942
DrugY	0.89	0.891

4.2.2. Naive-Bayes

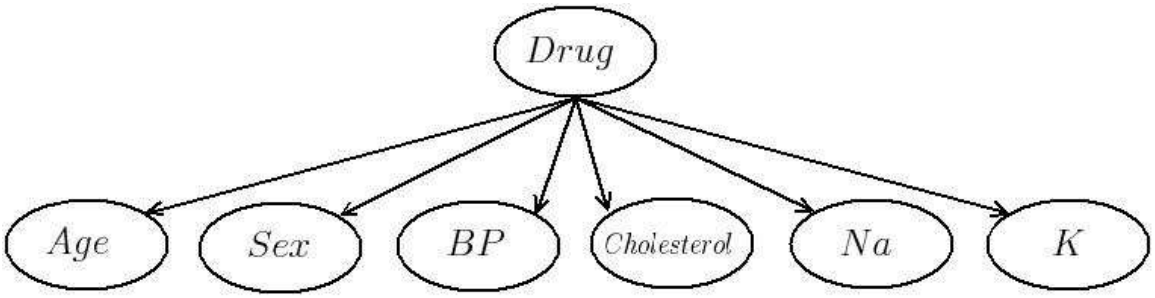

Figura 4.3: Clasificador *Naive-Bayes* para el modelo de rinitis alérgica.

El clasificador *NaiveBayes* implementado en Weka tiene dos versiones, la primera mantiene la distribución empírica de los datos, y la segunda supone una distribución normal para las variables en la red. Se utilizan ambas versionas y se compara la precisión del clasificador con y sin estimación de distribuciones.

Como se observa en la tabla (4.2), la ganancia en precisión entre el modelo con y sin estimación es marginal, por lo que se conserva el modelo que utiliza la distribución

empírica de los datos.

4.2.3. Red bayesiana

Se utiliza el clasificador *BayesNet* para evaluar dos modelos de redes bayesianas, uno, el clasificador Naive-Bayes aumentado, y el segundo una red bayesiana general.

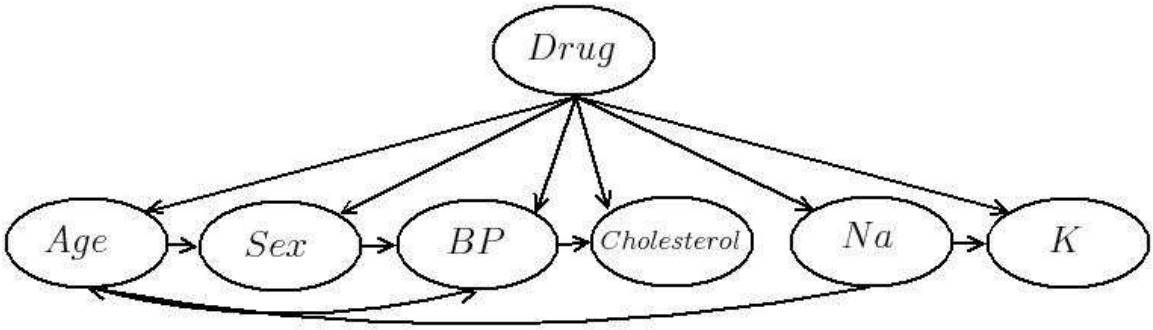

Figura 4.4: Modelo gráfico de un TAN usando la función de información MDL.

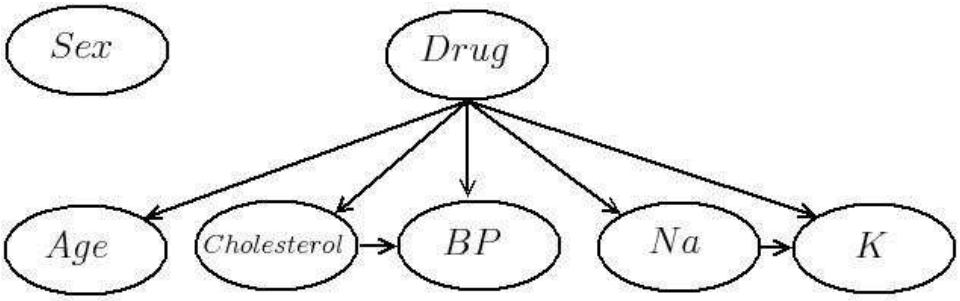

Figura 4.5: Red bayesiana general generada por una muestra obtenida mediante el algoritmo Simulated Annealing.

Cuadro 4.3: Precisión observada por los modelos de redes bayesianas.

	Precisión
Modelo TAN	0.9136
Modelo Red bayesiana general	0.9125

1. Usando el algoritmo descrito en (3.6.3) se contruye un Naive Bayes aumentado (TAN), utilizando como función de información la función MDL (3.7). Este procedimiento da como resultado la gráfica (4.4).

2. Mediante el método simulated annealing,(2.6) se genera una muestra para evaluar la función de información mutua (3.10) dentro del algoritmo (Chow,1968) que genera la gráfica (4.5).

Se observan una precisiones similares para ambos modelos, sin embargo, en el modelo de la red bayesiana general el atributo sexo no resultó incluido en la gráfica.

4.3. Resultados

La tabla (4.4) muestra la precisión de cada uno de los modelos utilizados, es importante mencionar que la precisión por sí sola no es un indicador para decidir el mejor modelo, todo depende del propósito de estudio.

La precisión fue calculada como el porcentaje de elementos correctamente clasificados del conjunto de validación, el conjunto de datos (n=2200) fue dividido en dos conjuntos, el 60 % como conjunto de entrenamiento y el resto como conjunto de validación. El estadístico κ es un índice entre las concordancias observadas y el

Cuadro 4.4: Precisión observada por cada modelo.

Modelo	Precisión	Estadístico κ
Perceptrón multicapa, 1 neurona	0.7022	0.5415
Perceptrón multicapa, 5 neuronas	0.9886	0.9837
Naive-Bayes	0.9147	0.8767
Red bayesiana,TAN	0.9136	0.8783
Red bayesiana, simulated annealing	0.9125	0.8765

total de observaciones, de tal forma que se excluyen las concordancias atribuibles al azar. Toma valores entre -1 y 1, mientras más cercano a 1 mayor es el grado de concordancia.

El perceptrón multicapa con 5 neuronas se observa como un buen clasificador, pero no es un modelo que muestre una relación entre las variables. Al mismo tiempo, los modelos basados en el teorema de Bayes también resultan buenos clasificadores, con la ventaja de que su estructura gráfica representa las relaciones de independencia entre los atributos y la clase.

El clasificador *naive-Bayes* sólo toma en cuenta la relación entre los atributos con la clase, e ignora la relación que entre ellos pueda existir, por lo que no explica en su totalidad las relaciones entre las variables. Con todo y ello, predice mejor el uso de los fármacos C, X y A como tratamiento de la rinitis alérgica.

El modelo definido por la gráfica TAN indica la aportación de información de cada variable con respecto a la clase y al menos a otro atributo, (3.10), para determinar la gráfica de peso máximo se utiliza una función que califique a la red, Weka utiliza varias funciones, entre ellas la función de entropía, (2.2),la función *minimum*

Cuadro 4.5: Precisón para cada fármaco.

Fármaco	PM, 1 capa	NB	BN-TAN	BN_SA
Drug Y	0.973	0.89	0.943	0.948
Drug C	0	0.959	0.894	0.894
Drug X	0.466	0.942	0.89	0.89
Drug A	0	0.918	0.858	0.853
Drug B	0	0.928	0.935	0.935

description length, (3.7),el criterio de Akaike, entre otras.

En la gráfica TAN se observa una asociación entre las variables Na y K, esto puede explicarse debido a que la concentración de una está en dependencia de la otra. La variable sexo no tiene gran aportación, sin embargo este modelo no rechaza ningún atributo. Por su parte, la variable que mide presión sanguínea está relacionada con la edad del paciente, en el caso del nivel de colesterol muestra una relación con la medición de presión sanguínea.

La gráfica resultante al usar el algoritmo *simulated annealing*, muestra que el atributo sexo no es relevante para el modelo y exhibe una tasa de clasificación similar al obtenido usando todas las variables. Esta gráfica elimina redundancias con respecto al modelo TAN y pone de manifiesto las relaciones más importantes.

Ambas redes cumplen con el propósito de resolver el problema de maximización propuesto en (3.7), las dos tienen una estructura gráfica parecida y los valores de las funciones de información son similares. En ambos modelos, la estructura gráfica no es única, y no hay garantía de que la dirección de las aristas sea la correcta, esto debido

a que la dirección es asignada con el propósito de encontrar una red que minimice el error de predicción.

4.4. Conclusiones

El funcionamiento de las redes neuronales consiste en encontrar una relación funcional entre los elementos de entrada y los de salida, por lo que funcionan bien como clasificadores. Como se observó mejoran su precisión con un mayor número de capas ocultas. Sin embargo, muchos autores mencionan que su funcionamiento como una *caja negra* y su estructura no admiten una explicación de las asociaciones entre variables.

Como vemos en las gráficas del perceptrón multicapa, distinguimos entre la capa de entrada y la de salida, sin tener una idea clara de la interacción de las variables con las capas ocultas. Si sólo interesara el poder predictivo del modelo las redes neuronales serían las indicadas.

Por su parte, las redes probabilistas también resultan clasificadores eficaces, con la ventaja de que incluyen las relaciones de independencia condicional entre variables en el modelo. Las relaciones de independencia condicional que se establecen hacen que el modelo sea sencillo de entender y de modificar. La descripción gráfica de estas relaciones hace posible identificar la interacción entre variables permitiendo agrupar variables relacionadas o eliminar variables que tengan poco impacto, como es el caso de la variable de sexo en la red bayesiana.

Capítulo 5

Conclusiones

El propósito de este trabajo es observar el desempeño de dos modelos matemáticos usados en la clasificación de elementos, las redes neuronales y las redes bayesianas. En el ejemplo presentado ambos modelos son usados con el fin de comparar su eficacia como modelos predictores.

La modelación de un problema mediante redes neuronales resulta en un modelo funcional que al asignar un peso a cada una de las variables, define un modelo predictivo que nos ayuda a indicar la pertenencia de un elemento a una clase. Las redes neuronales son eficaces clasificadores, su naturaleza funcional no permite estblacer una relación entre las variables involucradas.

Por otro lado, las redes bayesianas son modelos que representan las relaciones de independencia condicional entre las variables por lo que también son usadas en la modelación de causalidad. Las relaciones de independencia en las redes bayesianas se pueden establecer *a priori* utilizando la información conocida del problema, o bien, utilizando funciones de información cuya evaluación mide la información que una

variable X aporta a una variable Y, es decir, si el estado tomado por X interfiere en la definición del estado que puede tomar la variable Y.

El modelo de redes neuronales asume una relación no lineal entre todas las variables, por ello, en la representación gráfica las redes presentan conectividad total sin que esto indique una verdadera influencia de una variable sobre las demás. Mientras tanto, las redes bayesianas reproducen las relaciones de independencia por lo que gráficamente la conexión entre una variable y otra indica la influencia que una tiene sobre la otra. En el ejemplo, se utilizan funciones de información para determinar la magnitud de esta influencia.

Los algoritmos de aprendizaje entre ambos modelos representan diferentes ámbitos de la información. Las redes neuronales *aprenden* mediante funciones que representan relaciones secuenciales y jerárquicas, mientras que las redes neuronales utilizan el teorema de Bayes.

En base a los resultados obtenidos, se observa que ambos modelos funcionan como predictores, sin embargo dependerá del problema para para determinar el modelo a utilizar.

Apéndice 1

Cadenas de Markov

Un proceso estocástico se entiende como un proceso aleatorio que evoluciona en el tiempo, se considera una variable aleatoria X y los posibles estados $\{x_1, x_2, \ldots, x_t\}$ que puede tener en el tiempo discreto.

Una cadena de Markov es una secuencia de variables aleatorias, con espacio de estados E, que modifican su estado en el tiempo, con una probabilidad de transición entre un estado al tiempo t y otro en el tiempo $t+1$. Esta probabilidad de transición se especifica para los posibles estados que puede tomar la variable aleatoria X, suponemos que X tiene un espacio de estados finito, entonces, en el tiempo t, X puede tomar cualquiera de sus n estados, y en el tiempo $t+1$, X puede tomar $n-1$ estados difernte o no cambiar. De esta manera se deben especificar $n \times n$ probabilidades de transición, que pueden expresarse en una matriz M de $n \times n$, que se conoce como matriz de transición.

Cada una de las entradas de la matriz M indica la probabilidad de transición del estado i al estado j en el paso de una unidad de tiempos, es decir, $p_{ij} \leq 0$, y se

especifican como $p_{ij} = p(X_{t+1} = i \mid X_t = j)$; si además la suma de sus filas es 1, $\sum_{j=1}^{n} p_{ij} = 1$, entonce a M se le llama matriz estocástica.

De acuerdo a la definición de proceso estocástico es posible conocer la probabilidad de que el estado j sea alcanzado a partir del estado i en k unidades temporales, es decir, $p_{ij}^{k} = p(X_{k-1} = j \mid X_0 = i, X_1 = i_1, \ldots, X_{k-2} = i_{k-2})$. Particularmente, se llama cadena de Markov a un proceso donde las probabilidades de transición satisfacen la siguiente propiedad,

$$p(X_{t+1} = j \mid X_0 = i, X_1 = i_1, \ldots, X_t = i_t) = p(X_{t+1} = j \mid X_t = i_t).$$

Esta propiedad indica que las cadenas de Markov no guardan memoria de sus cambios de estado, por lo que sólo basta conocer su estado previo. Las cadenas de Markov permiten determinar la probabilidad de que el proceso entre en algún estado a un cierto tiempo, sin embargo, destaca el hecho de que ocurridas varias transiciones estas probabilidades convergen a valores particulares.

Por medio de la expresión de Chapman-Kolmogorov es posible calcular las probabilidades de transición a partir de estados intermedios de la cadena, lo cual queda expresado de la siguiente manera:

$$p_{i,j}^{n+m} = \sum_{z \in E} p_{i,z}^{n} p_{z,j}^{m} = p(X_{n+m} = j \mid X_0 = i).$$

Donde E indica un espacio de estados finito, y $p_{i,j}^{n+m}$ indica la entrada (i, j) de la $(n + m)-$ésima potencia de la matriz de transición M. Cuando el número de transiciones tiende a infinito se puede observar un comportamiento asintótico,

$$\lim_{n \to \infty} [M]^n = Q.$$

Q es una matriz que contiene un vector invariante π. Al observar la evolución de la matriz M pueden distinguirse los siguientes tipos de estados:

1. *Absorbentes,* $p(T_x = 1 \mid X_0 = x) = 1$.

2. *Transitorios,* $p(T_x < \infty \mid X_0 = x) < 1$.

3. *Recurrentes,* $p(T_x < \infty \mid X_0 = x) = 1$.

donde T_x indica el tiempo en el que la cadena entra por primera vez al estado x luego de que el estado inicial fue x. Utilizando estas definiciones se han identificado cadenas que describen un comportamiento particular y para las que se tienen resultados de convergencia, estas cadenas se conocen como cadenas especiales y se explican a continuación.

Cadenas especiales

La idea de convergencia de una cadena de Markov tiene que ver con la probabilidad de que la cadena tome un estado e cuando se han observado t transiciones, y t tiende a ∞. En esta sección se describen las llamadas cadenas regulares y cadenas absorbentes.

Una cadena de Markov se dice que es regular si existe alguna potencia de la matriz de transición M que tenga todas sus entradas estrictamente positivas. Para este tipo de cadenas existe el siguiente resultado:

Teorema 1.1 *Sea M la matriz de transición de una cadena de Markov regular $\{X_n\}_{n\in\mathbf{N}}$ con espacio de resultados finito. Se cumplen entonces las siguientes propiedades:*

1. *Si n tiende a infinito, las potencias de M^n se aproximan a una matriz Q tal*

que todos sus renglones son iguales a un mismo vector q de probabilidad con componentes estrictamente positivas.

2. *El vertor q resulta ser el vector de probabilidad invariante para la cadena y para cualquier otro vector v tal que* $vM = v$.

3. $\lim_{n\to\infty} p(X_n = x) = q_x$, *para algún* $x \in E$, *y* $\lim_{n\to\infty} p_y(X_n = x) = q_x$ *donde* $x, y \in E$. *La probabilidad* q_x *es la* $x-$*ésima componente de q.*

Lo que nos indica este teorema es que, dada una cadena regular a partir de la potencia $n-$ésima de la matriz de transición las probabilidades de transición no van a cambiar y podrá conocerse el comportamiento límite de la cadena.

Se dice que una cadena de Markov es *absorbente* si para una matriz de transición M se identifica al menos un estado absorbente y, que desde cualquier estado no absorbente es posible ir a alguno absorbente. El resultao para este tipo de cadenas queda resumido de la siguiente manera:

Proposición 1.2 *Si* $\{X\}$ *una cadena absorbente entonces, para cada estado* $x \in E$,

$$p(\textit{existe } n \in \mathbf{N} \textit{ tal que } X_n \in A \mid X_0 = x) = 1.$$

Donde A es un conjunto de estados absorbentes, de manera que $A \subset E$.

Considérese A un conjunto de E de estados absorbentes, la proposición indica que dada una cadena absorbente con probabilidad uno se puede llegar a un estado absorbente partiendo de cualquier estado de la cadena. Las demostraciones de esta proposición y el teorema anterior pueden consultarse en (Caballero,2004).

Existe la generalización de los teoremas de convergencia y se basa en la defincición del tiempo promedio de recurrencia„ es decir, con qué rapidez se vuelve a entrar a un estado. Este teorema de generalización se trata en (Caballero,2004).

Bibliografía

[1] Alexander, Igor y Morton,Hellen. *An Introduction to Neural Computing.* Chapman& Hall, 1990.

[2] Anderson,James A.. *An Introduction to Neural Networks.* The MIT Press, 1995.

[3] Becerril,Ángeles y Martín et al.. *Guía de práctica clínica. Diagnóstico y tratamiento de rinitis alérgica.* IMSS, 2010.

[4] Bonis, Julio y Sancho, Juan J. y Sanz, Ferran. *Sistema informático de soporte a la decisión clínica.* Medicina Clínica, 122:39–44, 2004.

[5] Caballero, M.E. y Uribe Bravo, G. y Velarde,C. y Rivero, V.M.. *Cadenas de Markov. Un enfoque elemental.* Sociedad Matemática Mexicana, 2004.

[6] Carter Tomm. *An introduction to information theory and entropy.* Complex Systems Summer School, 2007.

[7] Casella, George y Berger, Roger L. *Statistical Inference.* Duxbury Thompson Learning, 2001.

[8] Cheng, Jie y Greiner,Russell . *Learning bayesian belief network classifiers: Algorithms and system.* Procededings of the 14 Bienal Conference of the Canadian

Society on Computational Studies of Intelligence: Advances in Artificial Intelligence, 141–151, June 2001.

[9] Chow, C.K. y Liu,C.N..
Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory, IT-14, 1968.

[10] Lauritzen,Stefen y Spiegelhalter,David y Cowell,Robert y Dawid,Philip.
Probabilistic Networks and Expert Systems. Springer-Verlag, 1999.

[11] Faussett Laurene . *Fundamentals at Neural Networks.* Prentice Hall, 1994.

[12] Goldszmidt,Moises y Friedman, Nir y Geiger,Dan.
Bayesian network classifiers. Machine Learning, 29(2-3):131–163, 1997.

[13] Isasi Viñuela, Pedro y Galván León, Inés M..
Redes neuronales artificiales. Pearson Prentice Hall, 2004.

[14] Kak Avinash. *ML, MAP, and Bayesian - The Holy Trinity of Parameter Estimation and Data Prediction.* An RVL Tutorial Presentation, Summer 2008, 2010.

[15] Lauritzen,S.L. y Wermuth, N.. *Graphical models for associations between variables, some of which are qualitative and some quantitative.* The Annals of Statistics, 17:31–57, 1989.

[16] Tan,Tele y Mittal,Ankush y Kassim,Ashraf.
Bayesian Network Technologies: applications and graphical models. IGI Publishing, 2007.

[17] Paz A. . *A full characterization of pseudographoids in termns of families of undirected graphs.* Technical report, Technion Computer Science Departemnt Haifa and Congnitive Systems Laboratory UCLA, September 1987.

[18] Pearl Judea. *A constraint - propagation approach to probabilistic reasoning.* En Proceedings of the Uncertainty in Artificial Intelligence Annual Conference on Uncertainty in Artificial Intelligence (UAI-85), Amsterdam, NL, 1985. Elsevier Science.

[19] Ripley B.D.. *Pattern Recognition and Neural Networks.* Cambridge University Press, 1996.

[20] Robert Christian.
The Bayesian Choice: a Decision-Theoretic Motivation. Springer-Verlag, 1994.

[21] Robert, Christian y Cassella,George . *MonteCarlo Statistical Methods.* Springer-Verlag, second ed. edition, 2004.

[22] Spiegelhalter,D. y Wilks, W.R. y Richardson,S..
Markov Chain Monte Carlo in Practice. Chapman& HAll/CRC, 1996.