UNS

ESCUELA DE POSGRADO

"REDES BAYESIANAS CON ALGORITMOS BASADOS EN RESTRICCIONES, SCORES E HIBRIDOS APLICADOS AL PROBLEMA DE CLASIFICACIÓN"

TESIS PARA OBTENER EL GRADO DE DOCTOR EN ESTADÍSTICA MATEMÁTICA

AUTOR:

Dr. Carlos López de Castilla Vásquez

ASESORA:

Ph. Dra. Frida Rosa Coaquira Nina

NUEVO CHIMBOTE - PERÚ
2016

UNS
ESCUELA DE
POSGRADO

CONSTANCIA DE ASESORAMIENTO DE TESIS

YO, FRIDA ROSA COAQUIRA NINA, doy conformidad de haber sido asesora del informe de tesis titulado: "REDES BAYESIANAS CON ALGORITMOS BASADOS EN RESTRICCIONES, SCORES E HIBRIDOS APLICADOS AL PROBLEMA DE CLASIFICACIÓN"; Que tiene como autor al Doctorando Carlos López de Castilla Vásquez, que ha sido elaborado de acuerdo al Reglamento de Normas y Procedimientos para obtener el Grado De Doctor En Estadística Matemática, en la Escuela de Posgrado de la Universidad Nacional del Santa.

Nuevo Chimbote, Junio, del 2018.

Ph. Dra. FRIDA ROSA COAQUIRA NINA
ASESOR

UNS
ESCUELA DE
POSGRADO

HOJA DE CONFORMIDAD DEL JURADO EVALUADOR

"REDES BAYESIANAS CON ALGORITMOS BASADOS EN RESTRICCIONES, SCORES E HIBRIDOS APLICADOS AL PROBLEMA DE CLASIFICACIÓN"

TESIS PARA OPTAR EL GRADO DE DOCTOR EN ESTADÍSTICA MATEMÁTICA

Revisado y Aprobado por el Jurado Evaluador

Dr. César Higinio Menacho Chiok
PRESIDENTE

Dr. Jorge Chue Galardo
SECRETARIO

Ph. D. Frida Rosa Coaquira Nina
VOCAL

Dedicatoria

A mi hijo Matías
por todas las horas robadas en este trabajo
que no pude estar contigo

Agradecimientos

A mi asesora de tesis
PhD. Frida Coaquira Nina
por su continuo apoyo y sugerencias
sin los cuales no hubiera podido
culminar con este trabajo de tesis

Índice general

Índice de tablas

Índice de figuras

Resumen

Las redes Bayesianas son gráficos acíclicos dirigidos que codifican las relaciones de dependencia e independencia condicional en un conjunto de variables predictoras. En este trabajo de investigación se presentan tres algoritmos que permiten obtener la estructura que define una red Bayesiana. Sobre esta estructura se construyeron clasificadores, incluyendo una variable dependiente en el gráfico que tiene las clases o categorías de interés, obteniendo un rendimiento predictivo similar en comparación con los clasificadores por redes Bayesianas tradicionales Naive Bayes y TAN. Se presenta también el algoritmo de selección de variables Statistically Equivalent Signature obteniendo resultados similares a los clasificadores construidos con todas las variables predictoras. Finalmente, se presenta un caso de aplicación usando los datos correspondientes a la Encuesta Nacional de Innovación Manufacturera 2015 para analizar si las empresas peruanas realizan el proceso de innovación de producto, obteniendo una tasa de elementos correctamente clasificados de aproximadamente 73 %.

Palabras claves: redes Bayesianas, clasificador, naive Bayes, TAN, selección de variables.

Summary

Bayesian networks are directed acyclic graphs that code the relationships of dependence and conditional independence in a set of predictor variables. In this research work, three algorithms are presented to obtain the structure that defines a Bayesian network. Classifiers were built on this structure, including a dependent variable in the graph that has the classes or categories of interest, obtaining a similar predictive performance compared to the classifiers by traditional Bayesian networks Naive Bayes and TAN. The Statistically Equivalent Signature variable selection algorithm is also presented, obtaining similar results to the classifiers constructed with all the predictor variables. Finally, it is presented a case of application is presented using the data corresponding to the National Survey of Manufacturing Innovation 2015 to analyze if Peruvian companies carry out the product innovation process, obtaining a rate of correctly classified elements of approximately 73 %.

Keywords: Bayesian networks, classification, naive Bayes, TAN, variable selection.

Capítulo 1

Introducción

1.1. Realidad problemática

En muchas áreas de las ciencias es de interés realizar el proceso de clasificación de un grupo de observaciones en diferentes categorías o clases definidas por una variable respuesta Y. Las observaciones se encuentran descritas por un conjunto de variables predictoras $(X_1, X_2, \cdots, X_p)$ que en la mayoría de los casos se encuentran relacionadas. Uno de los clasificadores Bayesianos más populares es Naive Bayes gracias a su simplicidad, eficiencia y bajo error de clasificación [Duda et al., 1973]. Este clasificador considera que todas las variables predictoras son condicionalmente independientes dado el valor de la categoría definida por la variable respuesta. Su alto rendimiento predictivo es sorprendente, ya que el supuesto de independencia no es realista en la mayoría de contextos. Otro clasificador Bayesiano, basado en el algoritmo de [Chow and Liu, 1968] es llamado Tree Augmented Naive Bayes, TAN, [Friedman et al., 1997] que considera relaciones de dependencia entre una variable predictora y a lo sumo otra. Ambos clasificadores tienen, en términos generales, un buen desempeño predictivo.

Una de las ventajas de usar clasificadores Bayesianos es la posibilidad de poder observar de manera sencilla las relaciones de dependencia e independencia condicional que existen entre las variables predictoras a diferencia de aquellos clasificadores, como las redes neuronales, que utilizan algoritmos complejos que son difíciles de entender e interpretar. A pesar de contar en la actualidad con muchos modelos que permiten realizar el proceso de clasificación, éstos imponen algunas

restricciones sobre el conjunto de variables predictoras, dejando de lado las relaciones naturales que existen entre ellas y que pueden ser aprovechadas para mejorar el comportamiento predictivo del clasificador.

En este trabajo de investigación se presentan tres algoritmos basados en restricciones, scores e híbridos, que permiten obtener la estructura de dependencia e independencia condicional entre las variables predictoras para la construcción posterior del clasificador. Ademas, se presenta el algoritmo Statistically Equivalent Signature, SES, como un método de selección de variables predictoras inspirado en los principios del algoritmo basado en restricciones [Tsamardinos et al., 2012] que permite obtener grupos de variables predictoras con comportamiento predictivo equivalente.

1.2. Estado del arte del tema de investigación

1.2.1. Redes Bayesianas

Las redes Bayesianas han surgido en los últimos años como una poderosa técnica de minería de datos para el reconocimiento de patrones y el proceso de clasificación [Heckerman et al., 1995]. Una red Bayesiana es un modelo gráfico que permite representar las relaciones de dependencia e independencia condicional para un conjunto de variables. Se define como un gráfico acíclico dirigido en el que cada nodo representa una variable aleatoria que tiene asociada una función de probabilidad condicional [Edwards, 2000] y [Scutari and Strimmer, 2010].

La estructura de la red Bayesiana simplifica la representación de la función de probabilidad conjunta de las variables y el cálculo de las probabilidades a partir de su estructura. Las definiciones, conceptos y propiedades básicas de las redes Bayesianas se detallan en [Pearl, 1988, Pearl, 2009]. Una presentación formal de los modelos basados en la teoría de gráficos se pueden consultar en [Castillo et al., 2012], [Koller and Friedman, 2009] y [Murphy, 2012].

El proceso de estimación de una red Bayesiana consiste de una etapa de aprendizaje estructural y una etapa de aprendizaje paramétrico [Scutari and Strimmer, 2010]. La primera etapa consiste en obtener la estructura de la red y la segunda estima los parámetros de las funciones de probabilidad condicional. La estimación de la estructura basada en restricciones se introduce en [Neapolitan et al., 2004] y [Edwards, 2000].

Adicionalmente, la estimación de la estructura basada en scores se introduce en [Korb and Nicholson, 2004] y [Castillo et al., 2012].

Las redes Bayesianas discretas son el tipo más común estudiadas en la literatura: [Pearl, 1988] y [Castillo et al., 2012]. La estimación de parámetros se menciona en [Koller and Friedman, 2009] y [Neapolitan et al., 2004], mientras que la estimación de la estructura en [Korb and Nicholson, 2004] y [Murphy, 2012].

Las definiciones y propiedades básicas de las redes Bayesianas Gaussianas se cubren en [Koller and Friedman, 2009]. Los métodos de estimación basados en restricciones se desarrollan en [Korb and Nicholson, 2004], la estimación de paramétros y el proceso de inferencia en [Neapolitan et al., 2004].

Las redes Bayesianas mixtas o híbridas no se cubren en los libros para modelos gráficos debido a su complejidad en comparación con las redes Bayesianas discretas y Gaussinas, sin embargo pueden ser consultadas en [Scutari and Brogini, 2012]. El caso particular de las redes lineales Gaussianas condicionales, que combinan las variables discretas y continuas utilizando una mezcla de distribuciones normales, se explora en [Koller and Friedman, 2009] y [Koski and Noble, 2011].

1.2.2. Clasificación

El proceso de clasificación es una de las tareas básicas dentro de las técnicas de análisis de datos y reconocimiento de patrones [Heckerman et al., 1995]. Se requiere la construcción de un clasificador, es decir, una función que asigna una categoría a las observaciones de acuerdo a los valores que tomen sus variables predictoras de interés [Friedman et al., 1997]. El proceso de clasificación se dice que es supervisado cuando el clasificador se obtiene a partir un conjunto de observaciones previamente clasificada. Existen muchas técnicas que abordan la tarea de clasificación como los árboles de decisión, las redes neuronales y las redes Bayesianas.

Los árboles de decisión, ampliamente usados en el proceso de aprendizaje automático, tienen como representantes a CART [Breiman, 1984], ID3 [Quinlan, 1986], OCI [Murthy et al., 1994], C4.5 o C5.0 [Quinlan, 1993]. Un árbol de clasificación está formado por un conjunto de nodos, ramas y hojas. En cada nodo se toman decisiones usando los valores que toma una variable en particular. El nodo inicial es llamado nodo raíz y los nodos terminales, también llamados hojas, son aquellos en los que se

predice la clase a la que pertenece cada observación. Una de las grandes ventajas de los árboles de clasificación esta en la interpretabilidad de los resultados.

El método de los $k-$vecinos más cercanos es también uno de los algoritmos de clasificación de más sencilla comprensión [Verma and Pearl, 1991]. La idea básica consiste en clasificar una observación en la clase a la que pertenecen sus k vecinos más cercanos usando medidas de similitud, distancias, modelos de regresión, etc. Este método admite muchas variantes de acuerdo a la distancia que use, la ponderación asignada en cada variable o por la forma en la que se elige la clase a predecir a partir de los $k-$vecinos más cercanos identificados. Este método puede ser computacionalmente costoso para grandes conjuntos de datos.

Las redes neuronales definen una estructura en la que se interconectan los diferentes elementos del proceso que conforman la red, que son llamados neuronas artificiales [Bishop et al., 1995]. Los elementos del proceso se organizan usando una secuencia de capas con un determinado patrón de interconexión entre los elementos pertenecientes a capas distintas. Una de las principales características que tienen estas redes neuronales es responder a los estímulos del entorno mediante un proceso de aprendizaje por el cual se van adaptando los pesos de las conexiones de sus elementos. Dentro de los métodos de clasificación supervisada una de las estructuras más utilizada es la perceptrón multicapa [Suzuki, 1999].

Las redes Bayesianas son particularmente útiles en el proceso de clasificación supervisada [Schäfer and Strimmer, 2005], [Cheng and Greiner, 1999] ya que permiten simplificar la distribución conjunta de las variables predictoras usando distribuciones locales a partir de las relaciones de dependencia e independencia condicional existentes entre las variables. Las redes Bayesianas permiten representar la estructura probabilística de las variables predictoras que se puede utilizar para predecir la clase de pertenencia de cada observación usando el teorema de Bayes [Nagarajan et al., 2013] y [Koski and Noble, 2011].

1.2.3. Clasificadores por redes Bayesianas

Los clasificadores por redes Bayesianas se construyen sobre la estructura definida por las variables predictoras, a la cual se le agrega la variable que tiene las clases o categorías de interés. La estructura de la red correspondiente a las variables predictoras es la que define los diferentes tipos de clasificadores. El más simple de los

clasificadores por redes Bayesianas es llamado Naive Bayes [Duda et al., 1973] ya que asume que todas las variables predictoras son condicionalmente independientes dado el valor de la variable de clase. A pesar de tener un supuesto de independencia poco realista, Naive Bayes tiene un buen desempeño predictivo [Dougherty et al., 1995]y [Domingos and Pazzani, 1997].

El clasificador Tree Augmented Network, TAN, [Friedman et al., 1997] se considera una extensión de Naive Bayes ya que tiene una estructura donde las variables predictoras tienen relaciones de dependencia con la variable de clase y a lo más alguna otra variable predictora. El procedimiento para obtener este clasificador esta basado en el algoritmo de [Chow and Liu, 1968] que permite estimar las relaciones de dependencia de las variables predictoras usando el concepto de información mutua condicionada. El trabajo de [Friedman et al., 1997] comparó TAN con Naive Bayes obteniendo resultados similares en la tasa de elementos correctamente clasificados.

El clasificador Bayesiano $k-$dependiente [Sahami, 1996], al igual que TAN, es también una extensión de Naive Bayes. Este clasificador considera que cada variable predictora puede tener relaciones de dependencia directa con otras k variables predictoras, además de la variable de clase. El algoritmo propuesto permite estimar la estructura de la red Bayesiana usando también la información mutua condicionada para establecer las relaciones de dependencia. [Keogh and Pazzani, 1999] obtuvieron mejores resultados con este clasificador en comparación con Naive Bayes conforme aumentaba el valor de k.

Forest Augmented Naive Bayes [Lucas, 2004], FAN, es una modificación del algoritmo TAN ya que permite eliminar aquellos arcos cuya información mutua condicionada es menor a un umbral predefinido y de esta forma se omiten las relaciones irrelevantes entre variables predictoras. Como la estructura resultante no es estrictamente un árbol, este algoritmo es llamado Forest Augmented Naive Bayes ya que en muchas ocasiones se obtiene un gráfico con estructura del tipo bosque. Un estudio realizado por [Jiang et al., 2005] en 36 conjuntos de datos tomados de [Dheeru and Karra Taniskidou, 2017] FAN obtuvo un mejor comportamiento predictivo en 7 y 11 conjuntos de datos en comparación con Naive Bayes y TAN respectivamente.

El algoritmo Attribute Weighted Naive Bayes [Hall, 2007] considera que las variables predictoras más relevantes deberían tener mayor influencia en el proceso de clasificación. Por esta razón se incorporan pesos o ponderaciones a las variables predictoras para aumentar la influencia de aquellas que son altamente predictivas

y a la vez disminuir la influencia de las variables predictoras no relevantes para la tarea de clasificación. La principal ventaja de este método en comparación con otras técnicas es su simplicidad computacional. El clasificador obtenido presenta una mejora en el rendimiento predictivo en comparación con el clasificador Naive Bayes [Taheri et al., 2014].

Hidden Naive Bayes [Koc et al., 2012], HNB, es también una extensión del clasificador Naive Bayes. Se basa en la creación de un nivel adicional que tiene variables ocultas cuyas relaciones de dependencia sobre las otras variables predictoras se determina usando la información mutua condicional. En el estudio realizado por [Zhang et al., 2005], HNB superó a Naive Bayes y TAN en 36 conjuntos de datos tomados de [Dheeru and Karra Taniskidou, 2017].

El clasificador Escalable Bayesiano [Martınez et al., 2016], KDB, propone un algoritmo que selecciona un submodelo a partir del clasificador KDB completo usando solamente un paso adicional sobre la data de entrenamiento. En ese paso adicional el algoritmo selecciona un subconjunto de variables predictoras y la estructura de la red. Se evaluó el clasificador KDB usando 16 conjuntos de datos grandes obteniendo un comportamiento predictivo muy competitivo en el tiempo de estimación y predicción en comparación con otros clasificadores.

En el proceso de construcción de estructuras por redes Bayesianas es común realizar el proceso de discretización que consiste en transformar los valores de una variable continua en un conjunto de valores discretos. Un algoritmo de discretización divide el conjunto de valores que toma la variable continua en un número finito de intervalos disjuntos y luego se asigna a cada observación el valor correspondiente al intervalo que lo contiene [Dougherty et al., 1995]. Los métodos de discretización se dividen en:

- **Métodos Locales o Globales:** los métodos globales utilizan todas las observaciones para el proceso de discretización a diferencia de los métodos locales que sólo utilizan un subconjunto de las mismas.
- **Métodos Supervisados o no Supervisados:** los métodos de discretización supervisados utilizan la información de la categoría a la que pertenece cada observación mientras que los métodos no supervisados omiten esta información.
- **Métodos Estáticos o Dinámicos:** los métodos estáticos son aquellos que discretizan cada variable de forma independiente de las demás. Los métodos dinámicos son aquellos que discretizan todas las variables simultáneamente

tratando de utilizar las dependencias existentes entre ellas.

- **Métodos Top-down o Bottom-up:** los métodos top-down comienzan con una lista vacía de puntos de corte que se van agregando conforme se realiza el proceso de discretización. Los métodos bottom-up comienzan con una lista completa de todos los valores continuos de la variable como puntos de corte que se van eliminando en cada paso del proceso de discretización.

Los métodos de discretización no supervisados más sencillos dividen el rango de cada variable predictora en k intervalos de igual longitud o con igual cantidad de datos. El método 1R, desarrollado por [Holte, 1993], es un método de discretización supervisada que divide el conjunto de datos en intervalos disjuntos cuyos limites se determinan en base a la clase mayoritaria en los intervalos adyacentes. El proceso de discretización por mínima entropía es un método top-down supervisado propuesto por [Fayyad and Irani, 1993]. Este método selecciona recursivamente los puntos de corte mediante un algoritmo de minimización de la entropía usando el criterio de Longitud de Descripción Mínima como criterio de parada. El método de discretización bottom-up supervisado ChiMerge propuesto por [Kerber, 1992] usa el estadístico de prueba χ^2 para determinar si las frecuencias relativas en intervalos adyacentes son los suficientemente similares para justificar la fusión de ellos en un solo intervalo.

1.3. Caracterización y naturaleza del objeto de la investigación

El trabajo de investigación presentado es de naturaleza cuantitativa, ya que las relaciones de dependencia entre las variables predictoras se establecen a partir de pruebas de hipótesis de independencia condicional y las estructuras por redes Bayesianas se comparan usando el criterio de información Bayesiano, BIC. El trabajo es también de naturaleza predictiva ya que los clasificadores propuestos permiten establecer la clase de pertenencia de las observaciones de acuerdo a las probabilidades estimadas a partir de la estructura obtenida inicialmente. Finalmente el trabajo es de naturaleza aplicada ya que los clasificadores serán construidos, evaluados y comparados usando conjuntos de datos, obtenidos a partir del repositorio UCI Machine Learning [Dheeru and Karra Taniskidou, 2017], considerando también el proceso de selección de variables a través del algoritmo SES.

1.4. Formulación del problema

En Estadística es importante conocer las relaciones existentes entre el conjunto de variables predictoras utilizadas en una tarea específica. Sin embargo, muchos de los modelos usados en el proceso de clasificación consideran que las variables predictoras son independientes entre sí, lo cual es en realidad una suposición poco realista.

En general, las variables predictoras presentan relaciones de dependencia, por lo que resulta de interés conocer su comportamiento probabilístico a partir de la función de densidad conjunta. En ese sentido, las redes Bayesianas pueden ser de utilidad para poder estimar la estructura de las dependencias existentes.

El problema de investigación consiste en construir un clasificador a partir de un conjunto de variables predictoras considerando las relaciones de dependencia e independencia condicional existentes entre ellas. En la actualidad se han desarrollado nuevos algoritmos de estimación de la estructura de una red Bayesiana: basados en restricciones, en scores e híbridos. Sobre estas estructuras se pueden construir clasificadores que pueden ser comparados con los clasificadores tradicionales como Naive Bayes y TAN. Además, los conjuntos de datos suelen tener una cantidad importante de variables predictoras, por lo que es necesario realizar el proceso de selección de las variables que permitan alcanzar el mayor poder predictivo antes de construir los clasificadores.

1.5. Formulación de las hipótesis

1. Los nuevos algoritmos de estimación de la estructura de una red Bayesiana basados en restricciones, scores e híbridos permiten identificar las relaciones de dependencia e independencia condicional existentes entre las variables predictoras.

2. Los clasificadores construidos sobre las estructuras de red obtenidas por estos algoritmos permiten obtener mejores tasas de precisión que las obtenidas con los clasificadores tradicionales Naive Bayes y TAN.

3. El algoritmo de selección de variables Statiscally Equivalent Signatures permite encontrar clasificadores con mayor poder predictivo en comparación con los

clasificadores construidos con todas las variables predictoras.

1.6. Formulación de los objetivos de la investigación

1. Presentar las metodologías de estimación de la estructura de una red Bayesiana usando los algoritmos basados en restricciones, en scores e híbridos que permiten representar las relaciones de dependencia e independencia condicional entre variables predictoras.
2. Evaluar los clasificadores por redes Bayesianas de acuerdo a los algoritmos mencionados, comparando su tasa de precisión con los clasificadores tradicionales Naive Bayes y TAN.
3. Aplicar el algoritmo de selección de variables Statiscally Equivalent Signatures, comparando el poder predictivo de los clasificadores obtenidos antes y después de su aplicación.

1.7. Importancia y justificación de la investigación

El problema de clasificación consiste en establecer una regla que permita asignar un elemento a una de las k clases de una variable respuesta Y. Las relaciones de dependencia entre las variables predictoras permiten obtener las probabilidades de asignación de los elementos a cada una de las categorías definidas por la variable Y. El proceso de clasificación se realiza usando el teorema de Bayes para asignar una observación en la categoría con mayor probabilidad, [Murphy, 2012].

Las redes Bayesianas permiten conocer y describir la estructura de la función de probabilidad conjunta de las variables predictoras a partir de un conjunto de métodos desarrollados durante los últimos años, tales como los algoritmos basados en restricciones, basados en scores y los híbridos, los cuales podrían utilizarse para construir clasificadores cuyo rendimiento predictivo pueda compararse con los clasificadores tradicionales.

Dentro del área conocida como Machine Learning, el proceso de selección de variables se considera una de las primeras tareas a realizar antes de la aplicación de algún

algoritmo o procedimiento. En el contexto de las redes Bayesianas este proceso permite seleccionar las variables predictoras importantes usadas para la construcción del clasificador. De esta forma se pueden obtener estructuras más sencillas que podrían tener un rendimiento predictivo mayor en comparación con los clasificadores obtenidos a partir de todas las variables predictoras.

Capítulo 2

Marco Teórico

2.1. Fundamentos Filosóficos Teóricos de la Investigación

Una de las ventajas más importantes de las redes Bayesianas es que pueden representar tanto el aspecto cualitativo y cualitativo de un problema. El soporte teórico del aspecto cualitativo en las redes Bayesianas lo aporta la teoría de gráficos. Por otro lado hay tres elementos que caracterizan el aspecto cuantitativo de una red Bayesiana: el concepto de probabilidad como un grado de creencia subjetiva relativa a la ocurrencia de un evento, el Teorema de Bayes y un conjunto de funciones de probabilidad condicional [Neapolitan et al., 2004], [Koller and Friedman, 2009], [Korb and Nicholson, 2004].

Thomas Bayes, Londres 1702-1761, fue teólogo, matemático y miembro de la Royal Society desde 1742. En 1763 se publicó póstumamente "Essay Towards Solving a Problem in the Doctrine of Chances", donde el reverendo Bayes enuncia el teorema que lleva su nombre. Este trabajo fue entregado a la Royal Society por Richard Price y resulta ser la base para el desarrollo de la estadística Bayesiana.

En su forma básica, el teorema de Bayes es una expresión que relaciona probabilidades condicionales. Sean A y B dos eventos con $\Pr(A) > 0$. Entonces:

$$\Pr(B|A) = \frac{\Pr(A|B)\Pr(B)}{\Pr(A)} \tag{2.1.1}$$

El uso principal de este teorema es revertir el condicionamiento de los eventos, esto es, mostrar cómo la probabilidad de $B|A$ está relacionada con la probabilidad de $A|B$.

El teorema de Bayes es la regla básica de actualización de probabilidades en una red Bayesiana. Es una herramienta de cálculo útil cuando se requiere calcular probabilidades asociadas a un modelo del que recibimos evidencias sobre alguna de las variables predictoras implicadas. Lo que hace una red Bayesiana, a grandes rasgos, es actualizar probabilidades teniendo en cuenta los principios de independencia condicional.

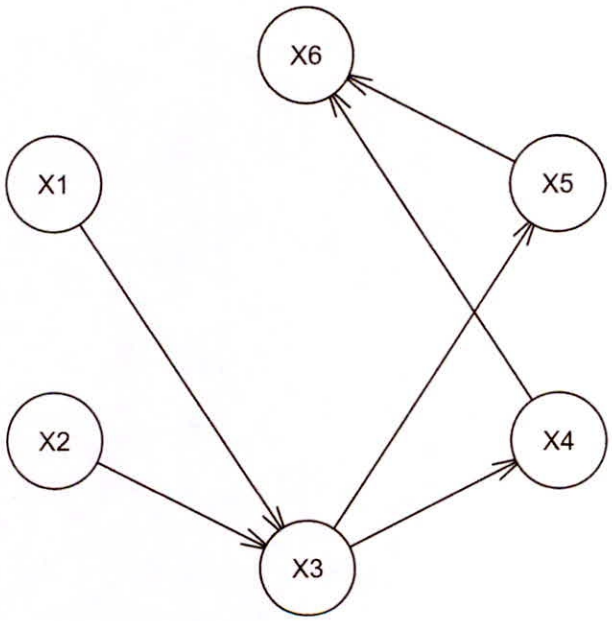


Figura 2.1: Estructura de una red Bayesiana

La Figura 2.1 muestra una red Bayesiana con 6 variables aleatorias X_1, X_2, X_3, X_4, X_5 y X_6 cada una representada por un nodo en el gráfico. Las relaciones de dependencia directa se representan como arcos entre pares de variables (es decir, $X_1 \longrightarrow X_3$ significa que X_3 depende de X_1). El nodo de donde parte el arco se llama

padre, mientras que el nodo hacia donde se dirige se llama hijo. Las relaciones de dependencia indirectas no están representadas de forma explícita. Sin embargo, pueden ser leídas desde la estructura como secuencias de arcos que conducen de una variable hacia otra a través de una o más variables mediadoras (es decir, la combinación de $X_1 \longrightarrow X_3$ y $X_3 \longrightarrow X_5$ significa que X_5 depende de X_1 a través de X_3). Tales secuencias de arcos se dice que forman un camino que conduce de una variable a otra. Los caminos de la forma $X_1 \longrightarrow \cdots \longrightarrow X_1$, conocidos como ciclos, no están permitidos. Por esta razón, los gráficos utilizados en redes Bayesianas se denominan gráficos acíclicos dirigidos.

Una red Bayesiana necesita un conjunto de funciones de probabilidad condicional, una por cada variable o nodo en la red, sobre los que ha de aplicarse la regla de Bayes. Es decir, cada variable de la red está caracterizada por una función de probabilidad condicional donde se representan los valores que puede tomar esa variable en función de los valores que toman el conjunto de variables de las que depende [Friedman et al., 1997].

La estructura de una red Bayesiana puede utilizarse para construir clasificadores identificando uno de los nodos con la variable de clase. El proceso de clasificación se realiza usando el teorema de Bayes para asignar una observación en la clase con mayor probabilidad. Sin embargo, el procedimiento anterior requiere el uso de la función de probabilidad conjunta de las variables involucradas. Una red Bayesiana permite representar esta función de probabilidad conjunta usando funciones de probabilidad definidas en menos variables y por consiguiente más sencillas. Esto permite simplificar el cálculo de las probabilidades de asignación [Nagarajan et al., 2013], [Koski and Noble, 2011].

2.2. Marco Conceptual

2.2.1. Gráficos, nodos y arcos

Un gráfico $G = (\mathbf{V}, A)$ consiste de un conjunto no vacío de nodos o vértices $\mathbf{V}$ y de un conjunto finito, pero posiblemente vacío, de pares de vértices A llamados arcos. Cada arco $a = (u, v)$ se define como un par ordenado, o no ordenado, de nodos que se encuentran adyacentes uno del otro, por lo que u y v son llamados vecinos. Si (u, v) es

un par ordenado entonces u es llamado **padre** y v es llamado hijo, en este caso el arco es dirigido y se representa por $(u \longrightarrow v)$. Si $(u.v)$ es un par no ordenado entonces se dice que el arco es no dirigido, es denotado por $e \in E$ y se representa por $(u - v)$.

Se dice que un gráfico es dirigido (denotado por $G = (\mathbf{V}, A)$) si todos sus arcos son dirigidos, gráfico no dirigido (denotado por $G = (\mathbf{V}, E)$) si todos sus arcos son no dirigidos y gráfico parcialmente dirigido (denotado por $G = (\mathbf{V}, A, E)$) si contiene arcos dirigidos y también no dirigidos.

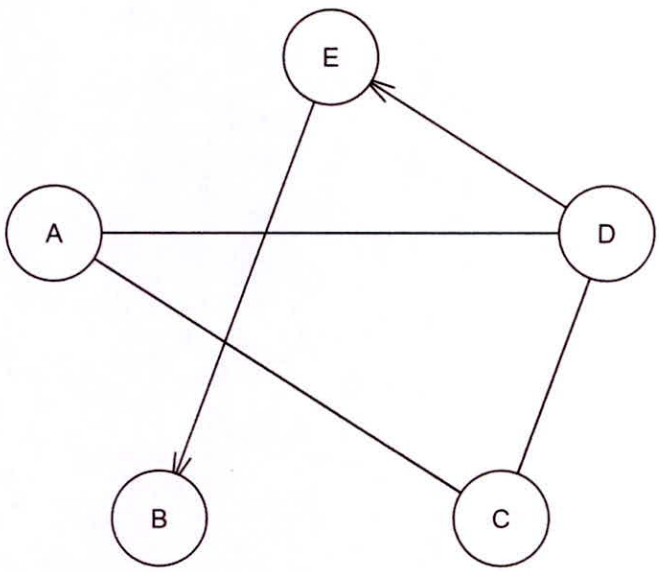


Figura 2.2: Gráfico parcialmente dirigido

2.2.2. La estructura de un gráfico

La estructura más simple es un gráfico vacío, es decir un gráfico que no tiene arcos. Por otro lado, un gráfico saturado es aquel donde todos los arcos se encuentran

conectados unos con otros. La estructura de un gráfico puede revelar propiedades estadísticas importantes. Una de las más importantes esta relacionada con los caminos. Los caminos son secuencias de arcos $(v_1, v_2, \cdots, v_n)$ que conectan pares de nodos. En un gráfico dirigido se asume que todos los arcos en un camino siguen la misma dirección, partiendo desde v_1 y finalizando en v_n. En un gráfico no dirigido y parcialmente dirigido los arcos en un camino pueden ser no dirigidos o seguir cualquier dirección. Los caminos en los que $v_1 = v_n$ son llamados ciclos.

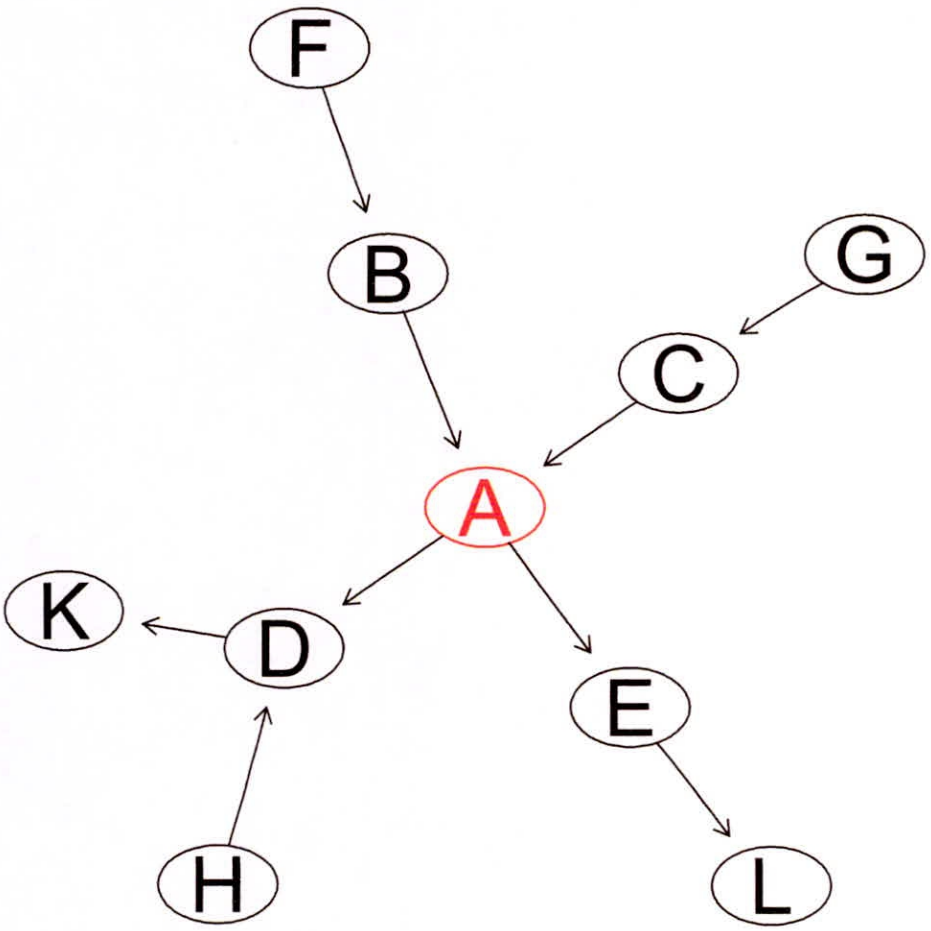


Figura 2.3: Padres, hijos, antepasados y descendentes del nodo A

La estructura de un gráfico dirigido define un ordenamiento parcial de los nodos solo si el gráfico es acíclico. El ordenamiento es inducido por la dirección de los arcos y se define de la siguiente forma: si el nodo v_i precede al nodo v_j, entonces no existe nodo que vaya desde v_j hacia v_i. De acuerdo a esta definición el primer nodo es llamado nodo raíz, aquel que no tiene arcos entrantes, y el último llamado nodo hoja, aquel que tiene al menos un arco entrante pero ninguno que sale de él. Además, si existe

un camino ordenado desde v_i hacia v_j, v_i es llamado un antepasado de v_j y v_j es llamado un descendiente de v_i. Si el camino esta formado por un solo arco entonces v_i es padre de v_j y v_j es hijo de v_i.

En la Figura 2.3, con respecto al nodo A, los antepasados y descendientes son $\{B, C. F. G\}$ y $\{D, E, K, L\}$ respectivamente, mientras que los padres e hijos respectivos son $\{B, C\}$ y $\{D, E\}$. Por otro lado, los vecinos del nodo A están formados por sus padres e hijos.

2.3. Redes Bayesianas

Las redes Bayesianas son una clase de modelos gráficos, que permiten una representación concisa de la estructura probabilística de datos multivariantes utilizando gráficos. Las redes Bayesianas están formadas por:

- Un conjunto de variables aleatorias $\mathbf{X} = \{X_1, X_2, \cdots, X_p\}$ que describen las variables de interés. La distribución de probabilidad multivariante de $\mathbf{X}$ se llama la distribución global de los datos, mientras que las que se asociaron con cada $X_i \in \mathbf{X}$ se llaman distribuciones locales.
- Un gráfico acíclico dirigido, GAD, denotado por $G = (\mathbf{V}, A)$. Cada nodo $v \in \mathbf{V}$ está asociada con una variable X_i. Los arcos dirigidos $a \in A$, que los conectan representan dependencias probabilísticas directas. Si no existe un arco que conecta dos nodos de las correspondientes variables entonces son independientes o condicionalmente independientes dado un subconjunto de las variables restantes.

Definición 2.3.1 Una red Bayesiana $\mathfrak{B} = (G, \mathbf{X})$ esta compuesta por un gráfico acíclico dirigido $G = (\mathbf{X}, A)$ y un vector aleatorio $\mathbf{X}$, cuya distribución global puede factorizarse de acuerdo a la estructura del GAD.

En la red Bayesiana de la Figura 2.1 la distribución global de $\mathbf{X} = \{X_1, X_2, \cdots, X_6\}$ puede ser factorizada de la siguiente manera:

$$\Pr(\mathbf{X}) = \Pr(X_1)\Pr(X_2)\Pr(X_3|X_1, X_2)\Pr(X_4|X_3)\Pr(X_5|X_3)\Pr(X_6|X_4, X_5) \qquad (2.3.1)$$

2.3.1. Conexiones fundamentales

Todas las posibles configuraciones con tres nodos y dos arcos forman estructuras simples conocidas como conexiones fundamentales y son los componentes básicos para las propiedades gráficas y probabilísticas de una red Bayesiana. En la Figura 2.1 se tienen:

- Estructuras de la forma: $X_2 \rightarrow X_3 \rightarrow X_5$ que se conocen como conexiones en serie, ya que los dos arcos tienen la misma dirección y se suceden uno tras otro.
- Estructuras de la forma: $X_5 \leftarrow X_3 \rightarrow X_4$ que se conocen como conexiones divergentes, ya que los dos arcos tienen direcciones divergentes desde un nodo central.
- Estructuras de la forma: $X_1 \rightarrow X_3 \leftarrow X_2$ que se conocen como conexiones convergentes, porque los dos arcos convergen a un nodo central. Cuando no hay arco que une los dos padres, es decir ni $X_1 \rightarrow X_2$ ni $X_1 \leftarrow X_2$, las conexiones convergentes se llaman $v-$estructuras.

2.3.2. Independencia condicional y separación gráfica

Las relaciones directas e indirectas entre dos variables se pueden leer desde el GAD comprobando si están conectadas de alguna manera. Si las variables dependen directamente entre sí, habrá un solo arco que conecta sus nodos. Si la dependencia es indirecta, habrá dos o más arcos que pasan a través de los nodos que median la asociación. En general, dos conjuntos de variables $\mathbf{X}$ y $\mathbf{Y}$ son independientes dado un tercer conjunto de variables $\mathbf{Z}$ si no existe un conjunto de arcos que los conecta que no esté bloqueado por las variables condicionantes. En otras palabras, se dice que $\mathbf{X}$ y $\mathbf{Y}$ están $d-$separados por $\mathbf{Z}$ y se denota por $\mathbf{X} \perp\!\!\!\perp_G \mathbf{Y}|\mathbf{Z}$.

El enlace entre la separación gráfica, denotada por $\perp\!\!\!\perp_G$, y la independencia probabilística, denotada por $\perp\!\!\!\perp_P$, proporciona una forma directa para expresar las relaciones de dependencia entre las variables.

Definición 2.3.2 Sea M la estructura de dependencia de la distribución de probabilidad *P* de $\mathbf{X}$, es decir, el conjunto de relaciones de independencia condicional que enlazan a cualquier triplete $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ de subconjuntos de $\mathbf{X}$. Un gráfico G es un mapa de dependencia, o $D-$mapa, de *M* si existe una correspondencia uno a uno entre

las variables aleatorias en $\mathbf{X}$ y los nodos $\mathbf{V}$ de G tal que para todos los subconjuntos disjuntos $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ de $\mathbf{X}$ se cumple:

$$\mathbf{A} \perp\!\!\!\perp_P \mathbf{B}|\mathbf{C} \Rightarrow \mathbf{A} \perp\!\!\!\perp_G \mathbf{B}|\mathbf{C} \tag{2.3.2}$$

Del mismo modo, *G* es un mapa de independencia, o $I-$mapa, de *M* si:

$$\mathbf{A} \perp\!\!\!\perp_P \mathbf{B}|\mathbf{C} \Leftarrow \mathbf{A} \perp\!\!\!\perp_G \mathbf{B}|\mathbf{C} \tag{2.3.3}$$

Se dice que G es un mapa perfecto de M si es tanto un $D-$mapa y un $I-$mapa, es decir:

$$\mathbf{A} \perp\!\!\!\perp_P \mathbf{B}|\mathbf{C} \Longleftrightarrow \mathbf{A} \perp\!\!\!\perp_G \mathbf{B}|\mathbf{C} \tag{2.3.4}$$

y en este caso se dice que G es fiel o isomorfo a *M*.

En el caso de un $D-$mapa, la distribución de probabilidad de $\mathbf{X}$ determina que arcos están presentes en G. Los nodos que están conectados corresponden a variables dependientes en $\mathbf{X}$; sin embargo, los nodos que están separados no necesariamente corresponden a las variables condicionalmente independientes. Por otra parte, en el caso de un $I-$mapa se tiene que los arcos presentes en G determinan qué variables son condicionalmente independientes en $\mathbf{X}$. Por lo tanto, los nodos que se encuentran separados corresponden a las variables condicionalmente independiente, pero los nodos que están conectados en G no corresponden necesariamente a las variables dependientes en $\mathbf{X}$. En el caso de un mapa perfecto, hay una correspondencia uno a uno entre la separación gráfica en G y la independencia condicional en $\mathbf{X}$. La correspondencia entre la estructura de G y las relaciones de dependencia condicional que ésta representa se establece con el criterio de la $d-$separación [Pearl, 1988].

Definición 2.3.3 Si $\mathbf{A}$, $\mathbf{B}$ y $\mathbf{C}$ son tres subconjuntos disjuntos de nodos en un GAD G, entonces $\mathbf{C}$ se dice que es $d-$separado de $\mathbf{A}$ desde $\mathbf{B}$, denotado por $\mathbf{A} \perp\!\!\!\perp_P \mathbf{B}|\mathbf{C}$, si

a lo largo de cada secuencia de arcos entre un nodo en $\mathbf{A}$ y un nodo en $\mathbf{B}$ existe un nodo D que satisface una de las dos condiciones siguientes:

1. D tiene arcos convergentes, además D y ninguno de sus descendientes están en $\mathbf{C}$.
2. D está en $\mathbf{C}$ y no tiene arcos convergentes.

La propiedad de Markov de las redes Bayesianas, que se obtiene directamente de la $d-$separación, permite la representación de la distribución de probabilidad conjunta de las variables aleatorias en $\mathbf{X}$ como el producto de las distribuciones locales asociadas con cada variable X_i. Lo anterior es una aplicación directa de la regla de la cadena [Korb and Nicholson, 2010]:

$$\Pr(\mathbf{X}) = \prod_{i=1}^{p} \Pr(X_i|\Pi_{X_i}) \tag{2.3.5}$$

donde Π_{X_i} es el conjunto de padres de X_i.

2.3.3. Estructuras Equivalentes

A partir de la Figura 2.1 es posible notar que las conexiones serial y divergente tienen como resultado la misma factorización ya que cualquiera puede ser obtenida a partir de la otra aplicando repetidamente el teorema de Bayes. Cada estructura equivalente en probabilidad es conocida como estructuras equivalentes de Markov. Como la equivalencia es simétrica, reflexiva y transitiva, cada conjunto de estructuras equivalentes forma una clase de equivalencia.

2.4. El manto de Markov

La descomposición de la distribución global en la Ecuación 2.3.1 proporciona una manera conveniente para dividir $\mathbf{X}$ en partes manejables, e identifica en los padres de cada nodo el conjunto de variables de condicionamiento en cada distribución local. El manto de Markov [Pearl, 1988] representa el conjunto de nodos que $d-$separan completamente un nodo especifico del resto de nodos en el gráfico.

Definición 2.4.1 El manto de Markov de un nodo $A \in \mathbf{V}$ es el mínimo subconjunto $\mathbf{S}$ de $\mathbf{V}$ tal que:

$$A \perp\!\!\!\perp_G \mathbf{V} - \mathbf{S} - A|\mathbf{S} \tag{2.4.1}$$

Suponiendo que G es fiel, se tiene que $\mathbf{S}$ es el subconjunto mínimo de $\mathbf{V}$ tal que:

$$A \perp\!\!\!\perp_P \mathbf{V} - \mathbf{S} - A|\mathbf{S} \tag{2.4.2}$$

y corresponde al subconjunto de nodos que hace que el resto sea redundante al realizar el proceso de inferencia en un nodo dado. En otras palabras, se considera la identificación del manto de Markov como un problema de selección de variables.

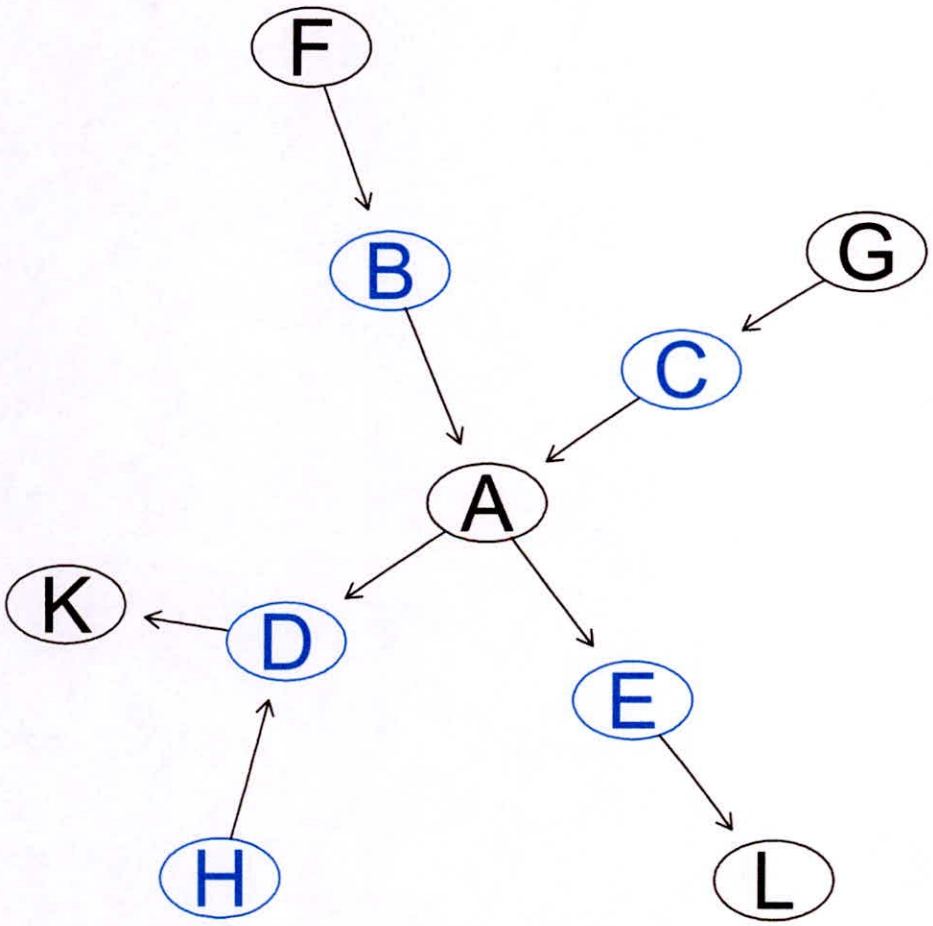


Figura 2.4: Manto de Markov del nodo A

En general, el manto de Markov de un nodo A es el conjunto formado por los padres de A, los hijos de A y todos los demás nodos que comparten un hijo con A. Si el nodo A está en el manto de Markov de B, entonces B está en el manto de Markov de A.

2.5. Estimación de una red Bayesiana

En el campo de las redes Bayesianas, la selección y estimación del modelo se conocen como aprendizaje, un nombre tomado de la inteligencia artificial y Machine Learning. El aprendizaje de la red Bayesiana se realiza como un proceso de dos pasos:

1. La estimación de la estructura del GAD.
2. La estimación de los parámetros de las distribuciones locales definidas por la estructura del GAD obtenido en el paso anterior.

Ambos pasos se pueden realizar utilizando la información proveniente de un conjunto de datos o entrevistando a expertos en los campos relevantes para el fenómeno que se está modelando. La combinación de ambos enfoques es también común. A menudo, la información previa disponible sobre el fenómeno no es suficiente para que un experto pueda especificar completamente una red Bayesiana. Muchas veces es casi imposible especificar la estructura del GAD cuando se tiene un gran número de variables involucradas, por ejemplo, para el análisis de red de genes.

Considere un conjunto de datos D y una red Bayesiana $B = (G, \mathbf{X})$. Si Θ denota los parámetros de la distribución global de $\mathbf{X}$, se puede suponer que Θ identifica de forma exclusiva $\mathbf{X}$ en la familia paramétrica de distribuciones escogidas para el modelado de D y es posible escribir $B = (G, \Theta)$. El aprendizaje de la red Bayesiana se puede formalizar como:

$$\underbrace{\Pr(B|D) = \Pr(G, \Theta|D)}_{\text{learning}} = \underbrace{\Pr(G|D)\Pr(\Theta|G, D)}_{\text{estim estruc}\quad\text{estim parám}} \tag{2.5.1}$$

La descomposición de $\Pr(G, \Theta|D)$ refleja los dos pasos descritos anteriormente. La estimación de la estructura se puede realizar en la práctica por la búsqueda del GAD G que maximiza:

$$\Pr(G|D) \propto \Pr(G)\Pr(D|G) = \Pr(G)\int \Pr(D|G, \Theta)\Pr(\Theta|G)d\Theta \tag{2.5.2}$$

usando el teorema de Bayes para descomponer la probabilidad posterior del GAD, es decir $\Pr(G|D)$, en el producto de la distribución a priori sobre los posibles GAD, es decir $\Pr(G)$, y la probabilidad de la data, es decir $\Pr(D|G)$. Claramente, no es posible calcular esta última sin también estimar los parámetros Θ de *G*. Por lo tanto, Θ tiene que estar integrada de la Ecuación (2.5.2) para hacer $\Pr(G|D)$ independiente de cualquier elección específica de Θ.

La distribución a priori $\Pr(G)$ proporciona una forma ideal para introducir información previa disponible sobre las relaciones de independencia condicional entre las variables en $\mathbf{X}$. Se puede, por ejemplo, considerar que uno o más arcos estén presentes o ausentes en el GAD, de acuerdo a los conocimientos adquiridos en los estudios anteriores. Es posible considerar también que algunos arcos, si están presentes en el GAD, estén orientados en una dirección específica cuando esa dirección es la única que tiene sentido a la luz de la lógica que sustenta el fenómeno que se está modelando.

La elección más común para $\Pr(G)$ es una a priori no informativa sobre el espacio de los posibles GAD, asignando la misma probabilidad a cada estructura. Algunos GAD pueden ser excluidos debido a la información a priori discutida anteriormente. Las distribuciones a priori complejas, llamadas a priori estructurales, también son posibles pero rara vez se utilizan en la práctica por dos razones. En primer lugar, el uso de la distribución de probabilidad uniforme hace que $\Pr(G)$ no afecte la maximización de $\Pr(G|D)$ lo cual es conveniente por razones computacionales y algebraicas. En segundo lugar, el número de posibles GAD aumenta super-exponencial en el número de nodos. En un GAD con p nodos tenemos $p(p-1)/2$ posibles arcos, dados por las parejas de diferentes nodos en $\mathbf{V}$. La especificación de una distribución a priori compleja sobre un gran número de GADs es una tarea difícil, incluso para conjuntos de datos pequeños.

El cálculo de $\Pr(D|G)$ es también problemático desde el punto de vista computacional y algebraico. A partir de la descomposición en las distribuciones locales, podemos continuar factorizando $\Pr(D|G)$ de una manera similar:

$$\begin{aligned}
\Pr(D|G) &= \int \prod_{i=1}^{p} [\Pr(X_i|\Pi_{X_i}, \Theta_{X_i}) \Pr(\Theta_{X_i}|\Pi_{X_i})] d\Theta \\
&= \prod_{i=1}^{p} \int [\Pr(X_i|\Pi_{X_i}, \Theta_{X_i}) \Pr(\Theta_{X_i}|\Pi_{X_i})] d\Theta_{X_i} \\
&= \prod_{i\ \ 1}^{p} E_{\Theta_{X_i}}[\Pr(X_i|\Pi_{X_i})] \qquad (2.5.3)
\end{aligned}$$

Las funciones que se pueden factorizar de esta manera se llaman descomponibles. Si todos los esperados se pueden calcular en forma cerrada $\Pr(D|G)$ se puede calcular en un tiempo razonable, incluso para grandes conjuntos de datos. Esto es posible tanto para la distribución multinomial, usada para redes Bayesianas discretas a través de su conjugada posterior Dirichlet, y para la distribución Gaussiana multivariante usada para redes Bayesianas continuas, a través de su distribución conjugada Wishart Inversa. Para redes Bayesianas discretas, $\Pr(D|G)$ puede ser estimada con la puntuación Bayesiana uniforme equivalente de Dirichlet, BDe, [Heckerman et al., 1995] que sume una distribución a priori plana sobre el espacio del GAD y el espacio paramétrico de cada nodo:

$$\Pr(G) \propto 1 \quad y \quad \Pr(\Theta_{X_i}|\Pi_{X_i}) = \alpha_{ij} = \frac{\alpha}{|\Theta_{X_i}|} \tag{2.5.4}$$

El único parámetro en BDe es el tamaño de muestra imaginario α asociado a la Dirichlet, que determina el peso que se asigna a la distribución a priori como el tamaño de una muestra imaginaria que lo soporta. Bajo estos supuestos, la BDe toma la forma:

$$\begin{aligned} \mathrm{BDe}(G, D) &= \prod_{i=1}^{p} \mathrm{BDe}(X_i|\Pi_{X_i}) \\ &= \prod_{i=1}^{p} \prod_{j=1}^{q_i} \left\{ \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + \alpha_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ij} + n_{ijk})}{\Gamma(\alpha_{ijk})} \right\} \end{aligned} \tag{2.5.5}$$

donde:

- *p* es el número de nodos en *G*.
- r_i es el número de categorías para el nodo X_i.
- q_i es el número de configuraciones de las categorías de los padres de X_i.
- n_{ijk} es el número de muestras que tiene la categoría *j* para el nodo X_i y la configuración k para sus padres.

La correspondiente distribución posterior para redes Bayesianas es llamada uniforme Gaussiana equivalente Bayesiana, BGe, [Geiger and Heckerman, 1994]. De manera similar al BDe asume una distribución a priori no informativa sobre el espacio del GAD y el espacio paramétrico de cada nodo siendo su único parámetro es el tamaño de la muestra imaginario α, resultando en una expresión muy complicada.

Como resultado de las dificultades antes mencionadas, se han desarrollado dos

alternativas al uso de $\Pr(D|G)$ en la estimación de la estructura. La primera de ellas es el uso del criterio de información Bayesiano, BIC, con una aproximación de $\Pr(D|G)$, dado por:

$$\mathrm{BIC}(G, D) \quad \rightarrow \quad \log \mathrm{BDe}(G, D) \tag{2.5.6}$$

conforme $n \rightarrow \infty$. El BIC se puede descomponer y sólo depende de la función de verosimilitud:

$$\mathrm{BIC}(G, D) \quad = \quad \sum_{i=1}^{p} \left[\log \Pr(X_i|\Pi_{X_i}) - \frac{|\Theta_{X_i}|}{2} \log n \right] \tag{2.5.7}$$

que hace que sea muy fácil de calcular para redes Bayesianas discretas y redes Bayesianas Gaussianas. La segunda alternativa es utilizar pruebas de independencia condicional para estimar la estructura del GAD.

Una vez estimada la estructura del GAD se pasa al proceso de estimación de los parámetros de $\mathbf{X}$. Asumiendo que los parámetros que provienen de diferentes distribuciones locales son independientes, se necesita estimar sólo los parámetros de una distribución local a la vez. Siguiendo el enfoque Bayesiano se requeriría encontrar el valor de Θ que maximiza $\Pr(\Theta|G, D)$ a través de sus componentes $\Pr(\Theta_{X_i}|X_i|\Pi_{X_i})$.

Las distribuciones locales en la práctica se refieren sólo a un pequeño número de nodos, es decir X_i y sus padres Π_{X_i}. Su dimensión por lo general no es proporcional a la cantidad de nodos en la red Bayesiana, evitando así la llamada maldición de la dimensionalidad. Esto significa que cada distribución local tiene un número relativamente pequeño de parámetros a estimar a partir de la muestra, y las estimaciones son más precisas debido a la mejor relación entre el tamaño de Θ_{X_i} y el tamaño de la muestra.

2.5.1. Estimación de la estructura

Varios algoritmos han sido presentados gracias a la aplicación de los resultados que se derivan de la teoría de la probabilidad, la teoría de la información y la teoría de la optimización. A pesar de la variedad de orígenes teóricos y la terminología todos ellos se pueden resumir en sólo tres algoritmos: basados en restricciones, basados en

scores y los híbridos. Todos estos algoritmos operan bajo un conjunto de supuestos comunes:

- Debe haber una correspondencia uno a uno entre los nodos del GAD y las variables aleatorias $\mathbf{X}$. Esto significa que no debe haber múltiples nodos que sean funciones deterministas de una sola variable.
- Todas las relaciones entre las variables en $\mathbf{X}$ deben ser de independencia condicional, porque son, por definición, el único tipo de relaciones que pueden ser expresados por la red Bayesiana.
- Cada combinación de los posibles valores de las variables en $\mathbf{X}$ debe representar un evento válido y observable. Este supuesto implica una distribución global estrictamente positiva, que es necesario que sea determinada unicamente por el manto de Markov y, por lo tanto, un modelo único identificable. Los algoritmos basados en restricciones funcionan incluso cuando esto no sea verdadero porque la existencia de un mapa perfecto es también una condición suficiente para la unicidad del manto de Markov [Pearl, 1988].

2.5.1.1. Algoritmos basados en restricciones

Los algoritmos basados en restricciones se basan en el trabajo de Pearl, en los mapas y su aplicación a los modelos gráficos causales. Su algoritmo de Causalidad Inductiva [Verma and Pearl, 1991] proporciona un marco para la estimación de la estructura de un GAD usando pruebas de independencia condicional.

Los detalles del algoritmo de Causalidad Inductiva se describen a continuación. El primer paso identifica qué pares de variables están conectados por un arco, independientemente de su dirección. Estas variables no pueden ser independientes dado cualquier otro subconjunto de variables, ya que no pueden ser $d-$separados. Este paso puede ser visto también como un procedimiento de selección hacia atrás a partir de un modelo saturado con un gráfico completo y su poda sobre la base de las pruebas estadísticas de independencia condicional. El segundo paso trata con la identificación de las $v-$estructuras entre todos los pares de nodos no adyacentes *A* y *B* con un vecino común *C*. Por definición, las $v-$estructuras son la única conexión fundamental en la que dos nodos no adyacentes son no independientes condicionados a un tercero. Por lo tanto, si existe un subconjunto de nodos que contiene *C* y D y $d-$separa A y B, los tres nodos son parte de una $v-$estructura centrada en C. Esta condición puede ser

verificada mediante la realización de la prueba de independencia condicional para A y B contra cada posible subconjunto de sus vecinos comunes que incluyen a C. Al final de la segunda etapa, tanto el esqueleto y las $v-$estructuras de la red son conocidos, por lo que la clase de equivalencia a la que pertenece la red Bayesiana se identifica de manera única. El tercer y último paso identifica arcos obligados y los orienta de forma recursiva para obtener información del CPDAG describiendo la clase de equivalencia identificada por los pasos anteriores.

El principal problema del algoritmo de Causalidad Inductiva es que los primeros dos pasos no se pueden aplicar en cualquier conjunto de datos debido al número exponencial de las posibles relaciones de independencia condicional. Esto ha llevado al desarrollo de algoritmos mejorados, tales como Grow-Shrink, GS, [Margaritis, 2003] que permite obtener la estructura la red Bayesiana identificando en el primer paso el manto de Markov de cada nodo, usando las pruebas de independencia condicional. Los pasos del algoritmo Grow-Shrink se presentan a continuación.

Algoritmo 2.1 Algoritmo Grow-Shrink

1. Para cada nodo A en $\mathbf{V}$ se obtiene su manto de Markov $\mathcal{MB}(A)$ verificando la propiedad de simetría.
2. Para cada par de nodos A y B en $\mathbf{V}$ se busca el conjunto $\mathbf{S}_{AB} \subset \mathbf{V}$ tales que A y B son independientes dado $\mathbf{S}_{AB}$ donde $A, B \notin \mathbf{S}_{AB}$. Si no existe tal conjunto, poner un arco no dirigido entre A y B.
3. Para cada par de nodos no adyacentes A y B con un vecino común C, comprobar si $C \in \mathbf{S}_{AB}$. Si esto no es cierto, tomar la dirección de los arcos $A - C$ y $C - B$ como $A \rightarrow C$ y $C \leftarrow B$.
4. Establecer la dirección de los arcos que todavía no se encuentran dirigidos aplicando de forma recursiva las dos reglas siguientes:
   a) Si A es adyacente a B y hay un camino estrictamente dirigido desde A hacia B entonces, establecer la dirección de $A - B$ como $A \rightarrow B$.
   b) Si A y B no son adyacentes pero $A \rightarrow C$ y $C - B$, entonces $C \rightarrow B$.
5. Devolver el GAD o GAPD resultante.

Las pruebas de independencia condicional utilizadas para estimar redes Bayesianas discretas son funciones de las frecuencias observadas $\{n_{ijk}, i = 1, \cdots, R, j = 1, \cdots, C, k = 1, \cdots .L\}$ para las variables aleatorias X y W y todas las configuraciones de las variables condicionantes $\mathbf{Z}$:

- La prueba de la información mutua definida como:

$$MI(X, W|\mathbf{Z}) = \sum_{i}^{R}\sum_{1}^{C}\sum_{j\ 1}^{L}\sum_{k=1}\frac{n_{ijk}}{n}\log\frac{n_{ijk}n_{++k}}{n_{i+k}n_{+jk}} \tag{2.5.8}$$

y es equivalente a la prueba de razón de log verosimilitud ya que se diferencian por el factor $2n$, donde n es el tamaño de la muestra.

- La clásica prueba X^2 de Pearson para tablas de contingencia:

$$\chi^2(X, W|\mathbf{Z}) = \sum_{i=1}^{R}\sum_{j=1}^{C}\sum_{k=1}^{L}\frac{(n_{ijk} - m_{ijk})^2}{m_{ijk}} \tag{2.5.9}$$

donde $m_{ijk} = \frac{n_{i+k}n_{+jk}}{n_{++k}}$.

Otra posibilidad es el estimador de la contracción de la información mutua estudiada en el contexto de las redes Bayesianas [Hausser and Strimmer, 2009], [Scutari and Brogini, 2012]. Para todas las pruebas anteriores, la hipótesis nula de independencia puede ser probada usando:

- La distribución asintótica $\chi^2_{(R-1)(C-1)L}$, que es muy rápida de calcular pero que requiere un tamaño de muestra adecuado.
- El enfoque de permutación de Monte Carlo o el enfoque secuencial de permutación de Monte Carlo [Edwards, 2000] que es más rápido. Ambos enfoques aseguran que las pruebas sean insesgadas, y que sean válidas incluso para tamaños de muestra pequeños. Sin embargo, son costosos computacionalmente.
- La distribución semiparamétrica χ^2 [Tsamardinos and Borboudakis, 2010], lo que representa un compromiso entre los dos enfoques anteriores.

Ejemplo: Algoritmo Grow-Shrink

Se presenta una aplicación del Algoritmo 2.1 usando el conjunto artificial de datos `learning.test`, disponible en la librería `bnlearn`, que contiene 5000 filas y 6 variables: A, B, C, D, E y F. En el primer paso se determina el manto de Markov de cada nodo usando las pruebas de independencia condicional basados en la información mutua.

Para el nodo A los resultados fueron:

- A es dependiente de B ($p-$valor = 0).
- A es dependiente de D dado B ($p-$valor = 0).
- A es dependiente de C dados B y D ($p-$valor = 0).

Tabla 2.1: Algoritmo Grow-Shrink (paso 1)

Nodo	Manto de Markov ($\mathcal{MB}$)
A	$\mathcal{MB}(A) = \{B, D, C\}$
B	$\mathcal{MB}(B) = \{A, E, F\}$
C	$\mathcal{MB}(C) = \{D, A\}$
D	$\mathcal{MB}(D) = \{A, C\}$
E	$\mathcal{MB}(E) = \{B, F\}$
F	$\mathcal{MB}(F) = \{E, B\}$

En el segundo paso se identifican los vecinos de cada nodo. Los vecinos del nodo A se obtienen con las siguientes pruebas de independencia condicional:

- A es dependiente de B ($p-$valor = 0).
- A es dependiente de B dado E ($p-$valor = 0).
- A es dependiente de B dado F ($p-$valor = 0).
- A es dependiente de D ($p-$valor = 0).
- A es dependiente de D dado C ($p-$valor = 0).
- A es independiente de C ($p-$valor = 0.8598).

Los vecinos de los nodos A, B, C, D, E y F se muestran en la Tabla 2.2.

Tabla 2.2: Algoritmo Grow-Shrink (paso2)

Nodo	Vecinos ($\mathcal{N}$)
A	$\mathcal{N}(A) = \{B, D\}$
B	$\mathcal{N}(B) = \{A, E\}$
C	$\mathcal{N}(C) = \{D\}$
D	$\mathcal{N}(D) = \{A, C\}$
E	$\mathcal{N}(E) = \{B, F\}$
F	$\mathcal{N}(F) = \{E\}$

En el paso 3 se agregan arcos no dirigidos entre cada nodo y sus vecinos. Se observa la presencia de dos $v-$estructuras:

$$A \rightarrow D \leftarrow C \quad \text{y} \quad B \rightarrow E \leftarrow F$$

La primera $v-$estructura, centrada en D, se determina ya que la prueba de independencia condicional correspondiente concluye que A es dependiente de C dado D. La segunda $v-$estructura, centrada en E, se determina con la prueba de independencia condicional que establece que B es dependiente de F dado E. El GAPD obtenido con el algoritmo Grow-Shrink se muestra en la Figura 2.5.

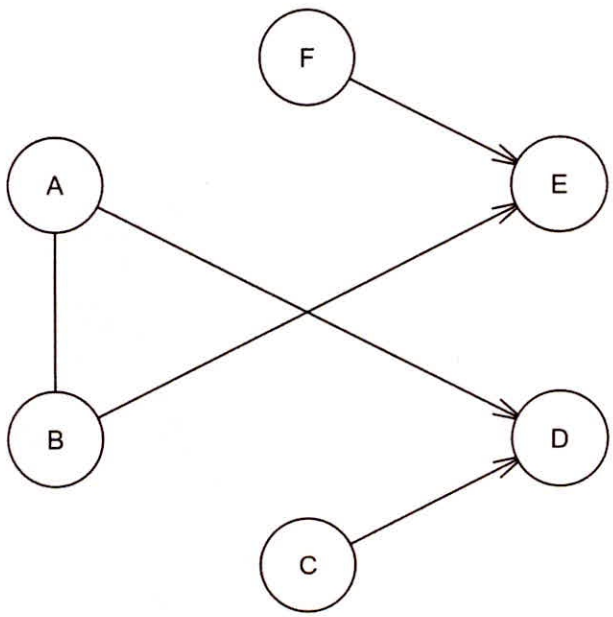


Figura 2.5: Ejemplo algoritmo Grow-Shrink

2.5.1.2. Algoritmos basados en scores

Los algoritmos basados en scores representan la aplicación de técnicas de optimización heurística al problema de estimación de la estructura de la red Bayesiana. A cada red candidata se asigna un puntaje de acuerdo a su estructura y que refleja su grado de bondad de ajuste. Algunos ejemplos de esta clase de algoritmos son:

- Hill-Climbing: explora el espacio de búsqueda a partir de la estructura de la red y añade, borra o invierte un arco a la vez hasta que el score ya no se pueda mejorar tal como se muestra en el algoritmo 2.2.
- Genetic algorithms: que imitan la evolución natural a través de la selección iterativa de los modelos más aptos [Larranaga et al., 1997].
- Simulated annealing: este algoritmo realiza una búsqueda local aceptando cambios que incrementan o disminuyen el score de la red con una probabilidad inversamente proporcional a la disminución del score [Bouckaert, 1995].

Algoritmo 2.2 Algoritmo Hill-Climbing

1. Elija una estructura de red G sobre $\mathbf{V}$, no necesariamente vacía.
2. Calcular el score de G denotado por $\mathrm{Score}_G = \mathrm{Score}(G)$.
3. Sea $\mathrm{maxScore} = \mathrm{Score}_G$.
4. Repita los siguientes pasos conforme $\mathrm{maxScore}$ aumenta:
   a) Para cada posible arco agregado, eliminado o invertido que no resulte en una red cíclica:
      1) Calcular el score de la red modificada $G*$, $\mathrm{Score}_{G*} = \mathrm{Score}(G*)$.
      2) Si $\mathrm{Score}_{G*} > \mathrm{Score}_G$, tomar $G = G*$ y $\mathrm{Score}_G = \mathrm{Score}_{G*}$.
   b) Actualizar $\mathrm{maxScore}$ con el nuevo valor de Score_G.
5. Devolver el GAD G.

Ejemplo: Algoritmo Hill-Climbing

Se considera inicialmente una red vacía con un valor $BIC = -28277.59$. En el primer paso se agrega a la red el arco $A \rightarrow B$ que es el que permite obtener el mayor incremento del score (1153.88). En el segundo paso se agrega el arco $A \rightarrow D$ a la red anterior usando el mismo criterio. Sin embargo, en este paso hay dos acciones

más a tomar en cuenta: eliminar el arco $A \rightarrow B$ o invertir la dirección como $B \rightarrow A$. El cambio en el score obtenido por estas acciones se muestran a continuación:

Tabla 2.3: Algoritmo Hill-Climbing (paso 2)

Acción	Incremento en el score
Agregar $A \rightarrow D$	1128.08
Eliminar $A \rightarrow B$	−1153.88
Invertir $A \rightarrow B$	0

Se decide agregar el arco $A \rightarrow D$. En el tercer paso se tiene que el mayor incremento en el score de la red anterior se obtiene cuando se agrega el arco $C \rightarrow D$. Las posibles acciones son:

Tabla 2.4: Algoritmo Hill-Climbing (paso 3)

Acción	Incremento en el score
Agregar $C \rightarrow D$	823.76
Eliminar $A \rightarrow B$	−1153.88
Eliminar $A \rightarrow D$	−1128.08
Invertir $A \rightarrow B$	0
Invertir $A \rightarrow D$	0

Se decide agregar el arco $C \rightarrow D$. El proceso anterior se repite hasta que ninguna de las posibles acciones incremente el valor del score obtenido con la estructura anterior. La red final se muestra en la Figura 2.6.

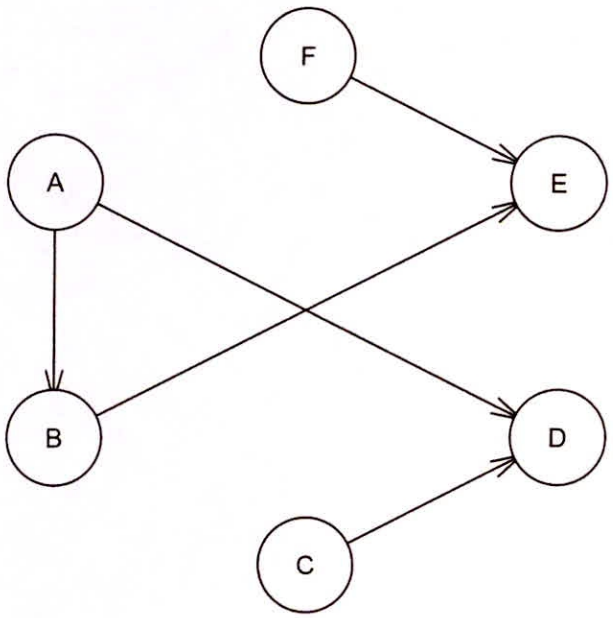


Figura 2.6: Ejemplo algoritmo Hill-Climbing

2.5.1.3. Algoritmos híbridos

Los algoritmos híbridos combinan algoritmos basados en restricciones y scores para compensar las respectivas debilidades de cada método y así producir estructuras más confiables en una amplia variedad de situaciones. Los dos miembros más conocidos de esta familia son el algoritmo Sparse Candidate, SC, [Friedman et al., 1999] y el algoritmo Max-Min Hill-Climbing, MMHC, [Tsamardinos et al., 2006]. El algoritmo MMHC se ilustra a continuación.

Algoritmo 2.3 Algoritmo Max-Min Hill-Climbing

1. Elija una estructura de red G sobre $\mathbf{V}$, no necesariamente vacía.
2. Realice los siguientes pasos:
   a) **Fase de restricción**: seleccionar un conjunto $\mathbf{C}_i \subset \mathbf{V}$ de los padres e hijos candidatos para cada nodo $X_i \in \mathbf{V}$.
   b) **Fase de maximización**: usar el algoritmo Hill-Climbing para encontrar la estructura de la red $G*$ que maximiza $\mathrm{Score}(G*)$ entre las redes en las que los padre o hijos de cada nodo X_i se encuentren incluidos en el correspondiente conjunto $\mathbf{C}_i$.
3. Devolver el GAD G.

Ambos algoritmos, SC y MMHC, se realizan en dos fases llamadas restricción y maximización. En el primer paso, el conjunto de candidatos para los padres e hijos de cada nodo X_i se reduce de todo el conjunto de nodos $\mathbf{V}$ a un conjunto pequeño de nodos $\mathbf{C}_i \subset \mathbf{V}$, cuyo comportamiento ha demostrado estar relacionado de alguna manera al nodo X_i. Esto a su vez da lugar a un espacio de búsqueda más pequeño y más regular. En el segundo paso se busca la red que maximiza una función de score dada, sujeto a las restricciones impuestas por los conjuntos de nodos en $\mathbf{C}_i$.

En el algoritmo Sparse Candidate estas dos fases se aplican iterativamente hasta que no haya cambio o no exista red que mejora el score. Por otro lado, en el algoritmo de Max-Min Hill-Climbing las fases de restricción y maximización se realizan una sola vez ya que los padres e hijos candidatos se utilizan con el algoritmo Hill-Climbing para hallar la red óptima.

Ejemplo: Algoritmo Max-Min Padres e Hijos

La fase de restricción de este algoritmo consiste en aplicar las pruebas de independencia condicional para hallar el conjunto candidato $\mathbf{C}_i$ de padres e hijos en cada nodo. Los resultados correspondientes al nodo A se muestran a continuación:

- B es dependiente de A ($p-$valor = 0).
- C es independiente de A ($p-$valor = 0.8598).
- D es dependiente de A ($p-$valor = 0).

- E es dependiente de A ($p-$valor = 0).
- F es independiente de A ($p-$valor = 0.5218).

El nodo B es aceptado como un candidato inicial para el nodo A. Luego se realizan las siguientes pruebas de independencia condicional que toman en cuenta el candidato encontrado:

- D es dependiente de A dado B ($p-$valor = 0).
- E es independiente de A dado B ($p-$valor = 0.8168).

Finalmente, los nodos B y D son aceptados como candidatos a ser padre o hijo del nodo A.

Tabla 2.5: Algoritmo Max-Min Padres e Hijos (fase de restricción)

Nodo	Padres e Hijos
A	$\mathcal{PH}(A) = \{B, D\}$
B	$\mathcal{PH}(B) = \{A, E\}$
C	$\mathcal{PH}(C) = \{D\}$
D	$\mathcal{PH}(D) = \{A, C\}$
E	$\mathcal{PH}(E) = \{A, B, F\}$
F	$\mathcal{PH}(F) = \{E\}$

En la fase de maximización se utiliza el algoritmo Hill-Climbing considerando los candidatos padres e hijos obtenidos en la fase anterior. Se considera inicialmente una red vacía usando los nodos A, B y D con un valor BIC = -15838.77. En el primer paso se agrega a la red el arco $A \rightarrow B$ que es el que permite obtener el mayor incremento del score (1153.88). En el segundo paso se agrega el arco $A \rightarrow D$ a la red anterior usando el mismo criterio. Sin embargo en este paso hay dos acciones mas a tomar en cuenta: eliminar el arco $A \rightarrow B$ o invertir la direccion como $B \rightarrow A$. El cambio en el score obtenido por estas acciones se muestran a continuación:

Tabla 2.6: Algoritmo Hill-Climbing (fase de maximización)

Acción	Incremento en el score
Agregar $A \rightarrow D$	1128.08
Eliminar $A \rightarrow B$	−1153.88
Invertir $A \rightarrow B$	0

Se decide agregar el arco $A \rightarrow D$. El gráfico obtenido al final del proceso se muestra en la Figura 2.7.

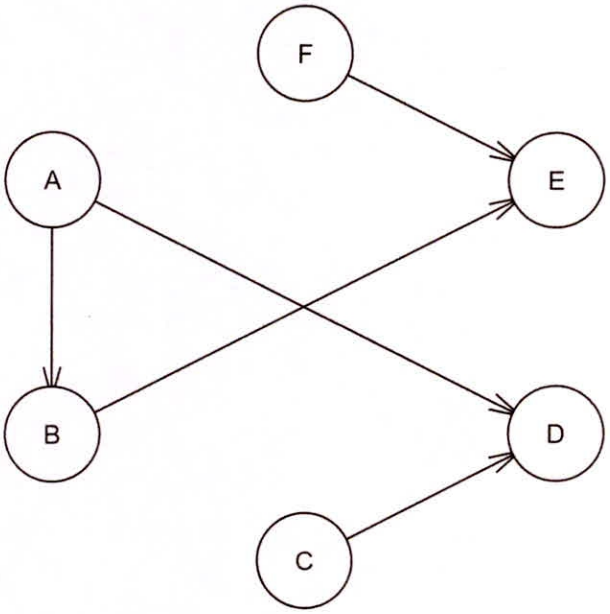


Figura 2.7: Ejemplo algoritmo Hill-Climbing y Max-Min Padres e Hijos

2.5.2. Estimación de parámetros

Una vez que la estructura de la red Bayesiana se ha obtenido a partir de la data, la tarea de estimación de los parámetros de la distribución global se simplifica en gran medida por la descomposición en las distribuciones locales. Dos enfoques son comunes en la literatura: la estimación de máxima verosimilitud y la estimación Bayesiana. Otras opciones, Existen otras opciones, como los estimadores de contracción [Hausser and Strimmer, 2009], [Schäfer and Strimmer, 2005].

Para las redes Bayesianas discretas los parámetros se estiman usando las

probabilidades condicionales en las distribuciones locales. Estos parámetros pueden ser estimados con las correspondientes frecuencias empíricas en el conjunto de datos. Lo anterior produce las clásicas estimaciones de probabilidad frecuentista y de máxima verosimilitud. A pesar de que este enfoque parece ser una buena aproximación hay que tomar en cuenta que muchas veces el conjunto de datos no abarca todas las posibles combinaciones de los valores de las variables predictoras, por lo que este tipo de estimación puede producir probabilidades iguales a cero.

Como alternativa, se pueden estimar las probabilidades condicionales en un escenario Bayesiano. En el caso de las redes Bayesianas discretas se asume una muestra multinomial, $X_i|\Pi_{X_i} \sim \mathcal{M}(\Theta_{X_i}|\Pi_{X_i})$, donde $\Theta_i|\Pi_{X_i}$ representa las probabilidades condicionales $\pi_{ijk} = \Pr(X_i = k|\Pi_{X_i} = j)$. Si se considera una distribución a priori conjugada Dirichlet, $\Theta_{X_i}|\Pi_{X_i} \sim \mathcal{D}(\alpha_{ijk})$, se obtiene una distribución posterior $\mathcal{D}(\alpha_{ijk} + n_{ijk})$, donde $\sum_{jk} \alpha_{ijk} = \alpha_i$ es conocido como el tamaño de muestra imaginario que determina cuánto peso se asigna a la distribución a priori en comparación con los datos cuando se obtiene la distribución posterior.

Es importante tener en cuenta que el enfoque utilizado para estimar la estructura de la red Bayesiana no determina necesariamente qué metodología puede ser utilizada en la estimación de parámetros. Por ejemplo, utilizando densidades posteriores tanto en la estimación de la estructura y los parámetros se logra que la interpretación de la red Bayesiana y el proceso de inferencia sea sencillo. Sin embargo, utilizando la prueba de permutación de Monte Carlo para la estimación de la estructura y la posterior estimación de los parámetros también es común.

A pesar de que las distribuciones locales en la práctica consideran sólo a un pequeño número de variables, y su dimensión general, no escala con el tamaño de la red Bayesiana, la estimación de parámetros sigue siendo problemática en algunas situaciones. Por ejemplo, es común tener tamaños de muestra mucho más pequeños en comparación al número de variables incluidas en el modelo. Esto es típico de los conjuntos de datos biológicos de alto rendimiento, como los microarrays, que tiene diez o cien observaciones y miles de genes. En este contexto, las estimaciones tienen una alta variabilidad a menos que se tomen precauciones tanto en la estimación de la estructura y de los parámetros.

2.6. Clasificadores por redes Bayesianas

Las redes Bayesianas son particularmente útiles en el proceso de clasificación supervisada [Schäfer and Strimmer, 2005], [Cheng and Greiner, 1999] ya que permiten simplificar la distribución conjunta de las variables predictoras usando distribuciones locales con una estructura más sencilla a partir de las relaciones de dependencia e independencia condicional existentes entre las variables.

Las redes Bayesianas permiten representar la estructura probabilística de las variables predictoras que se utiliza para predecir la clase de pertenencia de cada observación usando el teorema de Bayes:

$$\Pr(Y = y|X_1, X_2, \cdots, X_p) = \frac{\Pr(Y = y)\Pr(X_1, X_2, \cdots, X_p|Y = y)}{\Pr(X_1, X_2, \cdots, X_p)} \quad (2.6.1)$$

e asigna la observación en aquella clase o categoría que tenga la mayor probabilidad calculada a partir de la ecuación anterior.

2.6.1. Naive Bayes

Naive Bayes, y sus variantes, se encuentran entre los algoritmos más conocidos para construir clasificadores de documentos de texto [Kim et al., 2003], filtración de correo electrónico [Sahami, 1996], clasificación de galaxias [Bazell and Aha, 2001] y reconocimiento de emociones [Sebe et al., 2002]. A pesar de su simplicidad es comparable con clasificadores sofisticados como las redes neuronales y los árboles de decisión, ya que posee alta precisión y velocidad cuando es aplicada a conjuntos grandes de datos.

El clasificador Naive Bayes fue popularizado por [Duda et al., 1973] gracias a su simplicidad, eficiencia y bajo error de clasificación. Como ya se mencionó, este clasificador supone que todas las variables son condicionalmente independientes dado el valor de la variable de clase. Este supuesto simplifica la representación de $\Pr(X_1, X_2, \cdots, X_p|Y = y)$ así como su estimación a partir de la muestra de entrenamiento. La estructura de este clasificador puede representarse usando una red Bayesiana, tal como se muestra en la Figura 2.8.

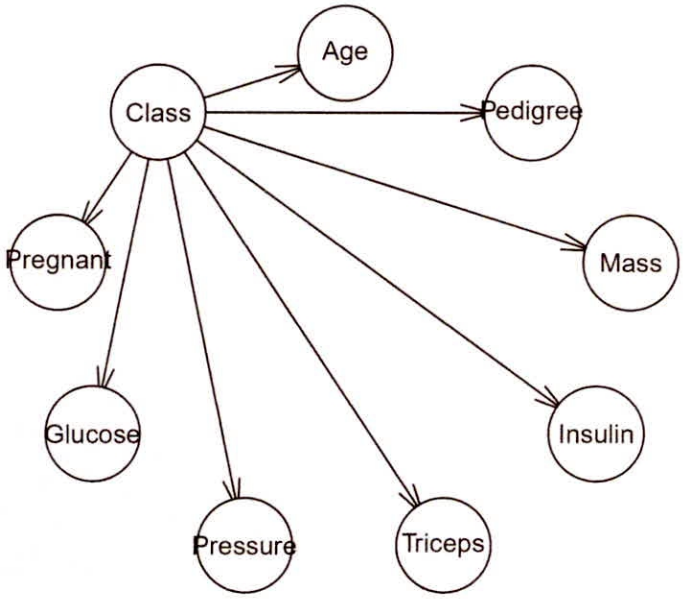


Figura 2.8: Clasificador Naive Bayes para la data `Diabetes`

El funcionamiento de Naive Bayes es sorprendente, ya que el supuesto de independencia no es realista. Considere un clasificador para la evaluación del riego en las solicitudes de crédito: es contra intuitivo ignorar las correlaciones entre edad, nivel de educación e ingreso. El ejemplo anterior lleva a la necesidad construir un clasificador que tome en consideración las relaciones de dependencia que existen en el conjunto de variables predictoras.

2.6.2. Tree Augmented Network

El clasificador Tree Augmented Network, TAN, fue propuesto por [Friedman et al., 1997] como una extencion de Naive Bayes que mantiene su estructura básica. Para mejorar el comportamiento del clasificador, propuso aumentar

la estructura de Naive Bayes con arcos entre las variables predictoras, cuando éstos sean necesarios, desechando así el supuesto de independencia. Estas estructuras son llamadas redes aumentadas Naive Bayes y estos arcos, arcos aumentados.

En una estructura aumentada, un arco desde X_i hacia X_j implica que la influencia de X_i en la asignación de la variable de clase Y también depende del valor de X_j. El propósito está en encontrar una red Naive Bayes de árbol aumentado, TAN, en la que la variable de clase no tenga padres y las variables predictoras tengan como padres a la variable de clase y a lo más alguna otra variable, tal como se muestra en la Figura 2.9.

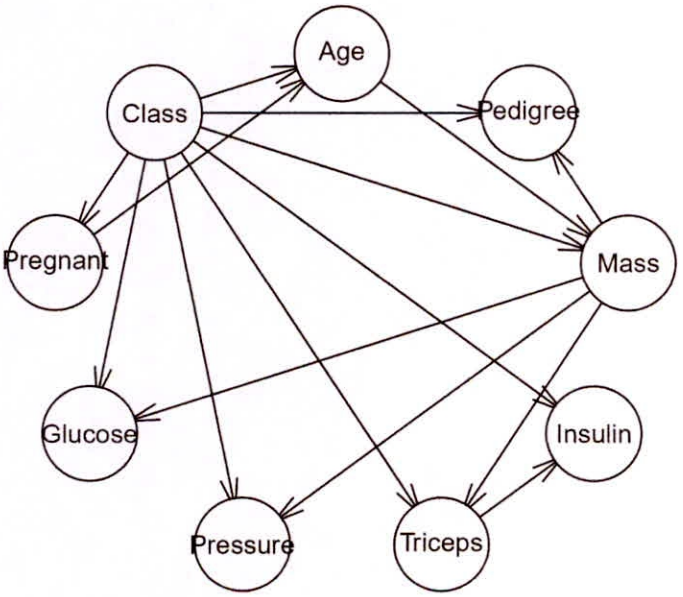


Figura 2.9: Clasificador TAN para la data `Diabetes`

El procedimiento para obtener estos arcos esta basado en el algoritmo de Chow y Liu [Chow and Liu, 1968] para estimar las relaciones de dependencia entre un conjunto de variables usando una estructura de árbol.

2.7. Proceso de discretización Chi-Merge

El algoritmo de discretización ChiMerge [Kerber, 1992] realiza un proceso de fusión de abajo hacia adelante, donde los intervalos adyacentes se juntan continuamente hasta que se cumple cierta condición de parada. Se ordenan los datos a discretizar y se considera que cada uno de ellos conforma un intervalo diferente. El proceso de fusión contiene dos pasos: (1) calcular el valor del estadistico de prueba χ^2 de independencia para cada par de intervalos adyacentes, y (2) fusionar el par de intervalos adyacentes con el valor χ^2 más bajo. El proceso continúa hasta que todos los pares de intervalos tienen valores χ^2 que exceden cierto umbral predefinido; es decir, hasta que todos los intervalos adyacentes se consideren significativamente diferentes por la prueba de independencia. El valor del umbral se determina seleccionando un nivel de significación y calculado luego el percentil correspondiente a la distribución chi-cuadrado con $k-1$ grados de libertad.

La fórmula para calcular el estadístico de prueba χ^2 es:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \tag{2.7.1}$$

donde:

- k, es el número de clases o categorías
- A_{ij}, es el número de datos en el $i-$ésimo intervalo y $j-$ésima clase
- R_i, es el número de datos en el $i-$ésimo intervalo
- C_j, es el número de datos en la $j-$ésima clase
- E_{ij}, es la frecuencia esperada de A_{ij}

.

2.8. Algoritmo SES para selección de variables predictoras

El proceso de selección de variables es una de las tareas fundamentales en el área del machine learning. En general se busca identificar un subconjunto de variables predictoras que son relevantes con respecto a una tarea específica; por ejemplo, en regresión y clasificación se busca seleccionar y retener el subconjunto de variables predictoras con el más alto poder predicitivo. Los principales objetivos de la selección de variables suelen ser [Guyon and Elisseeff, 2003]:

- Mejorar el rendimiento de un modelo predictivo.
- Evitar el costo asociado con el uso de todas las variables.
- Obtener una mejor comprensión del modelo predictivo eliminando las variables irrelevantes y redundantes.

Sin embargo, a menudo ocurre que varios subconjuntos de variables predictoras son aproximadamente iguales para una tarea dada. El saber que existen múltiples subconjuntos igualmente predictivos aumenta la comprensión del problema en cuestión. Por el contrario, la identificación de un solo subconjunto puede llevar a ignorar los factores que pueden jugar un papel importante para la comprensión del problema en estudio. En términos prácticos, los subconjuntos igualmente predictivos puede diferir en términos del costo y esfuerzo necesario para medir sus respectivos componentes. Por lo tanto, proporcionar subconjuntos alternativos puede tener un gran impacto en contextos donde algunos de los factores pueden ser técnicamente difíciles o excesivamente costosos de medir.

Se han desarrollado algoritmos que generan múltiples conjuntos de variables equivalentes [Statnikov and Aliferis, 2010], [Huang et al., 2014]. El algoritmo Statistically Equivalent Signature [Tsamardinos et al., 2012], SES, permite identificar múltiples subconjuntos de variables con rendimientos estadísticamente equivalentes. De esta forma, SES se presenta como una extensión de los métodos tradicionales de selección de variables predictoras.

Sea D un conjunto de datos definido sobre un conjunto de V variables predictoras y una variable respuesta Y. Los métodos de selección de variables basados en restricciones aplican de forma repetitiva pruebas de independencia condicional con

el objetivo de identificar el subconjunto de variables relacionadas con Y dado cualquier otro subconjunto de variables en V. Se denota por $\rho_{XY|W}$ al $p-$valor obtenido por una prueba estadística que permite evaluar la hipótesis nula donde las variables X y Y son condicionalmente independientes dado el conjunto de variables W. El algoritmo SES requiere dos parámetros: un umbral a, para considerar la existencia de independencia condicional y el número máximo de variables k que pueden ser incluidas en cualquier conjunto condicional. Este último parámetro limita la complejidad y los requerimientos computacionales del proceso. El resultado del algoritmo será un subconjunto E de listas de equivalencia Q_i, $i = 1, \cdots, n$ que contienen las variables que son equivalentes con las otras.

Al inicio del proceso se crea un conjunto vacío S de variables seleccionadas de tal forma que todas las variables predictoras en V podrían ser incluidas en S ($R \leftarrow V$, donde R es el conjunto de variables consideradas para inclusión) y se considera además que cada variable es equivalente solo consigo misma ($Q_i \leftarrow i$). Durante el proceso el algoritmo intenta:

1. Incluir en S la variable con mayor asociación con Y condicionada en todo subconjunto posible de variables previamente seleccionadas.

2. Excluir de S cualquier variable predictora que ya no se encuentre asociada con Y dado cualquier subconjunto Z de otras variables en S. Una vez que X es excluida de S no puede volver a incluirse en un paso posterior.

Sin embargo, antes de eliminar X de S, el algoritmo debe identificar cualquier otra variable W en Z que sea equivalente a X, al verificar si $\rho_{XY|Z*} > a$, donde $Z* \leftarrow (Z \cup \{X\})/W$. Si tal variable existe, la lista de variables equivalentes a X, Q_X, se agrega a Q_Y. Finalmente se presentan todas las listas de equivalencia Q_i, donde $i \in S$.

Se puede construir una firma predictiva (predictive signature) eligiendo una y solo una variable de cada lista de equivalencia Q_i. Suponga, por ejemplo, que E contiene tres listas de equivalencias: $Q_1 = \{X_1, X_4\}$, $Q_3 = \{X_3\}$ y $Q_5 = \{X_5, X_2\}$. Entonces hay un total de $2 \times 1 \times 2 = 4$ posibles firmas, es decir, $S_a = \{X_1, X_3, X_5\}$, $S_b = \{X_1, X_3, X_2\}$, $S_c = \{X_4, X_3, X_5\}$ y $S_d = \{X_4, X_3, X_2\}$. Por el contrario, los conjuntos $\{X_1, X_2\}$ y $\{X_1, X_4, X_3, X_5\}$ no son firmas equivalentes válidas, ya que la primera no selecciona ninguna variable de Q_3 y la última incluye dos variables dentro de Q_1.

Suponga que X es una variable predictora y W un subconjunto de variables predictoras. Se desea probar si Y es independiente de X dado W comparando el

modelo de regresión logística multinomial que incluye como predictores a X y W, con el modelo de regresión logística multinomial alternativo que solo incluye a W. Cuando el modelo completo muestra una mejora significativa en comparacion con el modelo alternativo, entonces se puede concluir que las variables X y Y se encuentran relacionadas dado W.

La prueba de independencia condicional basada en la regresión logística multinomial se plantea como una prueba de comparación entre el logaritmo de la función de verosimilitud del modelo completo (MC) y el logaritmo de la función de verosimilitud del modelo alternativo (MA). El estadístico de prueba:

$$D = 2(\log L_{MC} - \log L_{MA}) \tag{2.8.1}$$

sigue una distribución χ^2 con un grado de libertad. Conocido el valor de D y su distribución teórica se puede calcular $\rho_{XY|W}$ para evaluar la hipótesis de independencia condicional que permita determinar las listas de equivalencia Q_i.

Capítulo 3

Metodología

3.1. Métodos empleados en la investigación

En esta sección se describe el proceso de construcción de los clasificadores propuestos considerando las relaciones de dependencia existentes entre las variables predictoras. Los conjuntos de datos fueron tomados del repositorio UCI Machine Learning, disponibles en la siguiente dirección web: http://archive.ics.uci.edu/ml/. Los conjuntos de datos elegidos son los más utilizados en el área de las redes Bayesianas y han sido utilizados para comparar los clasificadores tradicionales como Naive Bayes y TAN. El nombre de cada conjunto de datos, el número de variables predictoras, el número de categorías que tiene la variable respuesta y el tipo de variables predictoras se muestran en la Tabla 3.1.

Tabla 3.1: Conjuntos de datos

N°	Nombre	Variables	Categorías	Tipo
1	Australian	14	2	Cuantitativas y cualitativas
2	Breast	9	2	Cualitativas
3	Corral	6	2	Cualitativas
4	Diabetes	8	2	Cuantitativas
5	German	20	2	Cuantitativas y cualitativas
6	Glass	9	6	Cuantitativas
7	Heart	13	2	Cuantitativas y cualitativas
8	Hepatitis	19	2	Cuantitativas y cualitativas
9	Iris	4	3	Cuantitativas
10	Vehicle	18	4	Cuantitativas
11	Vote	16	2	Cualitativas

Se presenta a continuación una breve descripción de los conjuntos de datos usados:

- `Australian`: Presenta un total de 690 observaciones y 14 variables predictoras para una compañía de crédito australiana. El objetivo es determinar si se debe conceder o no una tarjeta de crédito a los solicitantes.
- `Breast`: Son 683 observaciones y 9 variables predictoras de casos clínicos recopilados por el Dr. Wolberg en la Universidad de Winconsin por un periodo aproximado de dos años. El problema es averiguar si los tumores son benignos o malignos, basándose en los datos del cáncer de cada paciente.
- `Corral`: Es un conjunto de datos artificial con un total de 128 observaciones. Contiene 6 variables predictoras: $X_1, X_2, \cdots, X_6$ cuya variable respuesta es $(X_1 \wedge X_2) \vee (X_3 \wedge X_4)$.
- `Diabetes`: Contiene información proveniente de un estudio con 8 variables predictores y una muestra de 768 pacientes mujeres del Instituto Nacional de enfermedades Digestivas, Diabetes y de Riñón. La variable respuesta tiene dos categorías: prueba de diabetes positiva y prueba de diabetes negativa.
- `German`: Conjunto de datos con un total de 1000 observaciones y 20 variables predictoras. Se debe decidir si es posible la concesión de préstamos a clientes.
- `Glass`: El estudio de la clasificación de los cristales ha sido motivada por la investigación criminológica. En la escena del crimen, los cristales abandonados pueden servir de prueba siempre y cuando sea posible

identificarlos correctamente. Este conjunto de datos tiene un total de 214 observaciones y 9 variables predictoras. Se desea determinar la categoría a la que pertenece el cristal abandonado.

- `Heart`: Este conjunto de datos tiene 13 variables predictoras y 270 observaciones. El objetivo es predecir la presencia o ausencia de enfermedad coronaria.
- `Hepatitis`: Son 155 observaciones sobre una base de datos de hepatitis. Tiene un total de 19 variables predictoras y dos posibles categorías para la variable respuesta.
- `Iris`: Tiene un total de 150 elementos. El objetivo es realizar una clasificación de lirios a través de 4 variables predictoras de tipo continuo: el ancho y el largo tanto del pétalo como del sépalo. Hay un total de tres tipos diferentes de lirios.
- `Vehicle`: Este conjunto de datos tiene un total de 846 observaciones. El propósito es averiguar, a través de sus 18 variables predictoras continuas, si un determinado vehículo es un Opel, un Saab, un autobús o una furgoneta.
- `Vote`: Con un total de 435 observaciones este conjunto de datos incluye votos para los representantes al congreso de los Estados Unidos. A través de 16 variables predictoras nominales se debe decidir si los electores votaran por los demócratas o por los republicanos.

3.2. Técnicas e instrumentos empleados

Los algoritmos usados para obtener la estructura entre las variables predictoras de cada conjunto de datos, discutidos en el Capítulo 2, son:

- Basados en restricciones: Algoritmo Grow-Shrink.
- Basados en scores: Algoritmo Hill-Climbing.
- Híbridos: Algoritmo Max-Min Padres e Hijos.

3.2.1. Etapa de preprocesamiento

A cada conjunto de datos se le aplicó una etapa inicial de preprocesamiento que consta de dos procesos. El primer proceso elimina los valores perdidos antes de aplicar los algoritmos propuestos. Los conjuntos de datos que tienen valores perdidos son `Breast`, `Heart`, `Hepatitis` y `Vote`. El segundo proceso consiste en discretizar los valores correspondientes a las variables predictoras cuantitativas usando el algoritmo Chi-Merge, disponible en la librería `dprep` dentro del software estadístico R. El proceso de discretización se aplicó a todas las variables predictoras en los conjuntos de datos mencionados en la Tabla 3.1 a excepción de `Breast`, `Corral` y `Vote`.

3.2.2. Etapa de estimación de la estructura de red

En la segunda etapa se realizó el proceso de estimación de la estructura presente en las variables predictoras usando los algoritmos propuestos. El algoritmo basado en restricciones utiliza la prueba de independencia condicional de información mutua, mientras que el algoritmo basado en scores e híbrido usan el criterio de información Bayesiano BIC.

En esta etapa se aplicó también el proceso de selección de variables SES para construir la estructura usando una menor cantidad de variables predictoras. De esta forma es posible comparar el rendimiento predictivo de los clasificadores construidos con todas las variables predictoras y los clasificadores construidos con las variables predictoras seleccionadas por el algoritmo SES. El resultado final en esta etapa es un gráfico acíclico dirigido tal como el que se muestra en la Figura 3.1 obtenido para la data `Iris`.

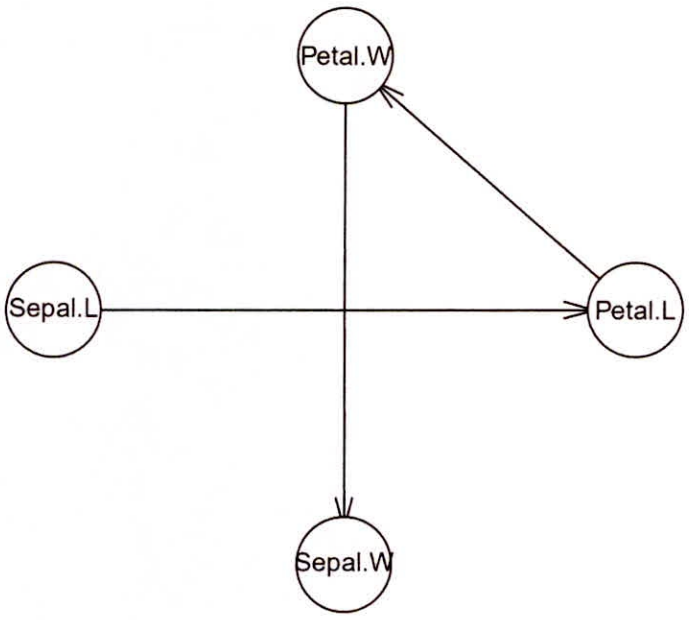


Figura 3.1: Red Bayesiana obtenida con la data `Iris`

3.2.3. Etapa de construcción del clasificador

Sobre la estructura obtenida en la etapa anterior se construyó el clasificador agregando un arco dirigido desde la variable respuesta hacia cada una de las variables predictoras. En esta etapa fue necesario escribir una función en R para incluir la variable respuesta en el GAD y los arcos dirigidos que la conectan con el resto de variables predictoras en la red. El clasificador obtenido para la data `Iris` se muestra en la Figura 3.2.

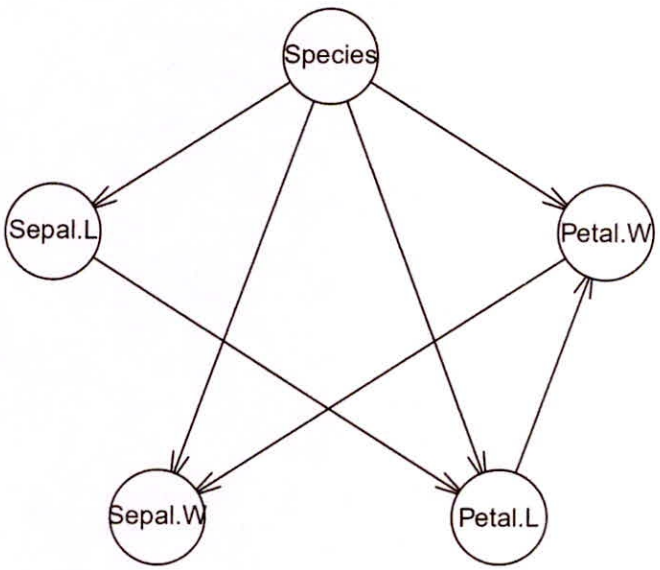


Figura 3.2: Clasificador obtenido con la data `Iris`

3.2.4. Etapa de estimación del error de clasificación

En la tercera etapa se realizó el proceso de validación cruzada 10 para calcular la precisión del clasificador usando la tasa de elementos correctamente clasificados. Los pasos a seguir son:

1. Se divide aleatoriamente el conjunto de datos en 10 partes.
2. Se realiza el proceso de estimación de parámetros usando 9 de estas partes como muestra de entrenamiento y evaluando el clasificador con la parte restante, llamada muestra de prueba, calculando la proporción de elementos correctamente clasificados.
3. Se repite el procedimiento anterior hasta que cada una de las 10 partes haya sido

utilizada como muestra de prueba.

Para efectos de evaluar la precisión del estimador se repite el proceso de validación cruzada 20 veces. Luego se calcula un promedio de las proporciones de observaciones correctamente clasificadas obtenidas en cada repetición. Se uso la librería `bnlearn` dentro del software estadístico R que ofrece una amplia variedad de algoritmos para la estimación de la estructura, estimación de parámetros y técnicas de inferencia en redes Bayesianas.

Capítulo 4

Análisis e interpretación

En este capítulo se presentan los resultados obtenidos al evaluar los clasificadores usando la tasa de elementos correctamente clasificados. Los clasificadores propuestos se construyen en dos escenarios:

- con todas las variables predictoras de cada conjunto de datos
- usando solamente las variables predictoras seleccionadas con el algoritmo SES.

Los resultados obtenidos en ambos escenarios serán comparados con los clasificadores Naive Bayes y TAN.

El GAD obtenido con el algoritmo basado en restricciones presentan, en algunos casos, arcos no dirigidos. La elección de la dirección para estos arcos es, de alguna manera, arbitraria ya que no siempre se conocen las relaciones existentes entre las variables predictoras con las que se trabaja. Sin embargo, las estructuras obtenidas de acuerdo a la elección realizada son equivalentes en términos de la cantidad de parámetros y de la medida de score utilizada, es decir el BIC. En el caso del algoritmo basado en scores y el algoritmo híbrido, las estructuras obtenidas tienen todos sus arcos dirigidos.

4.1. Clasificadores con todas las variables predictoras

La Tabla 4.1 muestra la tasa de elementos correctamente clasificados usando los clasificadores Bayesianos Naive Bayes, TAN y los obtenidos con los algoritmos

propuestos usando todas las variables predictoras de cada conjunto de datos.

Tabla 4.1: Tasa de elementos correctamente clasificados

N°	Nombre	Naive Bayes	TAN	Restricciones	Scores	Híbridos
1	Australian	0.894	0.816	0.895	0.896	0.897
2	Breast	0.976	0.962	0.973	0.966	0.973
3	Corral	0.865	0.995	0.829	0.865	0.865
4	Diabetes	0.843	0.749	0.836	0.833	0.841
5	German	0.924	0.674	0.919	0.908	0.908
6	Glass	0.783	0.730	0.772	0.760	0.772
7	Heart	0.869	0.674	0.860	0.838	0.863
8	Hepatitis	0.943	0.957	0.947	0.947	0.948
9	Iris	0.960	0.957	0.939	0.952	0.961
10	Vehicle	0.694	0.760	0.723	0.785	0.723
11	Vote	0.915	0.957	0.941	0.950	0.922

Al comparar el comportamiento predictivo de los clasificadores obtenidos se observa en la Tabla 4.2 que, en la mayoría de conjuntos de datos, la tasa de elementos correctamente clasificados es mayor con el algoritmo híbrido. Los conjuntos de datos en los que la tasa fue mayor en dos de los algoritmos considerados fueron `Breast`, `Corral` y `Glass`. En estos casos la estructura obtenida con estos algoritmos no es necesariamente la misma y la tasa de elementos correctamente clasificados llega a coincidir al utilizar tres decimales.

Tabla 4.2: Comparación entre los clasificadores propuestos

N°	Nombre	Algoritmo
1	Australian	Híbridos
2	Breast	Restricciones e Híbridos
3	Corral	Scores e Híbridos
4	Diabetes	Híbridos
5	German	Restricciones
6	Glass	Restricciones e Híbridos
7	Heart	Híbridos
8	Hepatitis	Híbridos
9	Iris	Híbridos
10	Vehicle	Scores
11	Vote	Scores

Para realizar el proceso de comparación de los clasificadores propuestos con Naive Bayes y TAN se construyen diagramas de dispersión para la tasa de elementos correctamente clasificados en cada conjunto de datos. Los puntos sobre la diagonal corresponden a los conjuntos de datos en los que el clasificador que se encuentra en el eje vertical tuve una mayor tasa de elementos correctamente clasificados en comparación al clasificador que se encuentra en el eje horizontal.

4.1.1. Naive Bayes y TAN versus Grow-Shrink

Los conjuntos de datos en los que el clasificador obtenido con el algoritmo Grow-Shrink presentó una pequeña mejora en el comportamiento predictivo en comparación con Naive Bayes fueron `Hepatitis`, `Vehicle` y `Vote`, tal como puede observarse en la Figura 4.1. Al realizar la comparación con el clasificador TAN, el clasificador propuesto obtuvo una mayor tasa de elementos correctamente clasificados en `Australian`, `Diabetes`, `German`, `Glass` y `Heart` según la Figura 4.2.

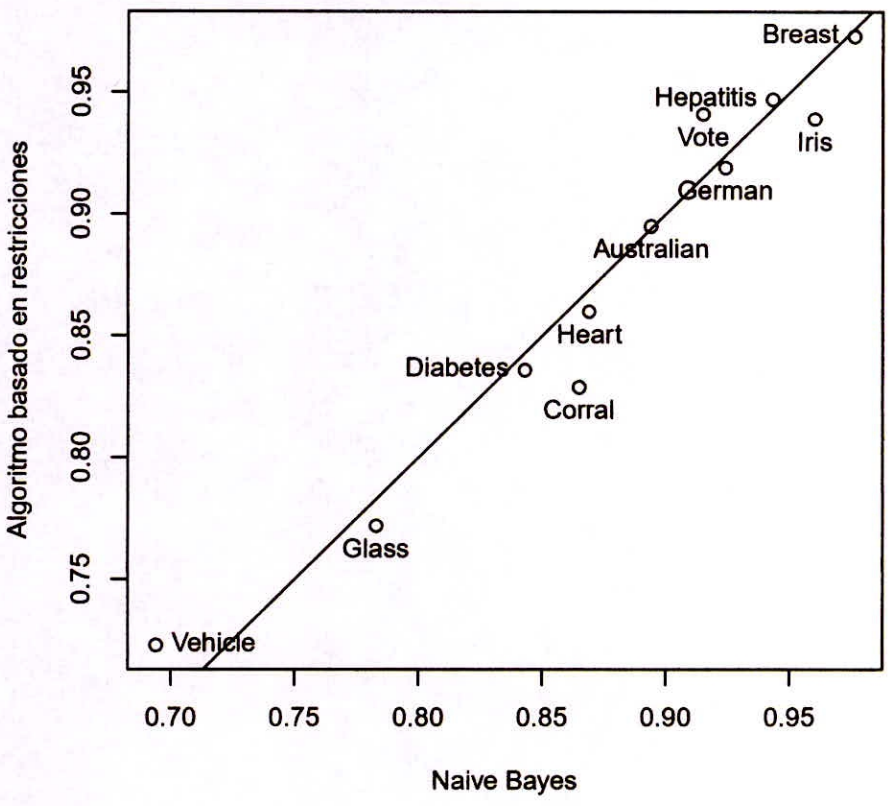


Figura 4.1: Comparación entre Naive Bayes y algoritmo basado en restricciones

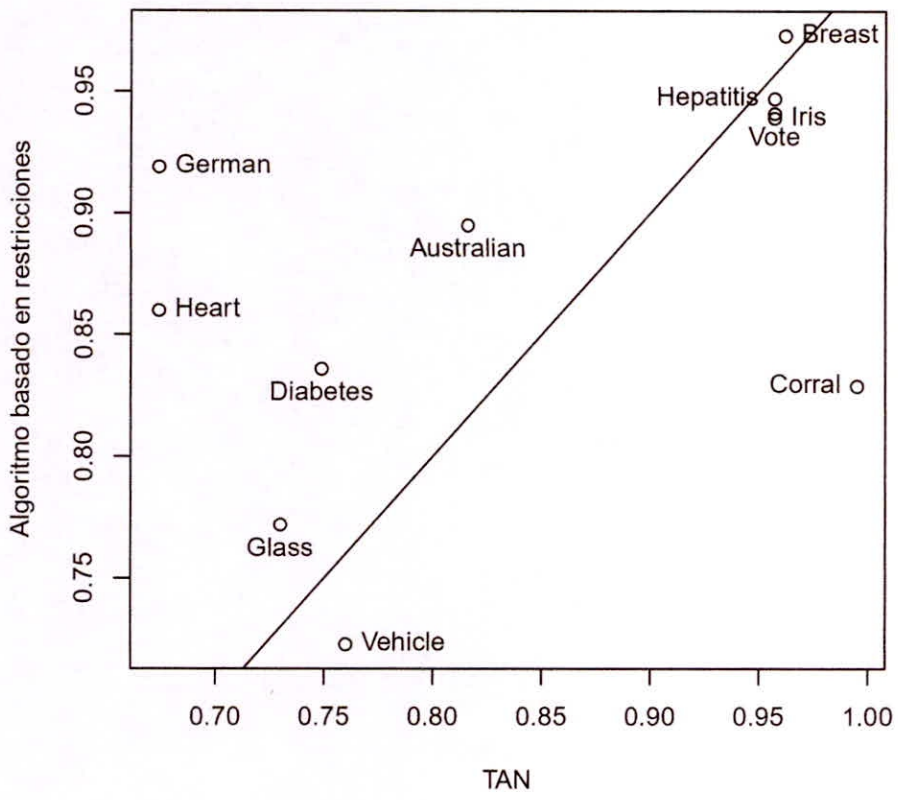


Figura 4.2: Comparación entre TAN y algoritmo basado en restricciones

En resumen, los clasificadores obtenidos con el algoritmo Grow-Shrink presentan un rendimiento predictivo similar con Naive Bayes. Al realizar la comparación con el clasificador TAN se obtienen mejores resultados en la tasa de elementos correctamente clasificados, sobre todo en los conjuntos de datos `Australian`, `Diabetes`, `German` y `Heart`.

4.1.2. Naive Bayes y TAN versus Hill-Climbing

En la Figura 4.3 se puede observar que en los conjuntos de datos `Hepatitis` y `Vote` el clasificador obtenido con el algoritmo Hill-Climbing tuvo una tasa de elementos correctamente clasificados ligeramente mayor en comparación con Naive Bayes, obteniéndose una mejora importante en `Vehicle`. En los conjuntos de datos

Australian, Breast, Diabetes, German, Glass, Heart y Vehicle se observa que el clasificador propuesto tuvo un rendimiento predictivo mayor en comparación con TAN, según la Figura 4.4.

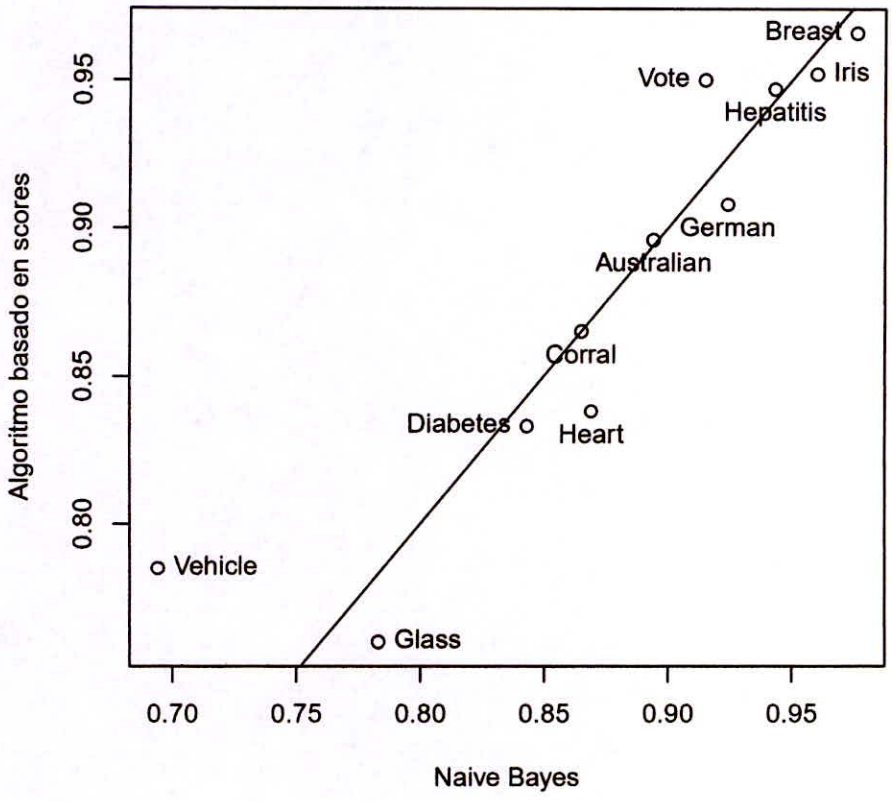


Figura 4.3: Comparación entre Naive Bayes y algoritmo basado en scores

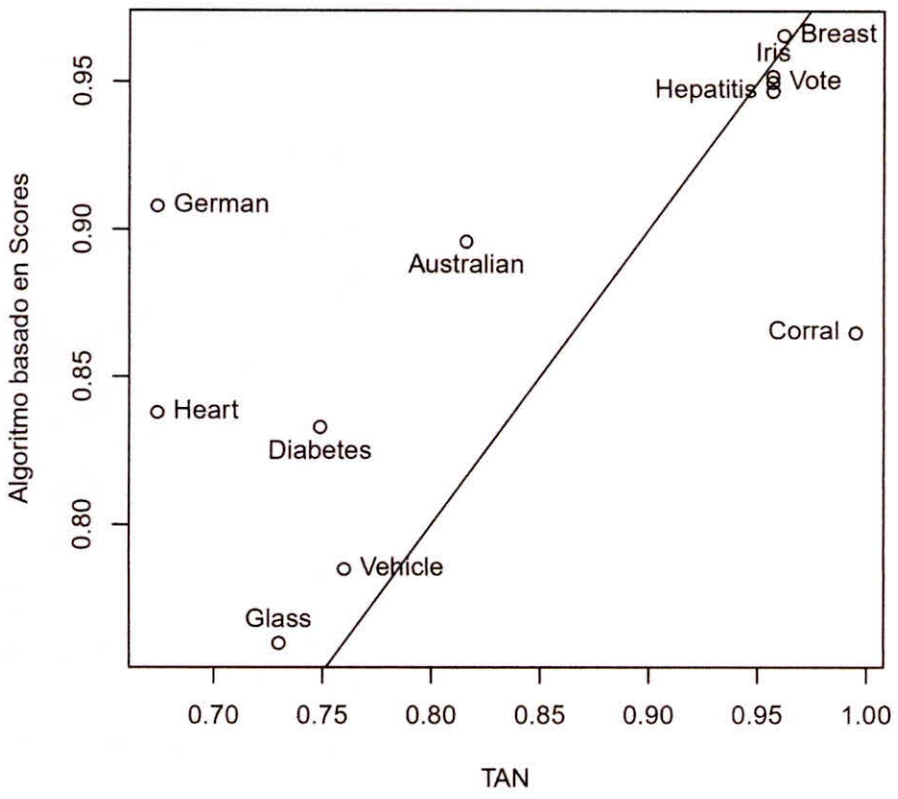


Figura 4.4: Comparación entre TAN y algoritmo basado en scores

Los clasificadores obtenidos con el algoritmo Hill-Climbing también presentan un rendimiento predictivo similar con Naive Bayes a excepción de `Vehicle` donde el algoritmo propuesto resulta ser superior. Al realizar la comparación con el clasificador TAN se obtienen nuevamente mejores resultados en términos predictivos en los conjuntos de datos `Australian`, `Diabetes`, `German` y `Heart`.

4.1.3. Naive Bayes y TAN versus Max-Min Padres e Hijos

El clasificador obtenido con el algoritmo Max-Min Padres e Hijos también tuvo un comportamiento predictivo ligeramente mayor en comparación a Naive Bayes en los conjuntos de datos `Australian`, `Hepatitis`, `Iris`, `Vehicle` y `Vote` tal como se observa en la Figura 4.5. Los conjuntos de datos en los que el clasificador propuesto

presenta una mayor tasa de elementos correctamente clasificados en comparación al clasificador TAN fueron `Australian`, `Breast`, `Diabetes`, `German`, `Glass`, `Heart` e `Iris` según se observa en la Figura 4.6.

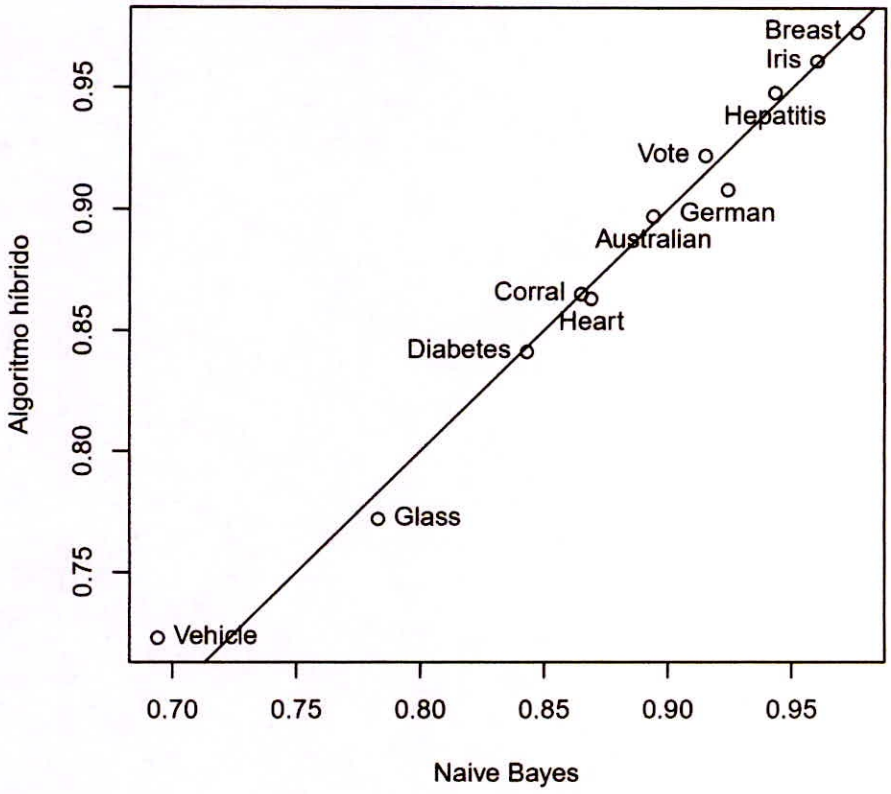


Figura 4.5: Comparación entre Naive Bayes y algoritmo híbrido

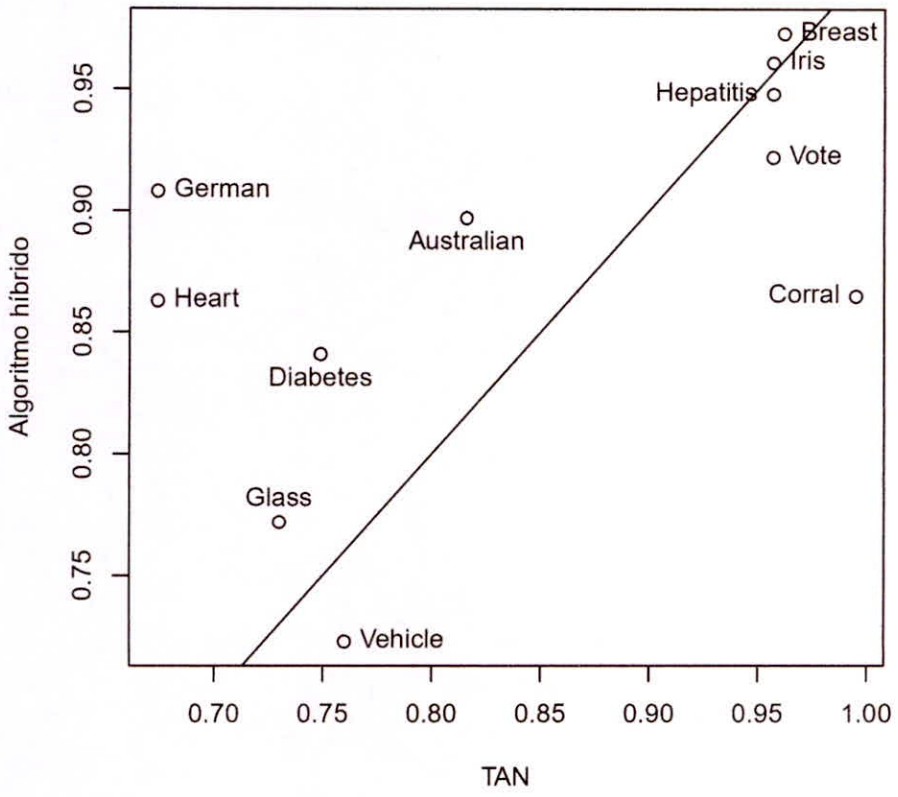


Figura 4.6: Comparación entre TAN y algoritmo híbrido

En el caso de los clasificadores obtenidos con el algoritmo Hill-Climbing también se observa un rendimiento predictivo similar con Naive Bayes. Al realizar la comparación con el clasificador TAN se obtienen nuevamente mejores resultados en términos predictivos, sobre todo en los conjuntos de datos `Australian`, `Breast`, `Diabetes`, `German`, `Glass` y `Heart`.

4.2. Clasificadores con las variables predictoras seleccionadas por SES

En esta sección se construyen los clasificadores propuestos usando las variables predictoras seleccionadas por el algoritmo Statistically Equivalent Signature. En la

Tabla 4.3 se muestra cada conjunto de datos, el número total de variables predictoras y las que fueron seleccionados luego de la aplicación del algoritmo SES.

Tabla 4.3: Variables seleccionadas con el algoritmo SES

N°	Nombre	Variables	Algoritmo SES
1	Australian	14	8
2	Breast	9	4
3	Corral	6	6
4	Diabetes	8	5
5	German	20	4
6	Glass	9	3
7	Heart	13	6
8	Hepatitis	19	3
9	Iris	4	2
10	Vehicle	18	13
11	Vote	16	3

La Tabla 4.4 muestra la tasa de elementos correctamente clasificados para Naive Bayes, TAN y los clasificadores obtenidos con los algoritmos propuestos, previo proceso de selección de variables.

Tabla 4.4: Tasa de elementos correctamente clasificados

N°	Nombre	Naive Bayes	TAN	Restricciones	Scores	Híbridos
1	Australian	0.898	0.802	0.860	0.896	0.896
2	Breast	0.976	0.948	0.971	0.964	0.976
3	Corral	0.865	0.995	0.829	0.865	0.865
4	Diabetes	0.848	0.751	0.832	0.847	0.847
5	German	0.925	0.833	0.899	0.923	0.923
6	Glass	0.721	0.685	0.677	0.716	0.716
7	Heart	0.866	0.774	0.846	0.853	0.868
8	Hepatitis	0.916	0.938	0.921	0.916	0.916
9	Iris	0.972	0.971	0.971	0.971	0.971
10	Vehicle	0.715	0.749	0.743	0.748	0.734
11	Vote	0.962	0.961	0.960	0.960	0.960

Se observa en la Tabla 4.5 que, en la mayoría de los casos, los clasificadores obtenidos con los algoritmos basados en scores e híbridos son los que permiten obtener las mayores tasas de elementos correctamente clasificados. Los conjuntos de datos en los

que la tasa fue mayor en dos de los algoritmos considerados fueron Corral, Diabetes, German, Glass y los conjuntos de datos en los que se obtuvo la misma tasa con los tres algoritmos fueron Iris y Vote.

Tabla 4.5: Comparación entre los clasificadores propuestos luego de aplicar SES

N°	Nombre	Algoritmo
1	Australian	Score e Híbridos
2	Breast	Híbridos
3	Corral	Scores e Híbridos
4	Diabetes	Scores e Híbridos
5	German	Scores e Híbridos
6	Glass	Scores e Híbridos
7	Heart	Híbridos
8	Hepatitis	Restricciones
9	Iris	Restricciones, Scores e Híbridos
10	Vehicle	Scores
11	Vote	Restricciones, Scores e Híbridos

4.2.1. Naive Bayes y TAN versus Grow-Shrink

Los conjuntos de datos en los que el clasificador obtenido con el algoritmo Grow-Shrink presentó una pequeña mejora en el comportamiento predictivo en comparación con Naive Bayes fueron Hepatitis y Vehicle, tal como puede observarse en la Figura 4.7. Al realizar la comparación con el clasificador TAN, el clasificador propuesto obtuvo una tasa de elementos correctamente clasificados ligeramente mayor en Australian, Breast, Diabetes, German y Heart según la Figura 4.8.

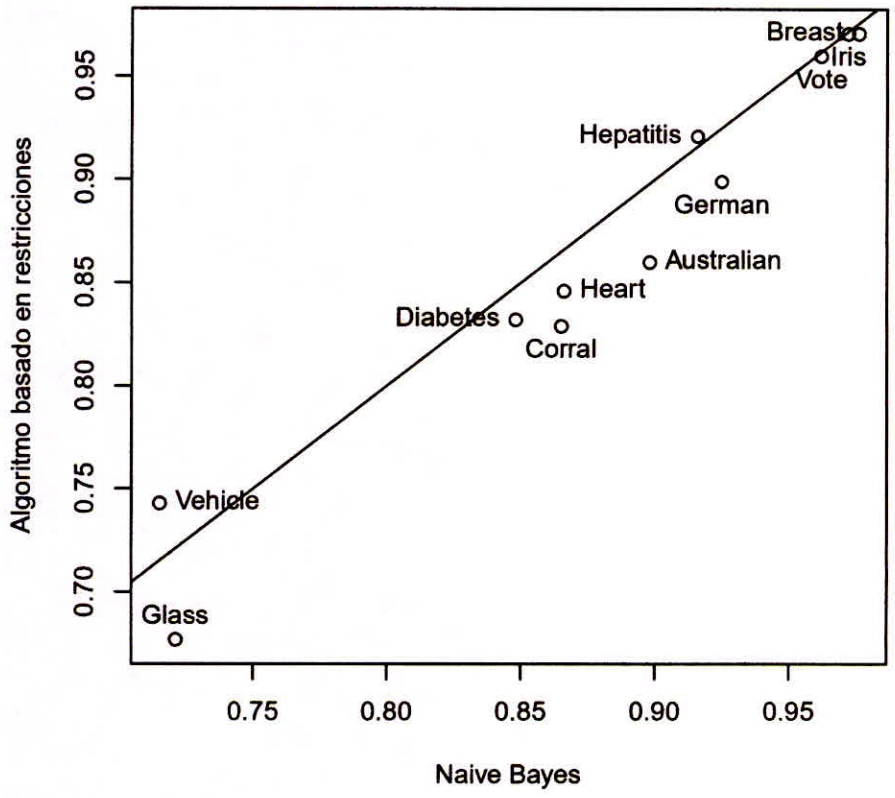


Figura 4.7: Comparación entre Naive Bayes y algoritmo basado en restricciones con SES

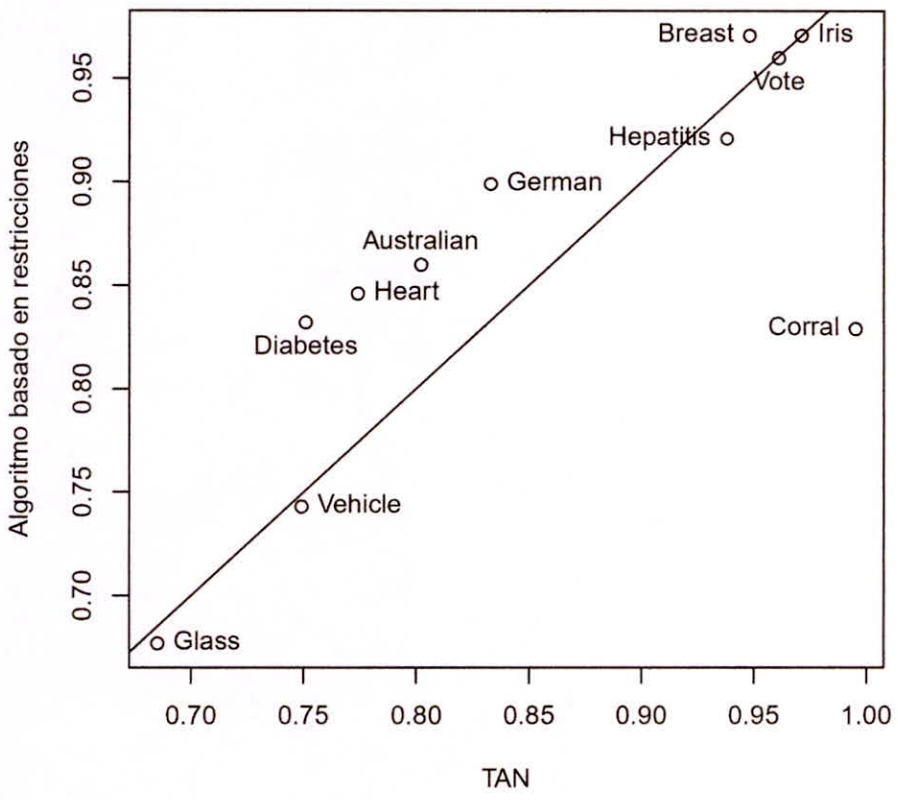


Figura 4.8: Comparación entre TAN y algoritmo basado en restricciones con SES

En resumen, los clasificadores obtenidos con el algoritmo Grow-Shrink, luego del proceso de selección de variables SES, presentan un rendimiento predictivo similar con Naive Bayes. Al realizar la comparación con el clasificador TAN se obtienen mejores resultados en términos predictivos en los conjuntos de datos previamente mencionados.

4.2.2. Naive Bayes y TAN versus Hill-Climbing

En la Figura 4.9 se puede observar que solamente en `Vehicle` el clasificador obtenido con el algoritmo Hill-Climbing tuvo una tasa de elementos correctamente clasificados ligeramente mayor en comparación con Naive Bayes. En los conjuntos de datos `Australian`, `Breast`, `Diabetes`, `German`, `Glass` y `Heart` se observa que el clasificador

propuesto tuvo un mejor rendimiento predictivo en comparación con TAN, según la Figura 4.10.

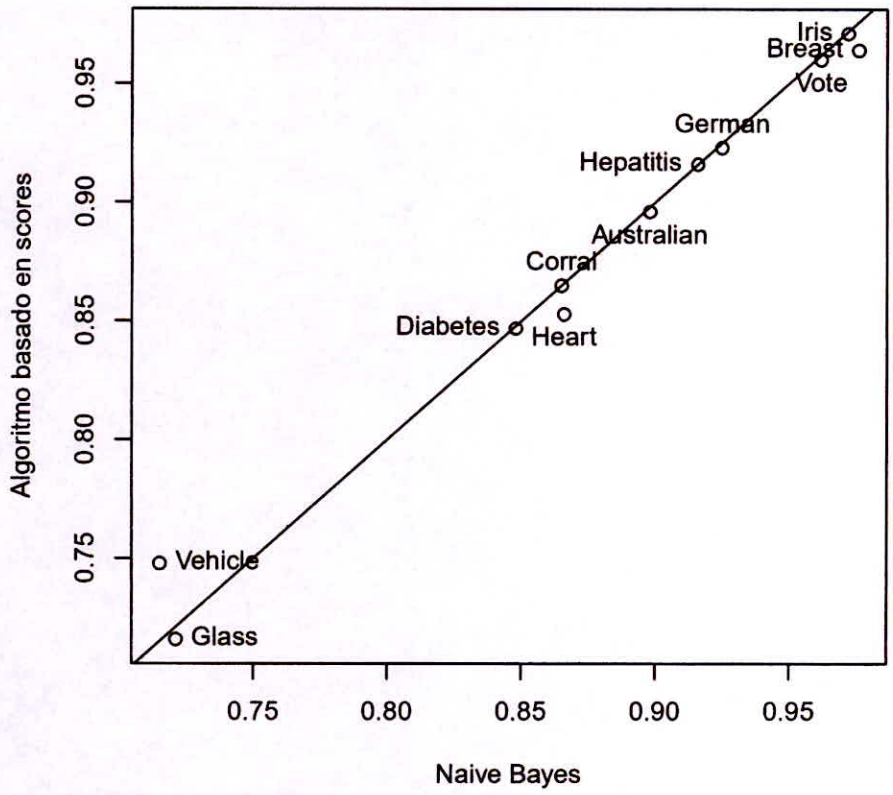


Figura 4.9: Comparación entre Naive Bayes y algoritmo basado en scores con SES

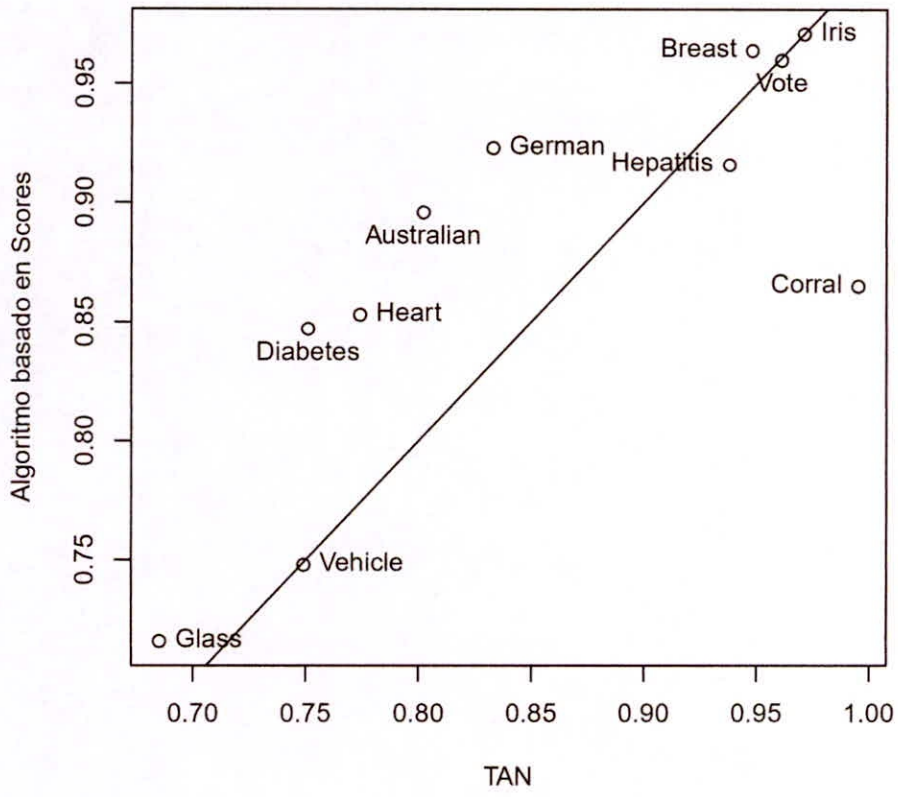


Figura 4.10: Comparación entre TAN y algoritmo basado en scores con SES

Los clasificadores obtenidos con el algoritmo Hill-Climbing, luego del proceso de selección de variables SES, también presentan un rendimiento predictivo similar con Naive Bayes. Al realizar la comparación con el clasificador TAN se obtienen nuevamente mejores resultados en términos predictivos, sobre todo en los conjuntos de datos `Australian`, `Diabetes`, `German` y `Heart`.

4.2.3. Naive Bayes y TAN versus Max-Min Padres e Hijos

El clasificador obtenido con el algoritmo Max-Min Padres e Hijos tuvo un comportamiento predictivo ligeramente mayor en comparación a Naive Bayes solo en el conjunto de datos `Vehicle` tal como se observa en la Figura 4.11. Los conjuntos de datos en los que el clasificador propuesto presenta una mayor tasa de elementos

correctamente clasificados en comparación al clasificador TAN fueron Australian, Breast, Diabetes, German, Glass y Heart según se observa en la Figura 4.12.

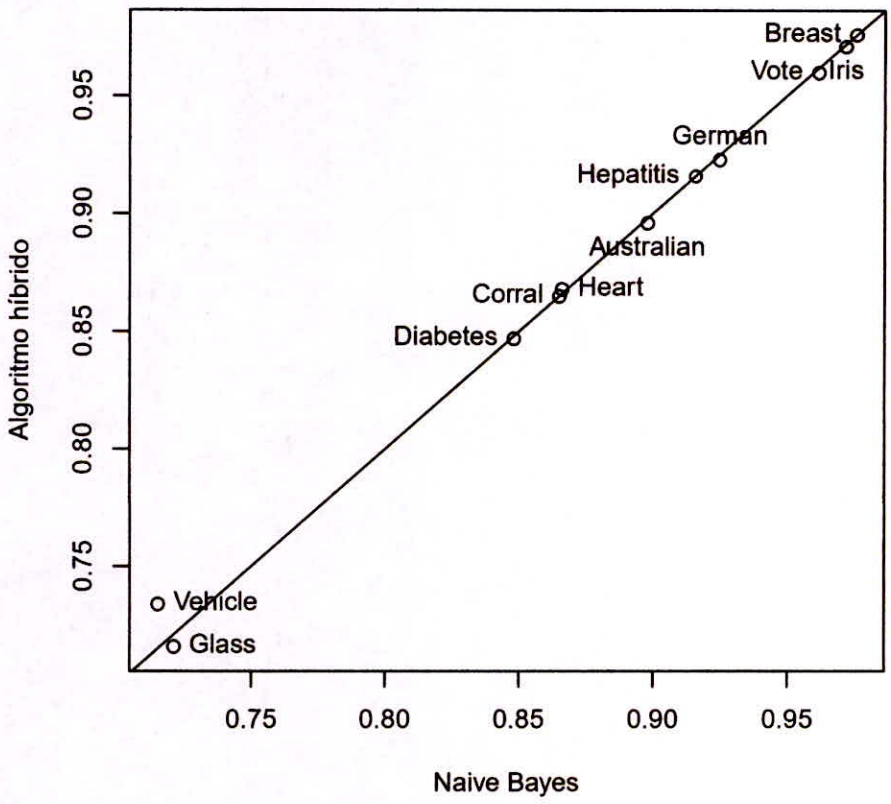


Figura 4.11: Comparación entre Naive Bayes y algoritmo híbrido con SES

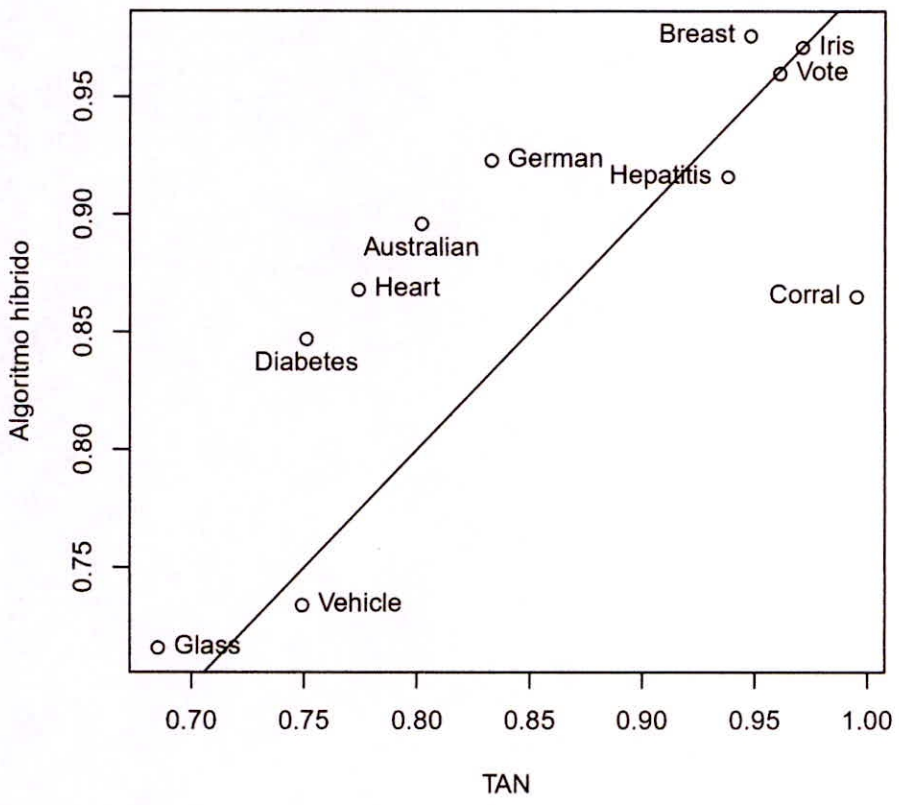


Figura 4.12: Comparación entre TAN y algoritmo híbrido con SES

En el caso de los clasificadores obtenidos con el algoritmo Hill-Climbing también se observa un rendimiento predictivo muy similar al obtenido con Naive Bayes. Al realizar la comparación con el clasificador TAN se obtienen nuevamente mejores resultados en términos predictivos en los conjuntos de datos previamente mencionados.

4.3. Comparación entre los clasificadores antes y después de aplicar el algoritmo SES

En esta sección se comparan los clasificadores construidos con todas las variables predictoras y los clasificadores obtenidos luego de aplicar el algoritmo de selección de variables SES. Se consideran los tres algoritmos propuestos para la construcción de la estructura presente entre las variables predictoras. Los resultados obtenidos en las secciones anteriores se resumen en la Tabla 4.6.

Tabla 4.6: Tasa de elementos correctamente clasificados

N°	Nombre	Restricciones		Scores		Híbridos	
		Sin SES	Con SES	Sin SES	Con SES	Sin SES	Con SES
1	Australian	0.895	0.860	0.896	0.896	0.897	0.896
2	Breast	0.973	0.971	0.966	0.964	0.973	0.976
3	Corral	0.829	0.829	0.865	0.865	0.865	0.865
4	Diabetes	0.836	0.832	0.833	0.847	0.841	0.847
5	German	0.919	0.899	0.908	0.923	0.908	0.923
6	Glass	0.772	0.677	0.760	0.716	0.772	0.716
7	Heart	0.860	0.846	0.838	0.853	0.863	0.868
8	Hepatitis	0.947	0.921	0.947	0.916	0.948	0.916
9	Iris	0.939	0.971	0.952	0.971	0.961	0.971
10	Vehicle	0.723	0.743	0.785	0.748	0.723	0.734
11	Vote	0.941	0.960	0.950	0.960	0.922	0.960

4.3.1. Algoritmo Grow-Shrink

En la Figura 4.13 se observa que el proceso de selección de variables con el algoritmo SES permite obtener una tasa de elementos correctamente clasificados ligeramente mejor solo en los conjuntos de datos `Iris`, `Vehicle` y `Vote` cuando el algoritmo usado para obtener la estructura del GAD es Grow-Shrink.

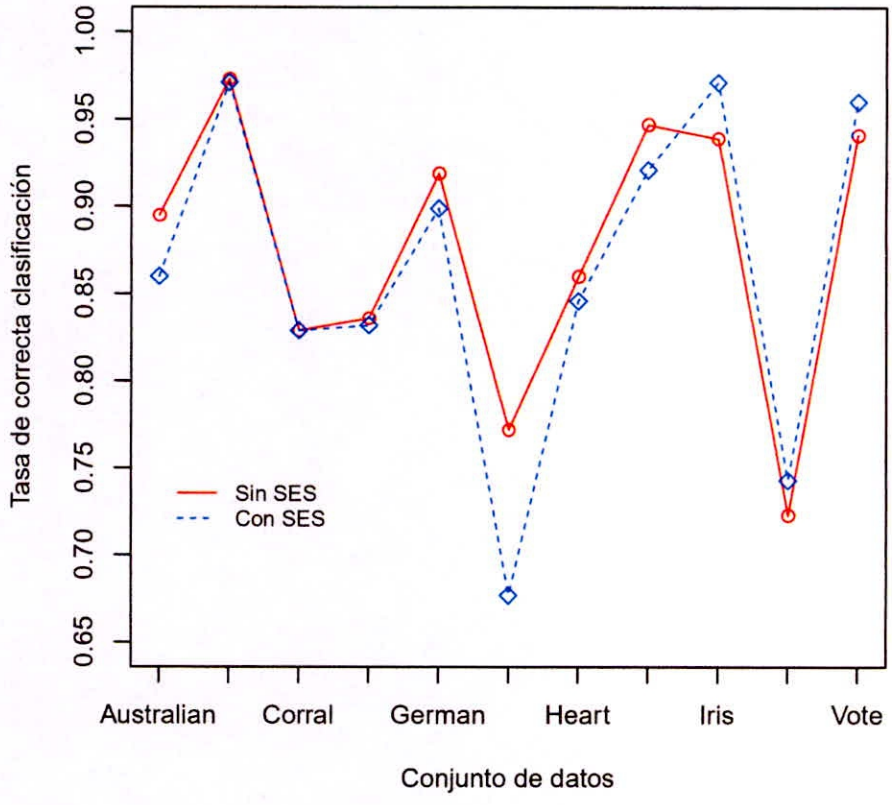


Figura 4.13: Algoritmo Grow-Shrink antes y después de aplicar SES

4.3.2. Algoritmo Hill-Climbing

Por otro lado, en la Figura 4.14 se observa una tasa de elemenos correctamente clasificados ligeramente mayor luego de aplicar el algoritmo SES en los conjuntos de datos `Diabetes`, `German`, `Heart`, `Iris` y `Vote` cuando el algoritmo usado para obtener la estructura del GAD es Hill-Climbing.

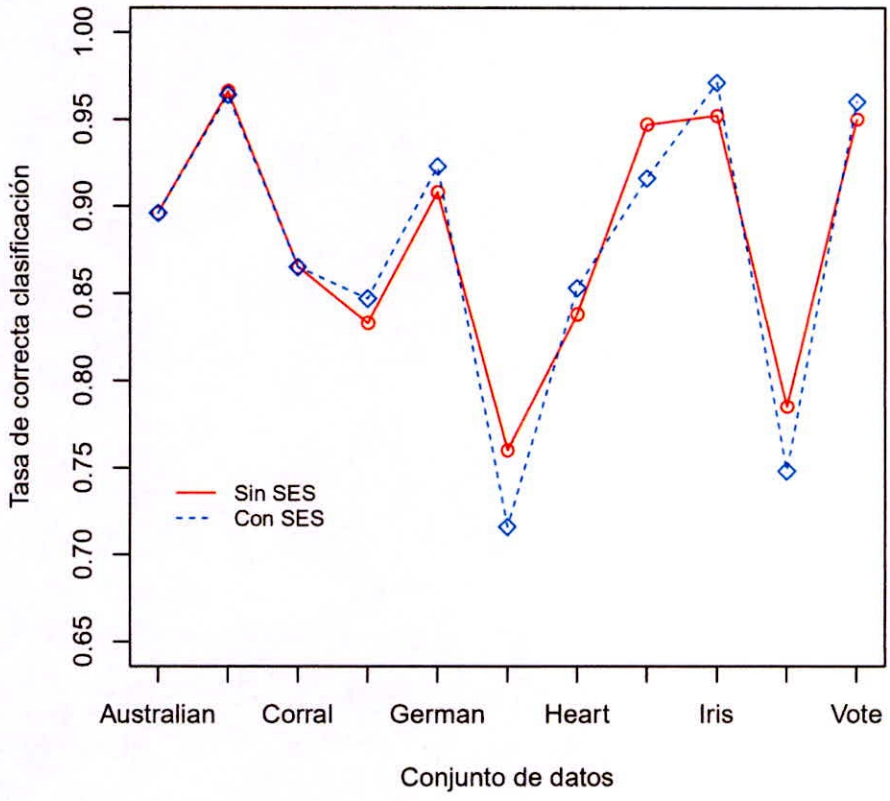


Figura 4.14: Algoritmo Hill-Climbing antes y después de aplicar SES

4.3.3. Algoritmo Max-Min Padres e Hijos

Al usar el algoritmo Max-Min Padres e Hijos se observa en la Figura 4.15 que, luego de aplicar el algoritmo SES, hay una tasa de elementos correctamente clasificados ligeramente mayor en los conjuntos de datos `Breast`, `Diabetes`, `German`, `Heart`, `Iris` y `Vote`.

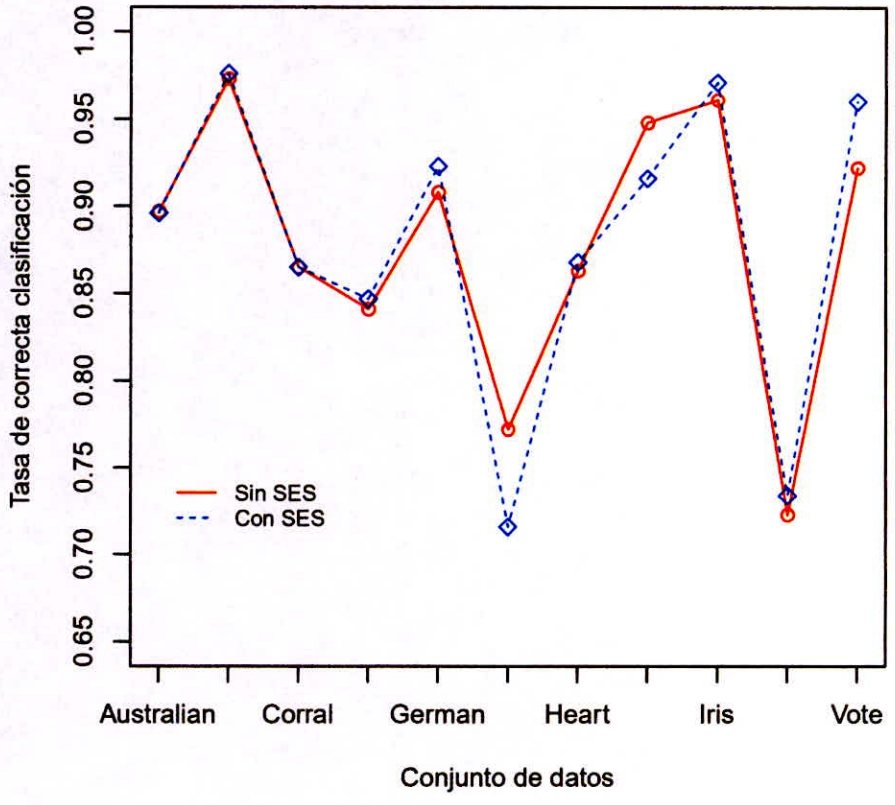


Figura 4.15: Algoritmo Max-Min Padres e Hijos antes y después de aplicar SES

4.4. Caso de aplicación: Encuesta Nacional de Innovación en la Industria Manufacturera 2015

El caso de aplicación se realizó usando los datos correspondientes a la Encuesta Nacional de Innovación Manufacturera 2015, analizados por Javier Fernando Del Carpio Gallegos, profesor de la Universidad ESAN. Se seleccionaron aleatoriamente 826 empresas de un total de 1452 y se obtuvo un modelo de regresión probit para analizar si las empresas peruanas de manufactura realizan de forma efectiva el proceso de innovación en producto, que se define como la introducción de un bien o servicio nuevo, o significativamente mejorado, en cuanto a sus características o en cuanto al uso destinado. Las variables consideradas en la construcción del modelo probit se muestran en la Tabla 4.7.

Tabla 4.7: Variables en la Encuesta Nacional de Innovación Manufacturera 2015

Variable	Descripción	Tipo
Y	Innovación de producto	Cualitativa
X_1	Fuente externa de conocimiento de mercado	Cualitativa
X_2	Fuente externa de conocimiento tecnológico	Cualitativa
X_3	Intensidad tecnológica interna	Cuantitativa
X_4	Tamaño de la empresa	Cuantitativa
X_5	Adquisición tecnológica	Cuantitativa
X_6	Capacitación	Cuantitativa
X_7	Capital humano	Cuantitativa

Las variables predictoras X_1 y X_2 toman dos posibles valores: 1 = Si y 0 = No. La variable X_3 se define como la proporción del gasto de la innovación y desarrollo interno entre las ventas totales anuales, X_4 es el logaritmo del número de empleados, X_5 se define como el logaritmo de los gastos en adquisición de maquinaria, hardware y software, X_6 es el logaritmo en los gastos en la actividad de capacitación del personal para actividades de innovación y X_7 se define como la proporción del personal científico e investigador entre el total de empleados. La tasa de elementos correctamente clasificados obtenidos con el modelo probit, usando las variables predictoras mencionadas, fue del 70.81 %.

A continuación se realiza el proceso de construcción de los clasificadores usando las metodologías propuestas. En la etapa de procesamiento al aplicar el método de

discretización Chi-Merge se obtuvo solo un intervalo para la variable X_5 = Adquisición tecnológica. En este caso se procedió a retirar dicha variable antes de usar los algoritmos de construcción de la estructura.

Las tasas de elementos correctamente clasificados obtenidas antes y después de aplicar el algoritmo SES son iguales para los tres algoritmos estudiados, Naive Bayes y TAN según la Tabla 4.8. En este conjunto de datos particular, el algoritmo de selección no distingue grupos de variables con comportamiento predictivo similar, por lo que los clasificadores se construyen usando todas las variables predictoras disponibles. En cualquiera de los casos la tasa de elementos correctamente clasificados es mayor en comparación al obtenido con la regresión probit.

Tabla 4.8: Tasa de elementos correctamente clasificados

Algoritmo	Sin SES	Con SES
Grow-Shrink	0.727	0.727
Hill-Climbing	0.728	0.728
Mix.Max Padres e Hijos	0.728	0.728
Naive Bayes	0.729	0.729
TAN	0.725	0.725

La estructura obtenida con el algoritmo Grow-Shrink define las siguientes relaciones de dependencia: la variable X_2 = Fuente externa de conocimiento tecnológico; depende de la variable X_1 = Fuente externa de conocimiento de mercado, la variable X_3 = Intensidad tecnológica interna; depende de las variables X_4 = Tamaño de la empresa y X_7 = Capital humano, la variable X_4 = Tamaño de la empresa; depende de la variable X_7 = Capital humano y la variable X_6 = Capacitación; depende de las variables X_1 = Fuente externa de conocimiento de mercado y X_3 = Intensidad tecnológica interna. Sobre la base de estas relaciones es posible simplificar la función de probabilidad conjunta de las variables predictoras condicionadas con la variable de clase Y usando las distribuciones locales:

$$\begin{aligned}\Pr(X_1, X_2, X_3, X_4, X_6, X_7|Y) &= \Pr(X_1|Y)\Pr(X_2|X_1, Y)\Pr(X_3|X_4, X_7, Y)\Pr(X_4|X_7, Y) \\ &\quad \Pr(X_6|X_1, X_3, Y)\Pr(X_7|Y)\end{aligned}$$

El clasificador obtenido con el primer algoritmo se muestra en la Figura 4.16.

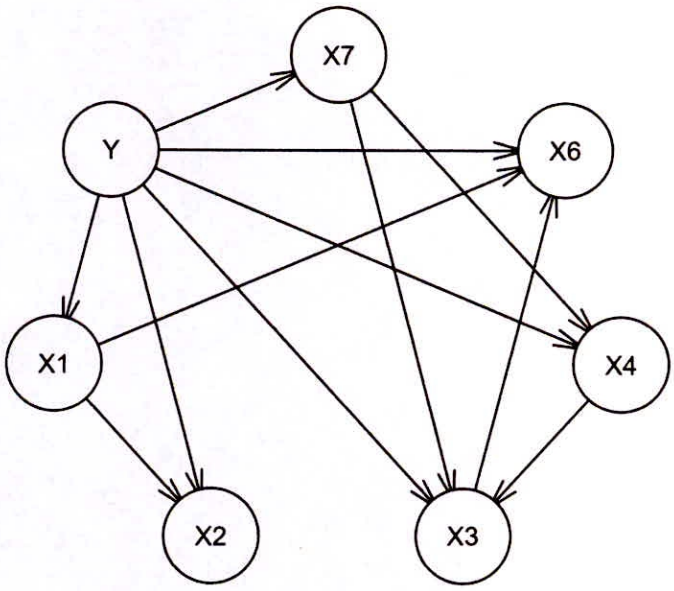


Figura 4.16: Clasificador Grow-Shrink con la data Innovación

Las tablas de probabilidad para las distribuciones locales, obtenidas usando el estimador Bayesiano, se presentan a continuación:

Tabla 4.9: Tabla de probabilidad para Y

Y	
0	1
0.5701	0.4299

Tabla 4.10: Tabla de probabilidad condicional para X_1

	Y	
X_1	0	1
0	0.1744	0.0598
1	0.8256	0.9402

Tabla 4.11: Tablas de probabilidad condicional para X_2

$X_1 = 0$	Y	
X_2	0	1
0	0.8039	0.3353
1	0.1961	0.6647

$X_1 = 1$	Y	
X_2	0	1
0	0.1159	0.0871
1	0.8841	0.9129

Tabla 4.12: Tablas de probabilidad condicional para X_3

$X_4 = 0, X_7 = 0$	Y	
X_3	0	1
0	0.9996	0.9481
1	0.0004	0.0519

$X_4 = 1, X_7 = 0$	Y	
X_3	0	1
0	0.9800	0.6885
1	0.0200	0.3115

$X_4 = 0, X_7 = 1$	Y	
X_3	0	1
0	0.9992	0.7949
1	0.0008	0.2051

$X_4 = 1, X_7 = 1$	Y	
X_3	0	1
0	0.8513	0.4932
1	0.1487	0.5068

$X_4 = 0, X_7 = 2$	Y	
X_3	0	1
0	0.9265	0.7240
1	0.0735	0.2760

$X_4 = 1, X_7 = 2$	Y	
X_3	0	1
0	0.7871	0.3336
1	0.2129	0.6664

Tabla 4.13: Tablas de probabilidad condicional para X_4

$X_7 = 0$	Y	
X_4	0	1
0	0.5143	0.3082
1	0.4857	0.6918

$X_7 = 1$	Y	
X_4	0	1
0	0.4857	0.3762
1	0.5143	0.6238

$X_7 = 2$	Y	
X_4	0	1
0	0.7881	0.6704
1	0.2119	0.3296

Tabla 4.14: Tablas de probabilidad condicional para X_6

$X_1 = 0, X_3 = 0$	Y	
X_6	0	1
0	0.9736	0.7975
1	0.0264	0.2025

$X_1 = 1, X_3 = 0$	Y	
X_6	0	1
0	0.9317	0.7558
1	0.0683	0.2442

$X_1 = 0, X_3 = 1$	Y	
X_6	0	1
0	0.7424	0.5000
1	0.2576	0.5000

$X_1 = 1, X_3 = 1$	Y	
X_6	0	1
0	0.4548	0.4081
1	0.5452	0.5919

Tabla 4.15: Tabla de probabilidad condicional para X_7

	Y	
X_7	0	1
0	0.4457	0.1833
1	0.2230	0.3296
2	0.3313	0.4871

La estructura obtenida con el algoritmo Hill-Climbing y Mix-Max Padres e Hijos es la misma y define las siguientes relaciones de dependencia: la variable X_1 = Fuente externa de conocimiento tecnológico; depende de la variable X_6 = Capacitación, la variable X_2 = Fuente externa de conocimiento tecnológico; depende de la variable X_1 = Fuente externa de conocimiento de mercado, la variable X_4 = Tamaño de la empresa; depende de la variable X_3 = Intensidad tecnológica interna, la variable X_6 = Capacitación; depende de la variable X_3 = Intensidad tecnológica interna y la variable X_7 = Capital humano; depende de las variables X_3 = Intensidad tecnológica interna y X_4 = Tamaño de la empresa. Sobre la base de estas relaciones es posible simplificar la función de probabilidad conjunta de las variables predictoras condicionadas con la variable de clase Y usando las distribuciones locales:

$$\begin{aligned}\Pr(X_1, X_2, X_3, X_4, X_6, X_7|Y) &= \Pr(X_1|X_6, Y)\Pr(X_2|X_1, Y)\Pr(X_3|Y)\Pr(X_4|X_3, Y) \\ &\quad \Pr(X_6|X_3, Y)\Pr(X_7|X_3, X_4Y)\end{aligned}$$

El clasificador obtenido con los dos últimos algoritmos se muestra en la Figura 4.17

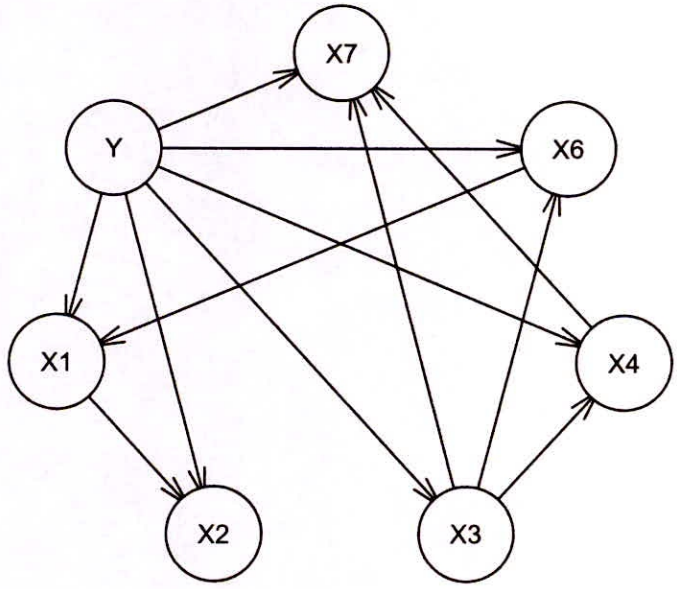


Figura 4.17: Algoritmos Hill-Climbing y Mix-Max Padres e Hijos con la data Innovación

Las tablas de probabilidad para las distribuciones locales, obtenidas usando el estimador Bayesiano, se presentan a continuación:

Tabla 4.16: Tabla de probabilidad para Y

Y	
0	1
0.5701	0.4299

Tabla 4.17: Tablas de probabilidad condicional para X_1

$X_6 = 0$	Y	
X_1	0	1
0	0.1835	0.0674
1	0.8165	0.9326

$X_6 = 1$	Y	
X_1	0	1
0	0.0776	0.0467
1	0.9224	0.9533

Tabla 4.18: Tablas de probabilidad condicional para X_2

$X_1 = 0$	Y	
X_2	0	1
0	0.8040	0.3353
1	0.1960	0.6647

$X_1 = 1$	Y	
X_2	0	1
0	0.1159	0.0871
1	0.8841	0.9129

Tabla 4.19: Tablas de probabilidad condicional para X_3

	Y	
X_3	0	1
0	0.9443	0.6308
1	0.0557	0.3692

Tabla 4.20: Tablas de probabilidad condicional para X_4

$X_3 = 0$	Y	
X_4	0	1
0	0.6134	0.6159
1	0.3866	0.3841

$X_3 = 1$	Y	
X_4	0	1
0	0.3476	0.3209
1	0.6524	0.6791

Tabla 4.21: Tablas de probabilidad condicional para X_6

$X_3 = 0$	Y	
X_6	0	1
0	0.9391	0.7586
1	0.0609	0.2414

$X_3 = 1$	Y	
X_6	0	1
0	0.5000	0.4123
1	0.5000	0.5877

Tabla 4.22: Tablas de probabilidad condicional para X_7

$X_3 = 0, X_4 = 0$	Y	
X_7	0	1
0	0.3956	0.1379
1	0.1869	0.2537
2	0.4175	0.6084

$X_3 = 1, X_4 = 0$	Y	
X_7	0	1
0	0.0046	0.0247
1	0.0046	0.2147
2	0.9908	0.7606

$X_3 = 0, X_4 = 1$	Y	
X_7	0	1
0	0.5812	0.3604
1	0.2675	0.4185
2	0.1513	0.2211

$X_3 = 1, X_4 = 1$	Y	
X_7	0	1
0	0.1192	0.1576
1	0.4696	0.4156
2	0.4112	0.4268

Capítulo 5

Conclusiones y sugerencias

5.1. Conclusiones

1. Las redes Bayesianas permiten representar las relaciones de dependencia e independencia condicional entre un conjunto de variables predictoras a través de los algoritmos presentados, utilizando de manera individual o combinada pruebas de independencia condicional entre variables y medidas de scores sobre la estructura obtenida. De esta forma es posible simplificar el cálculo de las probabilidades de interés, usando las distribuciones locales a partir de una estructura sencilla y que es mucho más fácil de manejar en términos computacionales. Por esta razón las redes Bayesianas pueden ser utilizadas en muchos otras aplicaciones, mas allá de la construcción de clasificadores.

2. Los clasificadores por redes Bayesianas construidos con los algoritmos basados en restricciones, scores e híbridos presentan un buen comportamiento predictivo, según los valores de la Tabla 4.4. Al ser comparados con los clasificadores tradicionales se obtuvieron tasas de elementos correctamente clasificados muy cercanos a los obtenidos con Naive Bayes y ligeramente superiores en el caso de TAN. Sin embargo, los clasificadores obtenidos tienen la ventaja de estar construidos sobre estructuras que representan las relaciones existentes entre las variables predictoras y en este caso los clasificadores propuestos tienen, en términos generales, un mejor comportamiento predictivo que TAN. El algoritmo basado en restricciones tiene la desventaja de obtener estructuras donde algunos de los arcos podrían no estar dirigidos. Lo anterior supone la tarea previa

de decidir la dirección de los arcos antes de construir los clasificadores. Sin embargo, la elección es arbitraria y no determina cambios importantes ni en las medidas de score calculadas sobre la estructura obtenida, ni sobre la tasa de elementos correctamente clasificados. El esfuerzo computacional requerido por los algoritmos propuestos para obtener la estructura entre variables predictoras, y posterior construcción del clasificador, es ligeramente mayor al que se requiere con los clasificadores tradicionales Naive Bayes y TAN. Las librerías `bnlearn` , `dprep` y `MXM` no presentaron inconveniente alguno al trabajar con los conjuntos de datos usados en el presente trabajo de investigación.

3. El algoritmo de selección de variables Statistically Equivalent Signatures, utilizado antes de la aplicación de los algoritmos de estimación de la estructura, permitió obtener clasificadores sobre una menor cantidad de variables predictoras. Sin embargo, no se obtiene una diferencia importante en la tasa de elementos correctamente clasificados en comparación con los clasificadores construidos con todas las variables predictoras según los valores de la Tabla 4.6. Por otro lado, elegir las variables predictoras a utilizar en una etapa inicial permite un menor gasto computacional al momento de aplicar los algoritmos propuestos.

5.2. Sugerencias y trabajo futuro

1. Los clasificadores usados en el presente trabajo de investigación requieren una etapa inicial de discretización de las variables cuantitativas continuas para llevar a cabo el proceso de estimación de la estructura a partir de los algoritmos propuestos. Sin embargo, esto puede llevarnos a una inevitable pérdida de información por lo que es importante considerar metodologías que permitan trabajar con ambos tipos de variables.

2. Es posible usar las redes Bayesianas Gaussianas para los conjuntos de datos que tienen solamente variables cuantitativas continuas. En este tipo de redes, se asume que la distribución global es normal multivariada y que cada distribución local se puede expresar como un modelo lineal Gaussiano clásico de regresión donde el nodo es la variable respuesta y sus padres son las variables explicativas.

3. Las redes Bayesianas mixtas permiten trabajar con variables discretas y continuas, pudiendo utilizar casi cualquier modelo de probabilidad para la

distribuciones locales dentro de lo razonable. Desafortunadamente, esta mayor flexibilidad hace que la red Bayesiana sea más compleja. No existe en R alguna librería que permita manejar este tipo de redes Bayesianas por lo que el proceso de estimación de la estructura y de los parámetros requieren de un esfuerzo de programación por parte del usuario.

4. Los algoritmos propuestos en el presente trabajo permiten estimar la estructura de la red Bayesiana sin considerar información a priori acerca de las relaciones de dependencia entre las variables predictoras. Si se tiene inicialmente un conjunto de relaciones conocidas, éstas pueden ser incluidas en cualquiera de los algoritmos propuestos como arcos fijos. El incluir estos arcos, como parte del conocimiento experto que se tiene sobre el problema que se analiza, puede llevarnos a obtener clasificadores con mejores rendimientos.

5. A lo largo de este trabajo se han definido las redes Bayesanas en términos de relaciones de dependencia e independencia condicional sin considerar que los arcos deben representar relaciones de causa y efecto. La existencia de clases de equivalencia de las redes, indistinguibles desde el punto de vista probabilístico, proporciona una prueba sencilla que las direcciones de los arcos no son indicativos de los efectos causales Sin embargo, desde un punto de vista intuitivo se puede argumentar que una buena red Bayesiana debe representar la estructura causal de los datos que está describiendo. Construir una red Bayesiana a partir del conocimiento experto en la práctica codifica conocimiento y relaciones causales esperadas para un fenómeno dado.

Capítulo 6

Anexos

6.1. Ejemplo: Encuesta Nacional de Innovación

```
> ########################################################################
> ############################ Funcion para crear el clasificador
> ########################################################################
>
> clasificador <- function(data, bn, class){
+   gad <- empty.graph(nodes = names(data))
+   arcs(gad) <- directed.arcs(bn)
+   nodes <- names(data)[-class]
+   for (i in 1:length(nodes)){
+     gad <- set.arc(gad, from = names(data)[class], to = nodes[i])
+   }
+   return(gad)
+ }
> ########################################################################
> ############################ Innovacion
> ########################################################################
>
> Innovacion.data <- read.table(file = "D:/Tesis/Innovacion.txt",
+ header = TRUE)
> library(dprep)
> Innovacion.data.d <- chiMerge(data = Innovacion.data, varcon = 4:8,
```

```
+ alpha = 0.05, out = "symb")
> Innovacion.data.d <- Innovacion.data.d[, -6]
> write.table(x = Innovacion.data.d, file = "D:/Tesis/Innovacion.data.d.txt",
+ row.names = FALSE)
> Innovacion.data.d <- read.table(file = "D:/Tesis/Innovacion.data.d.txt",
+ header = TRUE)
> for (i in 1:7){
+   Innovacion.data.d[, i] <- as.factor(Innovacion.data.d[, i])
+ }
> ############################## Algoritmos SES
>
> library(MXM)
> Innovacion.SES <- SES(target = "Y", dataset = Innovacion.data.d, max_k = 3,
+ threshold = 0.05, test = "testIndMultinom")
> Innovacion.SES
> ############################## Algoritmos basados en restricciones
>
> library(bnlearn)
> Innovacion.gs <- gs(Innovacion.data.d[, 2:7], test = "mi", alpha = 0.05,
+ undirected = FALSE)
> plot(Innovacion.gs)
> undirected.arcs(Innovacion.gs)
> directed.arcs(Innovacion.gs)
> Innovacion.gs <- empty.graph(nodes = c(names(Innovacion.data.d)[2:7]))
> arc.set <- matrix(c("X1", "X2", "X1", "X6", "X3", "X6", "X4", "X3",
+ "X7", "X3", "X7", "X4"), byrow = T, ncol = 2, dimnames = list(NULL,
+ c("from", "to")))
> arcs(Innovacion.gs) <- arc.set
> plot(Innovacion.gs)
> Innovacion.gs.clasif <- clasificador(data = Innovacion.data[, -6],
+ bn = Innovacion.gs, class = 1)
> plot(Innovacion.gs.clasif)
> set.seed(100)
> bn.cv(data = Innovacion.data.d, bn = Innovacion.gs.clasif, k = 10,
+ fit = "bayes", loss = "pred-lw", loss.args = list(target = "Y"), runs = 20)
> ############################## Algoritmos basados en scores
```

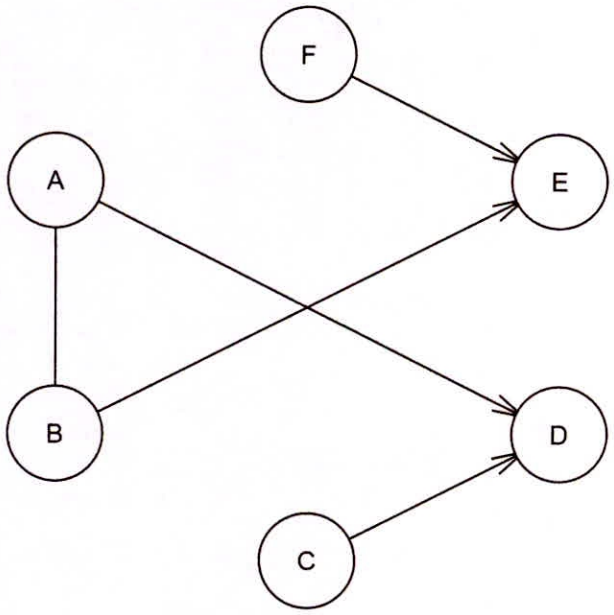
```
>
> Innovacion.hc <- hc(Innovacion.data.d[, 2:7], score = "bic")
> plot(Innovacion.hc)
> Innovacion.hc.clasif <- clasificador(data = Innovacion.data[, -6],
+ bn = Innovacion.hc, class = 1)
> plot(Innovacion.hc.clasif)
> set.seed(100)
> bn.cv(data = Innovacion.data.d, bn = Innovacion.hc.clasif, k = 10,
+ fit = "bayes", loss = "pred-lw", loss.args = list(target = "Y"), runs = 20)
> ############################## Algoritmos hibridos
>
> Innovacion.mmhc <- mmhc(Innovacion.data.d[, 2:7])
> plot(Innovacion.mmhc)
> Innovacion.mmhc.clasif <- clasificador(data = Innovacion.data[, -6],
+ bn = Innovacion.mmhc, class = 1)
> plot(Innovacion.mmhc.clasif)
> set.seed(100)
> bn.cv(data = Innovacion.data.d, bn = Innovacion.mmhc.clasif, k = 10,
+ fit = "bayes", loss = "pred-lw", loss.args = list(target = "Y"), runs = 20)
> ############################## Naive Bayes
>
> Innovacion.naive.clasif <- naive.bayes(Innovacion.data.d, "Y")
> plot(Innovacion.naive.clasif)
> set.seed(100)
> bn.cv(data = Innovacion.data.d, bn = Innovacion.naive.clasif, k = 10,
+ fit = "bayes", loss = "pred-lw", loss.args = list(target = "Y"), runs = 20)
> ############################## TAN
>
> Innovacion.TAN.clasif <- tree.bayes(Innovacion.data.d, "Y")
> plot(Innovacion.TAN.clasif)
> set.seed(100)
> bn.cv(data = Innovacion.data.d, bn = Innovacion.TAN.clasif, k = 10,
+ fit = "bayes", loss = "pred-lw", loss.args = list(target = "Y"), runs = 20)
```

6.2. Ejemplo: Algoritmo Grow-Shrink

```
> ######################################
> ######## Algoritmo Grow-Shrink ########
>
> learning.test.gs <- gs(learning.test, test = "mi", alpha = 0.05,
+ undirected = FALSE)
> plot(learning.test.gs)
> ##############################
> ######## Primer paso ########
>
> #### A es dependiente de B
> ci.test("A", "B", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  A ~ B
mi = 2341.8, df = 4, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0

> #### A es dependiente de D dado B
> ci.test("A", "D", "B", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  A ~ D | B
mi = 1809.7, df = 12, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0

> #### A es dependiente de C dados B y D
> ci.test("A", "C", c("B", "D"), test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  A ~ C | B + D
mi = 1323.5, df = 36, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0

> ##############################
```

```
> ######## Segundo paso ########
>
> #### A es dependiente de B dado E
> ci.test("A", "B", "E", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  A ~ B | E
mi = 1883.3, df = 12, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0

> #### A es dependiente de B dado F
> ci.test("A", "B", "F", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  A ~ B | F
mi = 2348.6, df = 8, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0

> #### A es dependiente de D
> ci.test("A", "D", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  A ~ D
mi = 2290.2, df = 4, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0

> #### A es dependiente de D dado C
> ci.test("A", "D", "C", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  A ~ D | C
mi = 3980.1, df = 12, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0

> #### A es independiente de C
> ci.test("A", "C", test = "mi", data = learning.test)
```

```
        Mutual Information (disc.)

data:  A ~ C
mi = 1.3091, df = 4, p-value = 0.8598
alternative hypothesis: true value is greater than 0

> ##############################
> ######## Tercer paso #########
>
> #### A es dependiente de C dado D
> ci.test("A", "C", "D", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  A ~ C | D
mi = 1691.2, df = 12, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0

> #### B es dependiente de F dado E
> ci.test("B", "F", "E", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  B ~ F | E
mi = 247.57, df = 6, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0
```



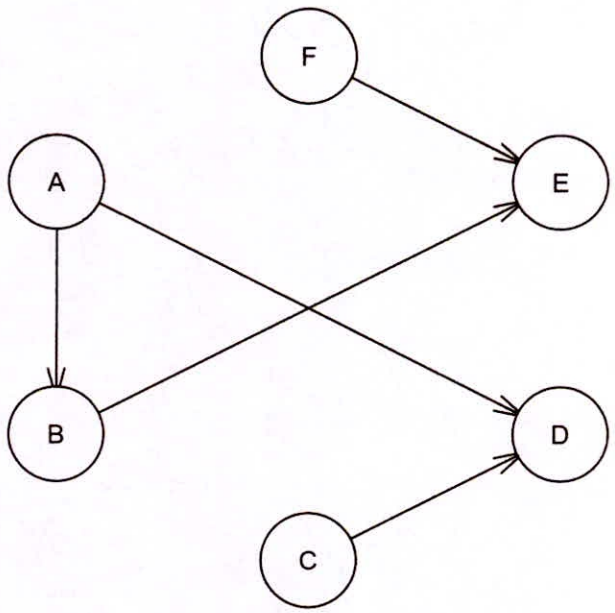
6.3. Ejemplo: Algoritmo Hill-Climbing

```
> ##############################
> ######## Algoritmo HC ########
>
> learning.test.hc <- hc(learning.test, score = "bic")
> plot(learning.test.hc)
> score <- bnlearn::score
> ##############################
> ######## Primer paso ########
>
> gad.0 <- empty.graph(nodes = c("A", "B", "C", "D", "E", "F"))
> score.gad.0 <- score(gad.0, type = "bic", data = learning.test)
```

```
> score.gad.0

[1] -28277.59

> #### Agregar A --> B
> gad.AB <- set.arc(gad.0, from = "A", to = "B")
> score.gad.AB <- score(gad.AB, type = "bic", data = learning.test)
> score.gad.AB - score.gad.0

[1] 1153.88

> #### Agregar A --> C
> gad.AC <- set.arc(gad.0, from = "A", to = "C")
> score.gad.AC <- score(gad.AC, type = "bic", data = learning.test)
> score.gad.AC - score.gad.0

[1] -16.37985

> #### Agregar A --> D
> gad.AD <- set.arc(gad.0, from = "A", to = "D")
> score.gad.AD <- score(gad.AD, type = "bic", data = learning.test)
> score.gad.AD - score.gad.0

[1] 1128.077

> #### Agregar A --> E
> gad.AE <- set.arc(gad.0, from = "A", to = "E")
> score.gad.AE <- score(gad.AE, type = "bic", data = learning.test)
> score.gad.AE - score.gad.0

[1] 216.0217

> #### Agregar A --> F
> gad.AF <- set.arc(gad.0, from = "A", to = "F")
> score.gad.AF <- score(gad.AF, type = "bic", data = learning.test)
> score.gad.AF - score.gad.0

[1] -7.866781

> ##############################
> ######## Segundo paso ########
>
```

```
> #### Agregar A --> D
> gad.AB.AD <- set.arc(gad.AB, from = "A", to = "D")
> score.gad.AB.AD <- score(gad.AB.AD, type = "bic", data = learning.test)
> score.gad.AB.AD - score.gad.AB

[1] 1128.077

> #### Eliminar A --> B
> score.gad.0 - score.gad.AB

[1] -1153.88

> #### Invertir A --> B
> gad.BA <- set.arc(gad.0, from = "B", to = "A")
> score.gad.BA <- score(gad.BA, type = "bic", data = learning.test)
> score.gad.BA - score.gad.AB

[1] 0

> ################################
> ######## Tercer paso #########
>
> #### Agregar C --> D
> gad.AB.AD.CD <- set.arc(gad.AB.AD, from = "C", to = "D")
> score.gad.AB.AD.CD <- score(gad.AB.AD.CD, type = "bic", data = learning.test)
> score.gad.AB.AD.CD - score.gad.AB.AD

[1] 823.7605

> #### Eliminar A --> B
> score.gad.AD - score.gad.AB.AD

[1] -1153.88

> #### Eliminar A --> D
> score.gad.AB - score.gad.AB.AD

[1] -1128.077

> #### Invertir A --> B
> gad.BA.AD <- set.arc(gad.AD, from = "B", to = "A")
> score.gad.BA.AD <- score(gad.BA.AD, type = "bic", data = learning.test)
```

```
> score.gad.BA.AD - score.gad.AB.AD

[1] 0

> #### Invertir A --> D
> gad.AB.DA <- set.arc(gad.AB, from = "D", to = "A")
> score.gad.AB.DA <- score(gad.AB.DA, type = "bic", data = learning.test)
> score.gad.AB.DA - score.gad.AB.AD

[1] 0
```



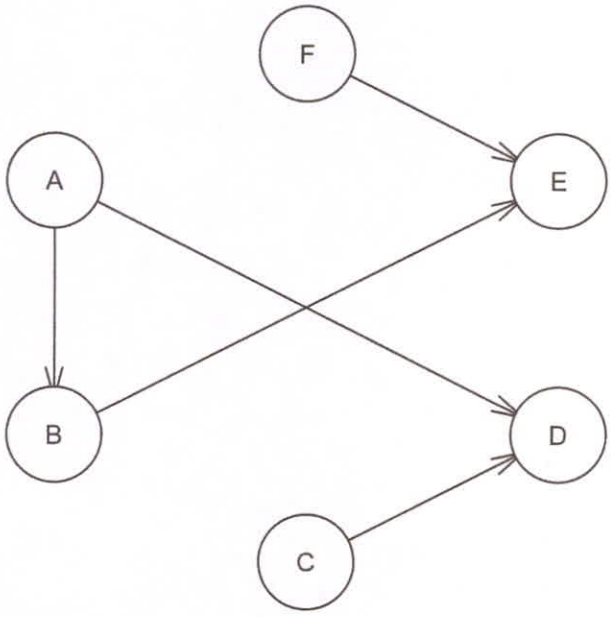
6.4. Ejemplo: Algoritmo Max-Min Padres e Hijos

```
> ################################
> ####### Algoritmo MMHC #######
```

```
>
> learning.test.mmhc <- mmhc(learning.test)
> plot(learning.test.mmhc)
> ###################################
> ########### Restricción ###########
>
> #### B es dependiente de A
> ci.test("B", "A", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  B ~ A
mi = 2341.8, df = 4, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0

> #### C es dependiente de A
> ci.test("C", "A", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  C ~ A
mi = 1.3091, df = 4, p-value = 0.8598
alternative hypothesis: true value is greater than 0

> #### D es dependiente de A
> ci.test("D", "A", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  D ~ A
mi = 2290.2, df = 4, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0

> #### E es dependiente de A
> ci.test("E", "A", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  E ~ A
mi = 466.11, df = 4, p-value < 2.2e-16
```

```
alternative hypothesis: true value is greater than 0

> #### F es dependiente de A
> ci.test("F", "A", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  F ~ A
mi = 1.3008, df = 2, p-value = 0.5218
alternative hypothesis: true value is greater than 0

> #### D es dependiente de A dado B
> ci.test("D", "A", "B", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  D ~ A | B
mi = 1809.7, df = 12, p-value < 2.2e-16
alternative hypothesis: true value is greater than 0

> #### E es dependiente de A dado B
> ci.test("E", "A", "B", test = "mi", data = learning.test)

        Mutual Information (disc.)

data:  E ~ A | B
mi = 7.5834, df = 12, p-value = 0.8168
alternative hypothesis: true value is greater than 0

> ####################################
> ########### Maximización ###########
>
> learning.test <- learning.test[, c(1,2,4)]
> ##############################
> ######## Primer paso ########
>
> gad.0 <- empty.graph(nodes = c("A", "B", "D"))
> score.gad.0 <- score(gad.0, type = "bic", data = learning.test)
> score.gad.0
```

```
[1] -15838.77

> #### Agregar A --> B
> gad.AB <- set.arc(gad.0, from = "A", to = "B")
> score.gad.AB <- score(gad.AB, type = "bic", data = learning.test)
> score.gad.AB - score.gad.0

[1] 1153.88

> #### Agregar A --> D
> gad.AD <- set.arc(gad.0, from = "A", to = "D")
> score.gad.AD <- score(gad.AD, type = "bic", data = learning.test)
> score.gad.AD - score.gad.0

[1] 1128.077

> ##############################
> ######## Segundo paso ########
>
> #### Agregar A --> D
> gad.AB.AD <- set.arc(gad.AB, from = "A", to = "D")
> score.gad.AB.AD <- score(gad.AB.AD, type = "bic", data = learning.test)
> score.gad.AB.AD - score.gad.AB

[1] 1128.077

> #### Eliminar A --> B
> score.gad.0 - score.gad.AB

[1] -1153.88

> #### Invertir A --> B
> gad.BA <- set.arc(gad.0, from = "B", to = "A")
> score.gad.BA <- score(gad.BA, type = "bic", data = learning.test)
> score.gad.BA - score.gad.AB

[1] 0
```


F
A
E
B
D
C

Bibliografía

[,]

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

[Bishop et al., 1995] Bishop, C., Bishop, C. M., et al. (1995). *Neural networks for pattern recognition*. Oxford university press.

[Bouckaert, 1995] Bouckaert, R. (1995). *Bayesian belief networks: from inference to construction*. PhD thesis, PhD thesis, Faculteit Wiskunde en Informatica, Utrecht University.

[Bouckaert, 2001] Bouckaert, R. R. (2001). *Bayesian belief networks: from construction to inference*. PhD thesis.

[Breiman, 1984] Breiman, L. (1984). Classification and regression trees.

[Breiman et al., 2005] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (2005). Classification and regression trees, wadsworth international group, belmont, california, usa, 1984; bp roe et al., boosted decision trees as an alternative to artificial neural networks for particle identification. *Nucl. Instrum. Meth. A*, 543:577.

[Castillo et al., 2012] Castillo, E., Gutierrez, J. M., and Hadi, A. S. (2012). *Expert systems and probabilistic network models*. Springer Science & Business Media.

[Cheng and Greiner, 1999] Cheng, J. and Greiner, R. (1999). Comparing bayesian network classifiers. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 101–108. Morgan Kaufmann Publishers Inc.

[Cheng et al., 2002] Cheng, J., Greiner, R., Kelly, J., Bell, D., and Liu, W. (2002). Learning bayesian networks from data: an information-theory based approach. *Artificial intelligence*, 137(1-2):43–90.

[Chow and Liu, 1968] Chow, C. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467.

[Dasarathy, 1991] Dasarathy, B. V. (1991). Nearest neighbor ({NN}) norms:{NN} pattern classification techniques.

[Denis and Scutari, 2014] Denis, J. and Scutari, M. (2014). Réseaux bayésiens avec r: Élaboration, manipulation et utilisation en modélisation appliquée.

[Dheeru and Karra Taniskidou, 2017] Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository.

[Domingos and Pazzani, 1997] Domingos, P. and Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2-3):103–130.

[Dougherty et al., 1995] Dougherty, J., Kohavi, R., and Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In *Machine Learning Proceedings 1995*, pages 194–202. Elsevier.

[Duda et al., 1973] Duda, R. O., Hart, P. E., Stork, D. G., et al. (1973). *Pattern classification*, volume 2. Wiley New York.

[Edwards, 2000] Edwards, D. (2000). Introduction to graphical modelling. springer. *2nd endition*.

[Fayyad and Irani, 1993] Fayyad, U. and Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning.

[Friedman et al., 1997] Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, 29(2-3):131–163.

[Friedman et al., 1999] Friedman, N., Nachman, I., and Peér, D. (1999). Learning bayesian network structure from massive datasets: the «sparse candidate «algorithm. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 206–215. Morgan Kaufmann Publishers Inc.

[Geiger and Heckerman, 1994] Geiger, D. and Heckerman, D. (1994). Learning gaussian networks. In *Uncertainty Proceedings 1994*, pages 235–243. Elsevier.

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

[Hall, 2007] Hall, M. (2007). A decision tree-based attribute weighting filter for naive bayes. *Knowledge-Based Systems*, 20(2):120–126.

[Hausser and Strimmer, 2009] Hausser, J. and Strimmer, K. (2009). Entropy inference and the james-stein estimator, with application to nonlinear gene association networks. *Journal of Machine Learning Research*, 10(Jul):1469–1484.

[Heckerman et al., 1995] Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243.

[Holte, 1993] Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90.

[Huang et al., 2014] Huang, G. T., Tsamardinos, I., Raghu, V., Kaminski, N., and Benos, P. V. (2014). T-recs: stable selection of dynamically formed groups of features with application to prediction of clinical outcomes. In *Pacific Symposium on Biocomputing Co-Chairs*, pages 431–442. World Scientific.

[Jiang et al., 2005] Jiang, L., Zhang, H., Cai, Z., and Su, J. (2005). Learning tree augmented naive bayes for ranking. In *International Conference on Database Systems for Advanced Applications*, pages 688–698. Springer.

[Kenett and Salini, 2011] Kenett, R. S. and Salini, S. (2011). *Modern analysis of customer surveys: with applications using R*, volume 117. John Wiley & Sons.

[Keogh and Pazzani, 1999] Keogh, E. J. and Pazzani, M. J. (1999). Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *AIStats*. Citeseer.

[Kerber, 1992] Kerber, R. (1992). Chimerge: Discretization of numeric attributes. In *Proceedings of the tenth national conference on Artificial intelligence*, pages 123–128. Aaai Press.

[Kim et al., 2003] Kim, S.-B., Seo, H.-C., and Rim, H.-C. (2003). Poisson naive bayes for text classification with feature weighting. In *Proceedings of the sixth international*

workshop on Information retrieval with Asian languages-Volume 11, pages 33–40. Association for Computational Linguistics.

[Koc et al., 2012] Koc, L., Mazzuchi, T. A., and Sarkani, S. (2012). A network intrusion detection system based on a hidden naïve bayes multiclass classifier. *Expert Systems with Applications*, 39(18):13492–13500.

[Kohavi et al., 1995] Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

[Koller and Friedman, 2009] Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

[Korb and Nicholson, 2004] Korb, K. and Nicholson, A. (2004). Bayesian artificial intelligence. chapman & hall/crc computer science and data analysis, boca raton, fl. *Bayesian artificial intelligence. Chapman & Hall/CRC Computer Science and Data Analysis, Boca Raton, FL.*

[Korb and Nicholson, 2010] Korb, K. B. and Nicholson, A. E. (2010). Bayesian artificial intelligence. *Florida: Chapman & Hall/CRC.*

[Koski and Noble, 2011] Koski, T. and Noble, J. (2011). *Bayesian networks: an introduction*, volume 924. John Wiley & Sons.

[Larranaga et al., 1997] Larranaga, P., Sierra, B., Gallego, M. J., Michelena, M. J., and Picaza, J. M. (1997). Learning bayesian networks by genetic algorithms: a case study in the prediction of survival in malignant skin melanoma. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 261–272. Springer.

[Lopez de Castilla, 2005] Lopez de Castilla, V. C. (2005). Clasificadores por redes bayesianas (spanish text).

[Lucas, 2004] Lucas, P. J. (2004). Restricted bayesian network structure learning. In *Advances in Bayesian Networks*, pages 217–234. Springer.

[Margaritis, 2003] Margaritis, D. (2003). Learning bayesian network model structure from data. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.

[Martınez et al., 2016] Martınez, A. M., Webb, G. I., Chen, S., and Zaidi, N. A. (2016). Scalable learning of bayesian network classifiers. *Journal of Machine Learning Research*, 17(44):1–35.

[Murphy, 2012] Murphy, K. P. (2012). Machine learning: A probabilistic perspective. adaptive computation and machine learning.

[Murthy et al., 1994] Murthy, S. K., Kasif, S., and Salzberg, S. (1994). A system for induction of oblique decision trees. *Journal of artificial intelligence research*, 2:1–32.

[Nagarajan et al., 2013] Nagarajan, R., Scutari, M., and Lèbre, S. (2013). Bayesian networks in r. *Springer*, 122:125–127.

[Neapolitan et al., 2004] Neapolitan, R. E. et al. (2004). *Learning bayesian networks*, volume 38. Pearson Prentice Hall Upper Saddle River, NJ.

[Pearl, 1988] Pearl, J. (1988). Probabilistic reasoning in intelligent systems: Networks of plausible inference. morgan kauffman pub.

[Pearl, 2009] Pearl, J. (2009). *Causality*. Cambridge university press.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

[Quinlan, 1993] Quinlan, J. R. (1993). C4. 5: Programming for machine learning. *Morgan Kauffmann*, 38:48.

[Russell et al., 2003] Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., and Edwards, D. D. (2003). *Artificial intelligence: a modern approach*, volume 2. Prentice hall Upper Saddle River.

[Sachs et al., 2005] Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529.

[Sahami, 1996] Sahami, M. (1996). Learning limited dependence bayesian classifiers. In *KDD*, volume 96, pages 335–338.

[Schäfer and Strimmer, 2005] Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4(1).

[Scutari and Brogini, 2012] Scutari, M. and Brogini, A. (2012). Bayesian network structure learning with permutation tests. *Communications in Statistics-Theory and Methods*, 41(16-17):3233–3243.

[Scutari and Strimmer, 2010] Scutari, M. and Strimmer, K. (2010). Introduction to graphical modelling. *arXiv preprint arXiv:1005.1036*.

[Sebe et al., 2002] Sebe, N., Lew, M. S., Cohen, I., Garg, A., and Huang, T. S. (2002). Emotion recognition using a cauchy naive bayes classifier. In *null*, page 10017. IEEE.

[Spirtes et al., 2000] Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, prediction, and search*. MIT press.

[Statnikov and Aliferis, 2010] Statnikov, A. and Aliferis, C. F. (2010). Analysis and computational dissection of molecular signature multiplicity. *PLoS computational biology*, 6(5):e1000790.

[Suzuki, 1999] Suzuki, J. (1999). Learning bayesian belief networks based on the minimum description length principle: basic properties. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 82(10):2237–2245.

[Taheri et al., 2014] Taheri, S., Yearwood, J., Mammadov, M., and Seifollahi, S. (2014). Attribute weighted naive bayes classifier using a local optimization. *Neural Computing and Applications*, 24(5):995–1002.

[Tsamardinos et al., 2003] Tsamardinos, I., Aliferis, C. F., Statnikov, A. R., and Statnikov, E. (2003). Algorithms for large scale markov blanket discovery. In *FLAIRS conference*, volume 2, pages 376–380.

[Tsamardinos and Borboudakis, 2010] Tsamardinos, I. and Borboudakis, G. (2010). Permutation testing improves bayesian network learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 322–337. Springer.

[Tsamardinos et al., 2006] Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78.

[Tsamardinos et al., 2012] Tsamardinos, I., Lagani, V., and Pappas, D. (2012). Discovering multiple, equivalent biomarker signatures. In *7th Conference of the Hellenic Society for Computational Biology and Bioinformatics (HSCBB12)*.

[Utschick et al., 1995] Utschick, W., Nachbar, P., Knobloch, C., Schuler, A., and Nossek, J. A. (1995). The evaluation of feature extraction criteria applied to neural network classifiers. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 315–318. IEEE.

[Verma and Pearl, 1991] Verma, T. S. and Pearl, J. (1991). Equivalence and synthesis of causal models. In *Uncertainty in artificial intelligence*, volume 6, page 255.

[Yaramakala and Margaritis, 2005] Yaramakala, S. and Margaritis, D. (2005). Speculative markov blanket discovery for optimal feature selection. In *Data mining, fifth IEEE international conference on*, pages 4–pp. IEEE.

[Zhang et al., 2005] Zhang, H., Jiang, L., and Su, J. (2005). Hidden naive bayes. In *AAAI*, pages 919–924.