C-MB: Utilización de métricas para obtener Fronteras de Markov de forma eficiente y efectiva

Silvia Acid, Luis M. de Campos, and Moisés Fernández

Departamento de Ciencias de la Computación e I.A., ETSIIT, Universidad de Granada, 18071, Granada, España
{acid,lci,moises}@decsai.ugr.es

Resumen En este artículo proponemos un algoritmo eficaz y eficiente, C-MB, capaz de aprender las fronteras de markov de una red bayesiana mediante el paradigma métrica+búsqueda. Sus resultados superan ampliamente a los algoritmos PCMB, IAMB y KIAMB basados en tests de independencia (TIs) que actualmente forman parte del estado del arte y demuestran ser competitivos. Demostraremos con una amplia batería de pruebas la gran potencialidad que tiene C-MB en comparación con los algoritmos basados en TIs.
Palabras clave: redes bayesianas, selección de características, frontera de Markov, tests de independencia, métricas.

1. Introducción

La selección de características, FSS (Feature subset selection), es un proceso muy utilizado en el campo del aprendizaje automático para reducir la dimensionalidad del problema. Este proceso de reducción puede ser crucial por varias razones. Primero, porque cuando se dispone de un gran número de características el proceso de computación resulta muy costoso para diferentes tareas de razonamiento: aprendizaje, clasificación, abducción, etc. Segundo, tener una elevada dimensionalidad es un gran inconveniente cuando el número de variables es grande y el número de datos disponibles es insuficiente para la estimación de parámetros. Tercero, muchas de estas características pueden ser redundantes o irrevelantes para la tarea de clasificación.

La clasificación probabilística es el proceso de asignar una etiqueta a una determinada variable (la clase C) a partir de los datos descritos por un conjunto de variables aleatorias, denominadas características. FSS nos ayudará a identificar el subconjunto mínimo de características que son necesarias para la clasificación probabilística y facilitar, de ese modo, el análisis de los datos. Por tanto, la selección de características elimina del dominio el resto de variables por ser irrelevantes o redundantes respecto a C. Actualmente pueden encontrarse una amplia variedad de algoritmos para FSS, a nosotros nos interesan los denominados filtros. El propósito de éstos es seleccionar aquellas características

que sean relevantes para la predicción de una variable de interés. Este proceso de selección es previo a la inducción del clasificador.

En este artículo nos centramos en las redes bayesianas como modelo gráfico probabilístico, y resolveremos el problema FSS mediante el aprendizaje de la Frontera de Markov (Markov Boundary). El MB de una variable aleatoria C está formado por el subconjunto minimal de las variables de un dominio capaz de hacer condicionalmente independiente C del resto de variables de la Red Bayesiana (RB).

Actualmente existen varios algoritmos considerados como parte del estado del arte (IAMB [9], KIAMB y PCMB [6]) que obtienen eficientemente el MB de un nodo C a partir de los datos. Todos ellos están basados en la aplicación de TIs sobre los datos como criterio para incluir o excluir atributos en el $MB(C)$. ¿Cuáles son sus deficiencias desde nuestra punto de vista? En primer lugar, sus resultados son menos eficaces que nuestro algoritmo C-MB basado en el paradigma métrica+búsqueda; para demostrarlo utilizaremos en la evaluación experimental (sección 5) tres medidas distintas: precisión, recall y distancia euclídea [6] para la recuperación de MBs de distintas RBs. En segundo lugar, dada la naturaleza inherente de los tests, éstos necesitan de una cantidad relativa de datos ya que los tests condicionales con datos insuficientes son poco fiables. La pregunta general que nos hacemos en este artículo es ¿podemos diseñar un algoritmo basado en un paradigma distinto al de los TIs para obtener el MB de una variable C de forma eficiente y efectiva? La respuesta es sí, describiremos ese algoritmo C-MB con paradigma métrica+búsqueda en la sección 4.

En la sección 2, estableceremos algunas definiciones básicas y revisaremos los algoritmos mencionados. En la sección 3 plantearemos la potencialidad de las métricas y comprobaremos lo ineficaces que son por sí solas para obtener un MB. Posteriormente en la sección 5 detallaremos el desarrollo experimental realizado; para finalizar con una serie de conclusiones en la sección 6.

2. Definiciones y Trabajos previos

2.1. Definiciones

Las siguientes definiciones pueden encontrarse en la mayoría de libros sobre Redes bayesianas, por ejemplo [5] [7] y [8]. Supongamos que tenemos un conjunto finito de variables aleatorias discretas $\boldsymbol{U}$= $\{X_1, X_2, ..., X_n\}$ y cada variable toma valores de un conjunto finito. Usaremos letras mayúsculas para indicar los nombres de las variables y letras minúsculas para los valores específicos que toman estas variables. Los conjuntos de variables se indican mediante letras mayúsculas en negrita y los valores de las variables en estos conjuntos mediante letras minúsculas en negrita.

Red Bayesiana: es una representación gráfica de una distribución de probabilidad conjunta [5]. Consta de dos componentes: el primero, un grafo dirigido acíclico (DAG) $G = (\boldsymbol{U}, E_G)$, donde $\boldsymbol{U}$ es el conjunto de nodos, representa las variables del dominio, y E_G, el conjunto de arcos, representa las relaciones de dependencia entre las variables; y el segundo, un conjunto de parámetros almacenados en cada variable de $\boldsymbol{U}$, normalmente distribuciones de probabilidad

condicional. Para cada variable $X \in \boldsymbol{U}$, disponemos de una familia de distribuciones condicionales $p(X \mid pa_G(X))$, una por cada configuración, $pa_G(X)$, del conjunto de padres de X en el grafo, $Pa_G(X) = \{Y \in \boldsymbol{U} \mid Y \rightarrow X \in E_G\}$. A partir de esas distribuciones condicionales podemos recuperar la distribución conjunta sobre $\boldsymbol{U}$ de la siguiente forma :

$$p(x_1, x_2, ..., x_n) = \prod_{X_i \in \mathbf{U}} p(x_i \mid pa_G(X_i))$$

Independencia condicional: Dos variables X e Y son condicionalmente independientes dado $\mathbf{Z}$, cuando $P(X|Y, \mathbf{Z}) = P(X|\mathbf{Z})$, $\forall x, \forall y, \forall z$, se suele notar de la siguiente forma: $X \perp Y|\mathbf{Z}$.
Frontera de Markov: Dados los atributos X_i de un dominio $\boldsymbol{U}$, el Manto de Markov (Markov Blanket) de un atributo $X \in \boldsymbol{U}$, $MM(X)$ es cualquier subconjunto de $\boldsymbol{U}\backslash\{X\}$ donde $\forall X_i \in \boldsymbol{U}\backslash\{X\}\backslash MM(X)$ tenemos que $X \perp X_i|MM(X)$. El $MM(X)$ tiene el efecto de hacer independiente X del resto de atributos. Un MM minimal recibe el nombre de Frontera de Markov (Markov Boundary). En este artículo proponemos un algoritmo que aprende Fronteras de Markov (MB). Dada una Red Bayesiana, el $MB(X)$ es fácilmente identificable ya que está formado por los padres, hijos y padres de los hijos del nodo objetivo X[5].

2.2. Trabajos previos

Al igual que el aprendizaje de RBs, el aprendizaje de MBs puede enfocarse según el paradigma *métrica+búsqueda* o *basado en restricciones*. En primer lugar nos ocuparemos del segundo grupo porque en él se encuentran los algoritmos que queremos estudiar. Éstos se caracterizan por realizar TIs χ^2 con el estadístico G^2 para decidir si existe independencia condicional entre X e Y, $Y \notin MB(X)$, es decir, $I(X, Y|MB(X))$. Por tanto, los TIs permiten descartar aquellas variables que no forman parte del $MB(X)$. Para medir la fuerza de dependencia entre X e Y, $dep(X, Y|MB(X))$, se usa un p-valor negativo procedente del estadístico G^2, el cual es obtenido a partir de los datos D.

Como trabajos previos podemos mencionar los algoritmos IAMB, PCMB y KIAMB descritos en [6], éstos son bien conocidos por su alto grado de exactitud, escalabilidad y eficacia sobre bases de datos grandes. Los tres métodos utilizan dos fases: (1) la primera consiste en aumentar el conjunto de características $MB(X)$ (conjunto inicialmente vacío) con aquellas variables que muestran una gran dependencia con la variable objetivo C, y por tanto, son inseparables de C según los TIs. Esta fase puede incluir algunos falsos positivos, por esa razón, en (2) la segunda fase se reduce el conjunto de características $MB(X)$ eliminando todas aquellas variables que son identificadas como falsos positivos.

IAMB trata de seleccionar en su primera fase aquellas variables que muestran una mayor dependencia con la variable objetivo y los incluye en el conjunto $MB(X)$; en la segunda fase elimina de ese conjunto los falsos positivos. Por otro lado, PCMB divide el problema de encontrar el $MB(X)$ en dos subproblemas: primero, identificar el conjunto de Padres e Hijos del nodo objetivo ($PC(X)$) y , segundo, identificar el resto de componentes del $MB(X)$, es decir, los padres de

los hijos de X. KIAMB es una versión estocástica con un parámetro $k \in [0, 1]$, donde k da un grado de aleatoriedad al tamaño del subconjunto condicionante a utilizar en los tests. Esto introduce una cierta aleatoriedad en la selección de las características que puede evitar introducir algunos falsos positivos. Los autores recomiendan un valor de 0.8.

Todos estos algoritmos asumen que los TIs son correctos, y por tanto, deciden la (in)dependencia a partir de los datos D. Para maximizar la eficacia los algoritmos realizan un test de independencia siempre y cuando éste sea fiable y en caso contrario no se realiza. Siguiendo lo comentado en [7], un test es fiable cuando el número de instancias en D es al menos cinco veces el número de grados de libertad en el test. Esto significa que el número de instancias requerido para seleccionar las características para el $MB(X)$, es al menos exponencial en el tamaño de $MB(X)$[1]. La eficacia de los tests estadísticos depende de la cantidad de datos disponibles, cuando los datos son insuficientes o bien no se efectúan los tests o bien éstos pueden fallar y se asumen relaciones de (in)dependencias incorrectas entre pares de variables. Para mostrar las limitaciones que pueden presentar estos algoritmos hemos desarrollado una serie de pruebas con BDs de tamaño reducido.

3. Métricas en un espacio de búsqueda local

Una vez que la RB ha sido aprendida mediante cualquier algoritmo a partir de unos datos D, se puede obtener el $MB(X)$ de manera muy simple desde la propia estructura de la red, seleccionando sus padres, hijos y padres de sus hijos. Las métricas han demostrado ser muy útiles para aprender las RBs a partir de datos. Métricas estándares como BDeu [3], MDL [4], MIT [2] entre otras, han sido profusamente utilizadas y han dado lugar a muchos algoritmos de aprendizaje de RBs. Nosotros argumentamos que las métricas son útiles para aprender MBs evaluados en su conjunto, en contraposición a establecer un ranking de cada una de las variables evaluadas individualmente. Las métricas mencionadas anteriormente se combinan con un *método de búsqueda* para atravesar el espacio de las posibles configuraciones. El propósito de la métrica es evaluar cada configuración candidata, y así poder establecer un orden de preferencia de las configuraciones para poder seleccionar la mejor.

Para comprobar la tesis de que las métricas son adecuadas para el aprendizaje de MB hemos elaborado un algoritmo tipo HillClimbing destinado a construir una red local a un nodo C para obtener así su MB, a este algoritmo lo hemos llamado CHillClimber (centrado en el nodo clase, es decir, local[2]). Los enlaces que el algoritmo CHillClimber va a considerar son los que pueden cambiar la composición del MB. Este algoritmo parte de un grafo inconexo y utiliza dos operaciones básicas *añadir* y *borrar* arco, cuyas restricciones son:

[1] El número de grados de libertad en un test es exponencial en el tamaño del conjunto condicionante

[2] No se considera el caso de HillClimber global porque aprender una RB completa para obtener un MB es poco operativo y en casos de grandes BDs inabordable.

- Añadir arco: que o bien conecte C con cualquier variable X (en cualquier dirección) o bien un arco de X a un hijo de C.
- Eliminar arco: se puede eliminar cualquier arco excepto aquél que va de C a X cuando éste (X) tenga más de un padre.

Se evalúa cada configuración obtenida de la aplicación sucesiva de las operaciones anteriores mediante algunas de las métricas seleccionadas, hasta que no haya mejora. Finalmente componemos el MB con los padres, hijos y padres de los hijos de C.

4. C-MB y un método de búsqueda específico

Como acabamos de apuntar, se pueden explotar las métricas si configuramos de forma específica el espacio de búsqueda. Nos basaremos en estructuras del tipo CRPDAG (Grafo parcialmente dirigido, restringido y centrado en la variable C) [1]. El valor de estas configuraciones es que representan de forma esencial patrones de independencia equivalentes a efectos de la clasificación de C y es determinante para la composición del MB.

El método utiliza una simple búsqueda local de tipo HillClimbing pero restringido a 12 operadores elementales. De forma abreviada los operadores son (en referencia a C): *AñadirPadre*, *AñadirHijo*, *AñadirHermano*, *AñadirCabeza-Cabeza*, *AñadirPadreDeHijo*, *AñadirCabezaCabezaDeHijo*, *EliminarPadre*, *EliminarHijo*, *EliminarHermano*, *EliminarCabezaCabeza*, *EliminarPadreDeHijo* y *EliminarCabezaCabezaDeHijo*. De forma intuitiva, los operadores *Añadir* pueden hacer crecer el conjunto MB con nuevas características mientras los de *Eliminar* pueden reducirlo; todos ellos se pueden aplicar en cualquier orden, guiados por el valor de la métrica. Estos son los operadores originalmente propuestos en [1] a los que se han incorporado los operadores *AñadirHijoDeHijo* y *EliminarHijoDeHijo* para evitar problemas de sobreajuste a la variable C. Esto se debe a que el algoritmo tiende a conectar muchas variables a C aunque exista una configuración mejor si se conecta a un nodo hijo. Por esta razón se amplía el espacio de búsqueda dando lugar a un nuevo tipo de configuración, el C-MB (CRPDAG-MB).

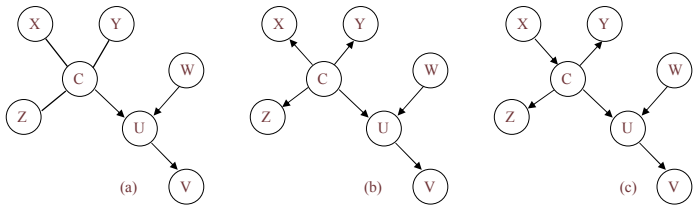


Figura 1. (a) Ejemplo de grafo C-MB, (b) y (c) DAGs equivalentes de (a)

Partiendo del CRPDAG inconexo se van explorando las configuraciones vecinas a partir de los operadores mencionados que van refinando la estructura hasta devolver la configuración C-MB que más se ajuste a los datos según la métrica

utilizada. En la figura 1 puede verse un ejemplo de la estructura C-MB y varios DAGs equivalentes al mismo. Finalmente, el MB se compone de los nodos conectados a la variable C mediante un enlace (dirigido o no dirigido) y los padres de los hijos de C.

5. Evaluación experimental

Para estudiar la eficiencia y efectividad real de nuestro algoritmo en comparación con CHillClimber, PCMB, KIAMB e IAMB hemos diseñado varias líneas de experimentación sobre una amplia variedad de bases de datos con diferentes tamaños. Los algoritmos C-MB y CHillClimber han sido implementados en Weka mediante el lenguaje de programación JAVA, en cambio, para PCMB, KIAMB e IAMB hemos utilizado la implementación realizada en C++ por Peña et al. [6]. Los parámetros a priori necesarios para las distintas ejecuciones son $\alpha = 0{,}01$ para el nivel de confianza de los TIs; $k = 0{,}8$ para el algoritmo KIAMB, $\alpha = 0{,}9999$ para la métrica MIT y distribución a priori uniforme con tamaño muestral equivalente 1 para la métrica BDeu.

5.1. Los Datos

Nuestra experimentación se inicia con la adquisición de 6 redes (Alarm, Barley, Boblo, Hailfinder, Insurance, Pigs) pertenecientes en su mayoría al repositorio de RBs (`http://www.cs.huji.ac.il/labs/compbio`), todas ellas proceden de diferentes campos: dianóstico médico, riesgo de seguros, meteorología y agricultura. La tabla 1 da algunos detalles sobre las RB utilizadas para la generación de bases de datos. Por cada RB hemos generado 5 bases de datos y los resultados obtenidos se han recogido con media aritmética de las 5. Por otro lado, todos los experimentos han sido realizados con diferentes tamaños: 100, 200, 500, 1000 y 10000, aunque por razones de espacio sólo presentaremos algunos resultados.

Redes Bayesianas	N. variables	N. enlaces	N. casos (Media)
Alarm	37	46	2-4 (2.8)
Insurance	27	52	2-5 (3.3)
Barley	48	84	2-67 (8.8)
Hailfinder	56	66	2-11 (3.3)
Boblo	23	24	2-6 (2.7)
Pigs	441	592	3 (3.0)

Tabla 1. Detalles sobre las redes bayesianas utilizadas

Durante este desarrollo experimental hemos ejecutado los algoritmos CHillClimber, PCMB, KIAMB, IAMB y C-MB (con diferentes métricas) para determinar el MB para cada variable del dominio de cada BD. Por tanto, los resultados obtenidos se evalúan mediante criterios conocidos como precisión, recall y distancia euclídea en promedio para cada uno de los nodos de las RBs. La *precisión* es el número de verdaderos positivos devueltos por el algoritmo (salida) dividido por el número de nodos en la salida. La *recall* es el número de verdaderos positivos

en la salida dividido por el número de verdaderos positivos en la RB. La *distancia euclídea* se define de la siguiente forma $\sqrt{(1-precision)^2+(1-recall)^2}$. Se aplica una penalización de -1.0 en la precisión y recall en caso de que el MB devuelto por el algoritmo esté vacío, esto es frecuente en BDs muy pequeñas como puede apreciarse en la tabla 3.

5.2. C-MB frente a CHillClimber

En primer lugar vamos a demostrar que las métricas pueden ser útiles para obtener los MB dependiendo del espacio de búsqueda empleado. Para ello vamos a comparar los algoritmos C-MB y CHillClimber sobre las mismas BDs empleando para ambos la métrica BDeu. En la tabla 2 se muestran los valores de precisión y recall obtenidos en promedio sobre los nodos de un dominio para cada una de las RBs. Un valor alto de precisión (cercano a 1.0) indica que los nodos seleccionados por el algoritmo son correctos; mientras que valores muy bajos (cercano a 0.0), como es el caso de CHillClimber, apuntan a que éste introduce muchos falsos positivos, esto es, es poco selectivo. Por otro lado, un valor de recall alto (cercano a 1.0) significa que se han seleccionado una gran proporción de los atributos que forman parte del MB real. Si observamos los valores de la tabla 2 el algoritmo CHillClimber obtiene mejores resultados de recall aunque los valores de C-MB son muy próximos a él. Sin embargo, para tener una idea de la efectividad de un algoritmo es necesario considerar ambas medidas conjuntamente, esto puede verse mediante la distancia euclídea.

	Alarm	Barley	Boblo	Hailfinder	Insurance
C-MB (Precisión)	**0,75**	**0,73**	**0,60**	**0,85**	**0,83**
CHillClimber (Precisión)	0,22	0,29	0,19	0,27	0,24
C-MB (Recall)	0,92	0,71	**0,79**	0,84	0,83
CHillClimber (Recall)	**0,95**	**0,80**	0,76	**0,92**	**0,95**

Tabla 2. Precisión y Recall para bases de datos de tamaño 10000

En la figura 2 se ilustra gráficamente los valores para la distancia euclídea obtenida con los valores de la tabla 2. En esta situación, valores bajos para la distancia (cercanos a 0.0) indican mayor eficacia del algoritmo, esto es, los MBs que obtiene son correctos. De los valores mostrados en esta figura podemos concluir que el CHillClimber introduce de forma indiscriminada variables en los MBs, de ahí que su distancia euclídea sea alta. C-MB introduce con mayor frecuencia verdaderos positivos y en menor medida falsos positivos, esto supone obtener una distancia muy baja. Con ello, demostramos que las métricas por sí solas no obtienen buenos resultados. Los valores que presentamos son los obtenidos para BDs de tamaño 10000; con otros tamaños los resultados se confirman.

5.3. C-MB frente a los algoritmos basados en independencias

En segundo lugar vamos a comparar C-MB y diferentes métricas (BDeu, MDL y MIT) con los algoritmos basados en independencias. En las tablas 3 y 4 se

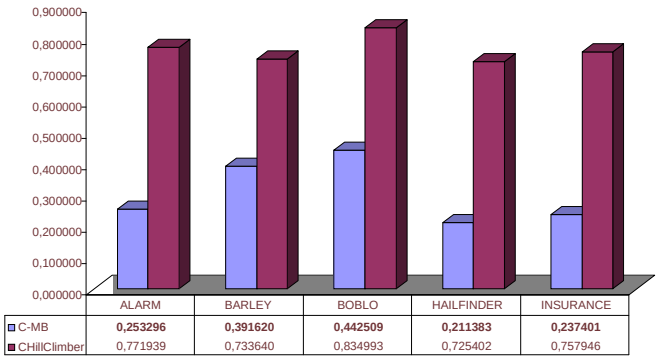


Figura 2. Distancia Euclídea para bases de datos de tamaño 10000

muestran respectivamente los valores de precisión y recall para BDs con tamaño 100. C-MB obtiene unos valores de precisión un poco inferiores a los algoritmos del estado del arte aunque sus valores de recall son considerablemente más altos; de ahí que en la figura 3 se ilustren unos resultados significativamente mejores en C-MB. Como podemos observar en la tabla 3 los TIs con datos insuficientes fallan en ocasiones dejando vacíos los MBs.

	Alarm	Barley	Boblo	Hailfinder	Insurance	Pigs
PCMB	0,24	−0,36	0,20	0,47	0,43	0,81
KIAMB(k=0.75)	0,51	−0,36	0,34	0,53	0,62	0,73
IAMB	**0,55**	−0,35	**0,41**	**0,58**	**0,70**	0,81
C-MB-BDeu	0,39	0,01	0,26	0,24	0,53	0,84
C-MB-MDL	0,44	−0,73	0,27	0,26	0,49	0,72
C-MB-MIT	0,40	**0,26**	0,22	0,55	0,52	**0,86**

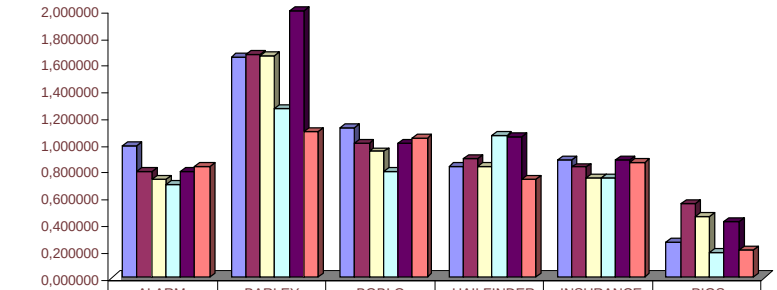
Tabla 3. Precisión para bases de datos de tamaño 100

	Alarm	Barley	Boblo	Hailfinder	Insurance	Pigs
PCMB	0,36	0,07	0,21	0,35	0,31	0,81
KIAMB (k=0.8)	0,37	0,03	0,23	0,24	0,26	0,51
IAMB	0,41	0,03	0,26	0,27	0,31	0,58
C-MB-BDeu	**0,64**	**0,20**	**0,70**	0,25	**0,41**	**0,89**
C-MB-MDL	0,42	0,01	0,30	0,24	0,28	0,68
C-MB-MIT	0,41	0,19	0,30	**0,41**	0,28	0,84

Tabla 4. Recall para bases de datos de tamaño 100

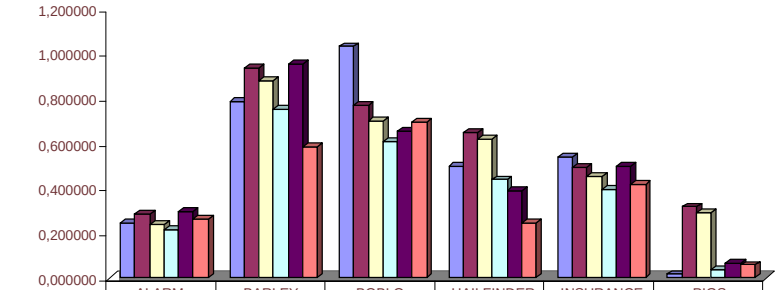
En la tabla 4 se puede ver que el algoritmo C-MB con métricas BDeu y MIT supera en general en recall a los algoritmos basados en independencias. Cabe destacar que la métrica MDL es la que obtiene los peores resultados dentro de las ejecuciones de C-MB debido a que ésta penaliza estructuras complejas en el proceso de exploración. Por ello, podemos concluir que las métricas pueden influir en los resultados obtenidos dentro de nuestro espacio reducido C-MB. Según los valores de la distancia euclídea, ver la figura 4, podemos concluir que

el algoritmo C-MB es más eficaz que los basados en independencias, concretamente C-MB obtiene menores distancias de forma significativa en muchos casos. La excepción es la BD Pigs (tamaño grande y poco densa) con la que algunos algoritmos basados en independencias obtienen mejores resultados.



	ALARM	BARLEY	BOBLO	HAILFINDER	INSURANCE	PIGS
PCMB	0,986228	1,651910	1,120870	0,831672	0,882641	0,262115
KIAMB k=0.8	0,792286	1,670126	1,003172	0,890036	0,826981	0,548617
IAMB	0,737362	1,658926	0,942137	0,835689	0,748029	0,455772
C-MB BDeu	**0,697205**	1,264690	**0,796554**	1,059237	**0,743723**	**0,185307**
C-MB MDL	0,795173	1,994902	1,001988	1,055273	0,876759	0,416565
C-MB MIT	0,835523	**1,091230**	1,042568	**0,736129**	0,859924	0,203431

Figura 3. Distancia euclídea para bases de datos de tamaño 100



	ALARM	BARLEY	BOBLO	HAILFINDER	INSURANCE	PIGS
PCMB	0,243157	0,790396	1,033284	0,498530	0,538931	**0,016304**
KIAMB k=0.8	0,282652	0,935749	0,769192	0,647863	0,493571	0,317029
IAMB	0,239447	0,881011	0,699482	0,622665	0,454362	0,288726
C-MB BDeu	**0,213421**	0,750572	**0,606165**	0,438474	**0,393052**	0,036354
C-MB MDL	0,295480	0,952720	0,657133	0,391302	0,500113	0,065125
C-MB MIT	0,260735	**0,583725**	0,695695	**0,243048**	0,418792	0,057007

Figura 4. Distancia euclídea para bases de datos de tamaño 1000

Las conclusiones extraídas se siguen manteniendo aunque de forma más moderada para BDs de mayor tamaño. Un ejemplo de ello puede verse en la figura 4 para BDs de tamaño 1000. Para observar el comportamiento de los algoritmos con BDs grandes hemos elegido tamaño 1000 en vez de 10000 debido a que PCMB y C-MB actualmente no son eficientes ni escalables para la BD Pigs.

6. Conclusiones y trabajos futuros

En este artículo hemos propuesto una alternativa diferente a los algoritmos basados en TIs para el aprendizaje de la frontera de Markov de un nodo C. Esta alternativa procedente del paradigma métrica + búsqueda puede utilizar diversas métricas estándares en el aprendizaje de RBs. Los resultados obtenidos mejoran a los del estado del arte con los que nos hemos comparado. Hemos demostrado que la eficacia de nuestro algoritmo no se debe sólo a las métricas sino principalmente al espacio de búsqueda utilizado (CRPDAG).

Una ventaja de utilizar nuestro algoritmo como FSS es que además de dar el MB, nos proporciona la topología del subgrafo parcialmente orientado inducido por el MB. Este subgrafo puede ser fácilmente completado para la obtención de una RB que actúe como clasificador. Los tiempos de ejecución de nuestro algoritmo para bases de datos grandes, por ejemplo Pigs de tamaño 10000, son ligeramente inferiores a los obtenidos por PCMB aunque considerablemente superiores a IAMB. Por este motivo actualmente estamos trabajando en el diseño de un "Fast C-MB" que reduzca considerablemente el número de configuraciones exploradas. Queda pendiente la evaluación de la efectividad del clasificador potencial obtenido mediante el algoritmo C-MB, teniendo en cuenta que los resultados sobre el MB son prometedores.

Agradecimientos

Este trabajo ha sido financiado por el programa de investigación española Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

Referencias

1. S. Acid, L.M. de Campos, and J.G. Castellano. Learning Bayesian network classifiers: Searching in a space of partially directed acyclic graphs. *Mach. Learn.*, 59:213–235, 2005.
2. L.M. de Campos. A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *J. Mach. Learn. Res.*, 7:2149–2187, 2006.
3. D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.*, 20:197–243, 1995.
4. W. Lam and F. Bacchus. Learning Bayesian belief networks: An approach based on the MDL principle. *Comput. Intell.*, 10:269–293, 1994.
5. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, 1988.
6. J.M. Peña, R. Nilsson, J. Björkegren, and J. Tegner. Towards scalable and data efficient learning of Markov boundaries. *Int. J. Approx. Reason.*, 45(2):211–232, 2007.
7. P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. Lecture Notes in Statistics 81. Springer Verlag, New York, 1993.
8. M. Studeny. *Probabilistic Conditional Independence Structures*. Springer-Verlag, 2005.
9. I. Tsamardinos, C.F. Aliferis, and A. Statnikov. Algorithms for large scale Markov blanket discovery. In *Proceedings of 16th International FLAIRS Conference*, pages 376–381, 2003.