Variante de Grow Shrink para mejorar la calidad de Markov blankets

Facundo Bromberg, Federico Schlüter

Dept. Sistemas de Información, Facultad Regional Mendoza,
Universidad Tecnológica Nacional, Mendoza, Argentina

Abstract. This work introduces *Grow-Shrink with Search* (**GSS**), a novel adaptation of the *Grow-Shrink* (**GS**) algorithm that learns a set of direct dependences of a random variable; called the *Markov Blanket* (**MB**) of the variable. We focus on the use of MBs for learning undirected probabilistic graphical models (a.k.a. Markov networks). As in the GS algorithm, GSS learns the MB by executing a series of statistical tests of conditional independence. The reliability of these tests decreases with the amount of data. While GS ignores this fact deciding on potentially incorrect MBs, GSS decides through a novel quality measure also introduced in this work, based on the posterior probability of a MB given the data. GSS proceeds as an optimization search over all possible independence assignments of the tests performed, searching for the assignment that maximizes this quality measure. This is in direct contrast to GS that performs a greedy optimization based on local decisions, i.e., the independence tests. Experimental results shows improvements up to 10% of the Hamming distance (normalized) of the learned MB vs. the real MB.

1 Introducción

El conocimiento está pasando a ser la pieza de mayor valor en nuestra sociedad. Junto con este cambio, es cada vez más ubicua la adquisición de datos en formato digital. Para aprovechar estos datos, buscamos mecanismos para generar conocimiento de modo automatizado a partir de ellos. En particular buscamos mecanismos de aprendizaje de bases de conocimiento probabilísticas, representadas a través de distribuciones de probabilidad sobre las proposiciones del sistema a modelar (modeladas a su vez por variables aleatorias X, Y, Z). Esta representación, además de generalizar la lógica proposicional, es susceptible a mejoras substanciales tanto en complejidad espacial como temporal, gracias a la explotación de las independencias condicionales entre las variables aleatorias del sistema. Los *modelos probabilísticos gráficos* [14, 10] formalizan esta explotación representando la distribución de probabilidad a través de una estructura de independencias entre las variables aleatorias del dominio, junto a un conjunto de parámetros numéricos. La estructura consiste en un grafo con un nodo por variable aleatoria y aristas (dirigidas o no dirigidas) que codifican las independencias entre las variables. Los parámetros numéricos consisten en tablas de valores

reales que junto con el grafo, resultan en un modelo probabilístico debidamente cuantificado. Nuestra investigación se concentra entonces en el aprendizaje de estructuras, dejando el aprendizaje de parámetros de lado.

Existen actualmente varios algoritmos que resuelven el problema de aprendizaje de estructuras, clasificados en dos enfoques distintos: algoritmos *basados en puntaje* [9, 8], y algoritmos *basados en independencias* [17]. Respecto de los primeros, la resolución del problema consiste en la búsqueda del mejor modelo, dados los datos, de acuerdo a un mecanismo de puntuación (e.g., *likelihood* de los datos dado el modelo, *minimum description length* [9], o *pseudo-likelihood* [4]). El cómputo de estos puntajes requiere la estimación de los parámetros numéricos. En contraste, los algoritmos basados en independencias aprenden la estructura directamente. Estos proceden mediante la ejecución de una secuencia de tests estadísticos sobre distintos pares de variables (e.g., X e Y), que estiman, a partir de los datos de entrada D, la independencia o no entre estas variables, condicionadas en un dado conjunto (*condicionante*) $\mathbf{Z}$. Denotamos esta independencia (dependencia) por $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$ $((X \not\perp\!\!\!\perp Y \mid \mathbf{Z}))$, y con un sub-indice D cuando estas afirmaciones son resultado de un test estadístico, e.g. $(X \perp\!\!\!\perp Y \mid \mathbf{Z})_D$ $((X \not\perp\!\!\!\perp Y \mid \mathbf{Z})_D)$. Los algoritmos basados en independencias presentan importantes ventajas sobre los basados en puntaje: (i) Ventajas computacionales por no requerir ni búsqueda, ni el cómputo intercalado de los parámetros durante la misma (exacerbado por ser el cómputo de estos parámetros NP-completo para modelos con grafos no-dirigidos [3]), y (ii) formales, por ser posible demostrar que las estructuras obtenidas son correctas (bajo condiciones, c.f. Section 2). En este artículo exponemos un nuevo enfoque para mejorar la calidad de algoritmos basados en independencias, atacando la principal crítica del enfoque: baja calidad de las estructuras que estos algoritmos producen cuando es alta la densidad de dependencias en la distribución. Esta baja calidad se debe principalemente a que los tests estadísticos de independencia, para garantizar la calidad, tienen un requerimiento de datos exponencial en el número de variables involucradas en el test [1]. Modelos densos en dependencias requieren muchos condicionantes en los tests de independencia, elevando el número de variables involucradas. La mejora de la calidad tiene relevancia en una gran diversidad de ramas de la ciencia que utilizan esta clase de modelos para resolver sus problemas. Citamos como ejemplos: visión computacional [5, 2] para restauración de imágenes ruidosas, clasificación de texturas y segmentación de imágenes; para investigación genética y diagnóstico de enfermedades [7]; y en todos los campos de investigación que utilizan el mineo de datos espacial para generar nuevo conocimiento, como ser geografía, transporte, agricultura, climatología, ecología [16]; y muchos otros.

2 Aprendizaje de estructuras basado en independencias.

El enfoque de aprendizaje de estructuras basado en independencias consiste en la evaluación iterada de tests de independencia condicional entre distintos tripletes $(X, Y \mid \mathbf{Z})$ ($\{X\}, \{Y\}, \mathbf{Z}$ subconjuntos disjuntos de $\mathbf{V}$, y $\mathbf{V}$ el conjunto dado por todas las variables aleatorias de un sistema). Cada nueva independencia puede

ser inconsistente con una o más estructuras, las cuales son eliminadas como candidatas. El algoritmo prosigue hasta que queda una sola estructura consistente con los tests realizados. La existencia de una única estructura consistente puede demostrarse bajo suposiciones [14, 17]). Cuando se cumple se dice que el algoritmo es *sólido*.

Una manera muy utilizada para el aprendizaje sólido de estructuras es el aprendizaje, sólido y basado también en independencias, de los Markov blankets de las variables en $\mathbf{V}$. El *Markov blanket* $\mathbf{B}^X$ de una variable X, consiste en el conjunto de aquellas variables que "escudan" a X de las variables restantes, i.e., $\forall Y \notin \{X\} \cup \mathbf{B}^X, (X \perp\!\!\!\perp Y \mid \mathbf{B}^X)$. Para redes Markovianas, se ha demostrado [14] que la estructura correcta (i.e., aquella que satisface las independencias y dependencias del modelo subyacente), puede construirse agregando una arista (X, Y) para cada variable $X \in \mathbf{V}$ y cada variable $Y \in \mathbf{B}^X$. Para redes Bayesianas, es necesario además dirigir las aristas. Este enfoque ha sido ejemplificado para redes Bayesianas por los algoritmos GSBN y IAMB (y variantes) de [13, 18], respectivamente; y para redes Markovianas por los algoritmos GSMN y GSIMN de [6].

Todos estos algoritmos utilizan el algoritmo **GS** (**G**row-**S**hrink) [13] para el aprendizaje del Markov blanket. Dados como entrada $X \in \mathbf{V}$ y el conjunto de datos D, retorna el blanket $\mathbf{B}^X$ de X. El algoritmo mantiene un conjunto $\mathbf{S}$, inicialmente vacío, al cual le agrega variables durante la etapa *grow*, y le quita variables durante la etapa de *shrink*, de acuerdo al resultado de distintos tests estadísticos. A continuación, el algoritmo GS:

1: $\mathbf{B}^X \leftarrow \varnothing$
2: **for each** $Y \in \mathbf{V} - \{X\}$, **if** $(X \not\perp\!\!\!\perp Y \mid \mathbf{B}^X)_D$ **then** $\mathbf{B}^X \leftarrow \mathbf{B}^X \cup \{Y\}$ //**Grow**
3: **for each** Y in $\mathbf{B}^X$, **if** $(X \perp\!\!\!\perp Y \mid \mathbf{B}^X - \{Y\})_D$ **then** $\mathbf{B}^X \leftarrow \mathbf{B}^X - \{Y\}$ //**Shrink**
4: **return** $\mathbf{B}^X$

En teoría (y para la demostración de solidez), se asume la existencia de un oráculo que provee información precisa respecto de las independencias condicionales de las variables del dominio. En la práctica no existe un oráculo tal, pero puede aproximarse mediante tests estadísticos de independencia sobre el conjunto de datos D. Por ejemplo, para datos discretos puede usarse el test de independencia condicional χ^2 de Pearson [1], o el *test Bayesiano* de Margaritis [11, 12]. Para datos Gaussianos continuos, un test estadístico que puede utilizarse para medir la independencia condicional es la correlación parcial [17]. En este trabajo usaremos el test Bayesiano.

Para determinar independencia condicional entre dos variables X e Y dado un conjunto Z a partir de los datos, el test Bayesiano compara la probabilidad a posteriori del modelo de independencia con el umbral 0.5, i.e., $(X \not\perp\!\!\!\perp Y \mid \mathbf{Z}) \iff \mathtt{Pr}((X \not\perp\!\!\!\perp Y \mid \mathbf{Z}) \mid D) \geq 0.5$. Esta probabilidad se computa en base a las verosimilitudes del modelo que asume independencia y del que asume dependencia, i.e., $\mathtt{Pr}(D \mid (X \perp\!\!\!\perp Y \mid \mathbf{Z}))$ y $\mathtt{Pr}(D \mid (X \not\perp\!\!\!\perp Y \mid \mathbf{Z}))$ respectivamente:

$$\mathtt{Pr}((X \not\perp\!\!\!\perp Y \mid \mathbf{Z}) \mid D) = \frac{\mathtt{Pr}(D \mid (X \not\perp\!\!\!\perp Y \mid \mathbf{Z}))}{\mathtt{Pr}(D \mid (X \not\perp\!\!\!\perp Y \mid \mathbf{Z})) + \mathtt{Pr}(D \mid (X \perp\!\!\!\perp Y \mid \mathbf{Z}))}.$$

Un *test erróneo* es entonces aquel cuya decisión de independencia es errada, puede entenderse como aquel test que arroja una probabilidad a posteriori tan errada que "cruza" el umbral. De Eq. (2) se desprende que la decisión de independencia es entonces equivalente a comparar $\texttt{Pr}(D \mid (X \not\perp\!\!\!\perp Y \mid \mathbf{Z})) \geq \texttt{Pr}(D \mid (X \perp\!\!\!\perp Y \mid \mathbf{Z}))$. Así, un test errado es aquel que arroja valores de verosimilitud tan errados que esta desigualdad se invierte.

3 Mejora de la calidad

Para mejorar la calidad de los algoritmos basados en independencias se distinguen dos alternativas: (i) mejorar la calidad de los tests, o, (ii) dado el test, diseñar algoritmos que mejoren la calidad de las estructuras. Un ejemplo de la primer alternativa se expone en [6]. Ésta consiste en la corrección de los valores de independencia arrojados por los tests por medio de la explotación de ciertas restricciones que estos deben cumplir: los *axiomas de independencia* [14]. Como estos derivan directamente de los axiomas de probabilidad, sólo independencias erróneas violarían estas restricciones. En [6], se propone con éxito un formalismo para decidir eficientemente si un test es errado en base al conjunto de restricciones satisfechas y violadas por este test. Como alternativa, podemos mejorar la calidad teniendo en cuenta que el resultado arrojado por los tests de independencia no es del todo confiable. Esta propuesta, introducida por el algoritmo PFMN [12], consiste en mantener una distribución de probabilidad a posteriori $\texttt{Pr}(G \mid D)$ de una estructura G dado el conjunto de datos D (c.f. Section 3.1). El algoritmo PFMN obtiene importantes mejoras en la eficiencia, pero sin mejoras perceptibles de la calidad. En este trabajo generalizamos GS por medio del uso de esta probabilidad para al aprendizaje de Markov blankets que la maximicen.

3.1 Probabilidad a posteriori

Nuestra principal contribución se sustenta en la hipótesis de que la probabilidad (a posteriori, i.e., dados los datos) de una estructura es una medida respresentativa de su calidad. Por ello, nuestro algoritmo maximiza la probabilidad (ver Sección siguiente). En esta sección reproducimos una expresión que calcula esta probabilidad, obtenida de [12]. Luego presentamos una descomposción de esta probabilidad sobre estructuras en la probabilidad de los Markov blankets de cada variable. Esto reduce la maximización sobre estructuras a maximización del blanket de cada variable.

La siguiente es una expresión aproximada, pero fuertemente fundamentada para calcular la probabilidad $\texttt{Pr}(G \mid D)$ de una estructura dados los datos, i.e., a posteriori:

$$\texttt{Pr}(G \mid D, \mathcal{T}) \propto \texttt{Pr}(G) \prod_{t \in \mathcal{T}} \texttt{Pr}(D \mid a_t^G). \tag{1}$$

La probabilidad de G dados los datos D y el resultado de un conjunto de tests estadísticos sobre los tripletes en $\mathcal{T}$ es igual al producto de la probabilidad $\texttt{Pr}(G)$ a priori de G con la productoria, sobre todo triplete $t \in \mathcal{T}$, de la probabilidad $\texttt{Pr}(D \mid A_t = a_t^G)$ de los datos D dado el hecho de que el triplete t toma el valor de independencia $a_t^G \in \{\texttt{verdadero}, \texttt{falso}\}$ consistente (i.e., codificado

por) G. La cantidad en la productoria es la verosimilitud de los datos dado el valor de independencia de t en G. Estas verosimilitudes pueden calcularse usando el test Bayesiano de Margaritis, que se explicó en la Sección 2, que calcula las verosimilitudes como paso intermedio.

En la sección siguiente presentamos el algoritmo **GSS** (**G**row-**S**hrink with **S**earch), que generaliza el algoritmo **GS** para el aprendizaje de Markov *blankets* presentado en la Sección 2. Dada una variable $X \in \mathbf{V}$, la generalización consiste en modificar GS tal que maximice $\mathtt{Pr}(\mathbf{B}^X \mid D)$, la probabilidad del Markov *blanket* $\mathbf{B}^X$ de X, dados los datos. Para este cálculo, consideremos el conjunto $\mathcal{T}^X$ de tripletes testeados durante la ejecución de GS para el aprendizaje de $\mathbf{B}^X$. Basados en el hecho que $\mathcal{T}^X$ es suficiente para aprender $\mathbf{B}^X$ (por definición), y que aprender $\mathbf{B}^X$ para cada $X \in \mathbf{V}$ es suficiente para aprender G ([14] y Sección 2), tenemos que la unión $\mathcal{T} = \bigcup_{X \in \mathbf{V}} \mathcal{T}^X$ de estos conjuntos de tripletes sobre todas las variables es suficiente para aprender G. De ésto y Eq.(1):

$$\mathtt{Pr}(G \mid \mathcal{D}) \propto \mathtt{Pr}(G) \prod_{t \in \mathcal{T}} \mathtt{Pr}(D_t \mid A_t = a_t^G) = \mathtt{Pr}(G) \prod_{X \in \mathbf{V}} \prod_{t \in \mathcal{T}^X} \mathtt{Pr}(D_t \mid A_t = a_t^G)$$

y asumiendo que la distribución incondicional $\mathtt{Pr}(G)$ sobre G es uniforme, obtenemos que

$$\mathtt{Pr}(G \mid \mathcal{D}) \propto \prod_{X \in \mathbf{V}} \mathtt{Pr}(\mathbf{B}^X \mid \mathcal{D}), \tag{2}$$

$$\mathtt{Pr}(\mathbf{B}^X \mid \mathcal{D}) \propto \prod_{t \in \mathcal{T}^X} \mathtt{Pr}(D_t \mid A_t = a_t^{\mathbf{B}^X(G)}). \tag{3}$$

donde $a_t^{\mathbf{B}^X(G)}$ denota el valor de independencia del triplete t de acuerdo al blanket $\mathbf{B}^X$ de X en el grafo G (i.e., los vecinos de X en G).

4 Enfoque: Búsqueda del Markov blanket más probable.

En esta sección presentamos nuestra contribución: el algoritmo **GSS** (**G**row **S**hrink with **S**earch), una variante del algoritmo GS (c.f. Sección 2). Argumentamos también porqué esperamos que produzca mejoras en la calidad de los Markov blankets generados, frente a dos algoritmos competidores: GS básico y GS-IAMB de [18].

GS presenta dos limitaciones importantes ya reconocidas por trabajos anteriores [13]: ordenamiento predefinido y confianza ciega en el resultado de los tests. En GS, el ordenamiento de variables para recorrer los bucles de ambas fases (grow y shrink) proviene de una heurística sencilla con riesgo de incorporar falsos positivos como miembros del blanket $\mathbf{S}$, i.e., variables encontradas dependientes por el test en la etapa de grow que no pertenecen al blanket, y por lo tanto son removidas en la etapa de shrink. El problema de los falsos positivos es el agrandamiento innecesario de $\mathbf{S}$, con el potencial de producir tests menos confiables y por ende errores en el blanket aprendido.

Consideremos como ejemplo la evolución de GS para aprender el blanket $\mathbf{B}^X$ de la variable 1 en el dominio de tres variables $\mathbf{V} = \{1, 2, 3\}$, cuando la heurística determina el ordenamiento $[2, 3]$:

Verosimilitud independencia	Verosimilitud dependencia	**S**	C	T	E
$\texttt{Pr}(D \mid (1 \perp\!\!\!\perp 2 \mid \emptyset)) = 0.35$	$< \texttt{Pr}(D \mid (1 \not\perp\!\!\!\perp 2 \mid \emptyset)) = \mathbf{0.43}$	$\{2\}$	I	D	D
$\texttt{Pr}(D \mid (1 \perp\!\!\!\perp 3 \mid \{2\})) = \mathbf{0.050}$	$> \texttt{Pr}(D \mid (1 \not\perp\!\!\!\perp 3 \mid \{2\})) = 0.045$	$\{2\}$	D	I	I
$\texttt{Pr}(\mathbf{B}^X \mid D) = 0.43 \times 0.50$	$= 2.15 \times 10^{-2}$				

Table 1: Ordenamiento $[2, 3]$

La tabla muestra (en orden izquierda a derecha de columnas): las verosimilitudes de los datos dados la independencia y dependencia, el valor del conjunto **S** luego de cada test, el valor de independencia correcto (**C**), i.e., en el modelo verdadero subyacente, el valor de independencia arrojado por el test (**T**) que corresponde con el valor de independencia de máxima verosimilitud (en negrita), y el valor elegido (**E**) por el algoritmo (en el caso de GS, este es siempre igual a T ya que confía en los tests ciegamente). El último renglon muestra la probabilidad del blanket calculada de acuerdo a Eq. (3).

Vemos que para este ordenamiento se eligen asignaciones incorrecta en ambas filas. Consideremos el ordenamiento $[3, 2]$:

Verosimilitud independencia	Verosimilitud dependencia	**S**	C	T	E
$\texttt{Pr}(D \mid (1 \perp\!\!\!\perp 3 \mid \emptyset)) = 0.007$	$< \texttt{Pr}(D \mid (1 \not\perp\!\!\!\perp 3 \mid \emptyset)) = \mathbf{0.11}$	$\{3\}$	D	D	D
$\texttt{Pr}(D \mid (1 \perp\!\!\!\perp 2 \mid \{3\})) = \mathbf{0.030}$	$> \texttt{Pr}(D \mid (1 \not\perp\!\!\!\perp 2 \mid \{3\})) = 0.0000007$	$\{3\}$	I	I	I
$\texttt{Pr}(\mathbf{B}^X \mid D) = 0.11 \times 0.30$	$= 3.3 \times 10^{-2}$				

Table 2: Ordenamiento $[3, 2]$

Vemos que además de ser la probabilidad del blanket mayor para $[2, 3]$ ($\mathbf{3.3 \times 10^{-2}} > 2.15 \times 10^{-2}$), ambas filas resultan en tests correctos, es decir, mayor probabilidad produce mejor calidad. En vista a estas falencias, en [18] se discute la variante IAMB de GS que produce blankets de mejor calidad con el uso de una mejor heurística de ordenamiento de GS. Ésta considera, en cada iteración de la fase de grow, la incorporación a **S** de la variable más dependiente (según lo indica un test condicionado en el valor de **S** para esa iteración). Si bien GS-IAMB mejora en la calidad, tanto GS como GS-IAMB confían plenamente en el resultado de los tests a pesar de que alguno de ellos pueda ser erróneo.

Ilustremos cómo una asignación alternativa, es decir, una asignación que a diferencia de GS, contradice las asignaciones de independencias de algunos tests, puede resultar en un blanket más probable y correcto. Consideremos el caso del ordenamiento $[2, 3]$, pero con asignación alternativa:

Verosimilitud independencia	Verosimilitud dependencia	**S**	C	T	E
$\texttt{Pr}(D \mid (1 \perp\!\!\!\perp 2 \mid \emptyset)) = 0.35$	$< \texttt{Pr}(D \mid (1 \not\perp\!\!\!\perp 2 \mid \emptyset)) = \mathbf{0.43}$		I	D	I
$\texttt{Pr}(D \mid (1 \perp\!\!\!\perp 3 \mid \emptyset)) = 0.007$	$> \texttt{Pr}(D \mid (1 \not\perp\!\!\!\perp 3 \mid \emptyset)) = \mathbf{0.11}$	$\{3\}$	D	D	D
$\texttt{Pr}(\mathbf{B}^X \mid D) = 0.35 \times 0.11$	$= 3.55 \times 10^{-2}$				

Table 3: Ordenamiento $[2, 3]$ con asignación alternativa

Vemos que aquí, a diferencia de GS ejemplificado arriba, las decisiones del algoritmo (columna E) no respetan la decisión del test (columna T). Esto resulta en una probabilidad de blanket (tercer renglón) superior a la obtenida para GS (2.15×10^{-2}). Más aún, este aumento de probabilidad coincide con tests más correctos, siendo ambos en este caso (contra cero en el caso de GS).

4.1 Algoritmo GSS: Búsqueda de ordenamiento y asignación óptima

El algoritmo propuesto, GSS, flexibiliza tanto el ordenamiento de variables, como la asignación de valores de independencias de los tests durante la etapa de grow. Así, para el aprendizaje de $\mathbf{B}^X, X \in \mathbf{V}$, GSS realiza una búsqueda sistemática sobre todos los posibles ordenamientos de las variables restantes $\mathbf{V} - \{X\}$ y de las 2^{n-1} asignaciones de cada una de ellas, buscando aquella cuya secuencia de tests $\mathcal{T}^X$ resulte en el blanket de mayor probabilidad. Para ello, realiza una búsqueda en árboles [15] en busca de la rama con costo de camino óptimo. En la Fig.1 se ilustra esta búsqueda para los casos explicados en la Sección 4. Los caminos del árbol A, B y C serían los recorridos por GSS para los casos de la primera, segunda y tercer tabla, respectivamente. La búsqueda comienza con 1, la variable para la cual se quiere aprender el blanket. Para considerar todos los posibles ordenamientos y todas las posibles asignaciones, el algoritmo considera como sucesores de 1 a todos las restantes variables, i.e., 2 y 3, y para los tests correspondientes a estas variable las asignaciones de independencia y dependencia. Es decir, los nodos sucesores $2I, 2D, 3I, 3D$. Esto continúa recursivamente. Por ejemplo, los sucesores de $2I$ serían todas las restantes variables, 3 solamente en este caso, con las dos posibles asignaciones $3I$ y $3D$.

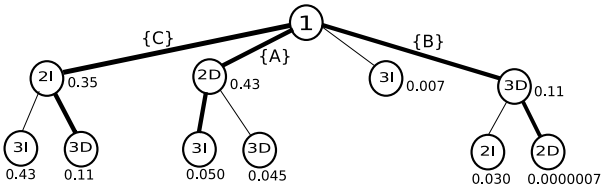

Fig. 1: Expansión de un árbol de búsqueda de GSS

Así, en el nivel $j = n - 1$ encontraremos todas las posibles variantes de GS para cada ordenamiento y cada asignación posible. Para poder calcular la probabilidad del blanket, demarcamos en cada arista la verosimilitud del test correspondiente (e.g., para la segunda arista del camino 1-3D-2I el test es sobre $(1, 2 \mid \{3\})$). El algoritmo de búsqueda en árboles *minimiza* el costo de camino, i.e., la suma de los costos de cada arista, mientras que nosotros buscamos *maximizar* la probabilidad, un producto de verosimilitudes. Por ello, siendo que el logaritmo es una función monótona, es correcto asignar como costo de cada arista el negativo del logaritmo de la verosimilitud. En nuestros experimentos consideramos dos versiones de búsqueda en árboles. La del algoritmo de costo uniforme que siempre encuentra la solución óptima, y el algoritmo A* que utiliza una heurística para eficientizar la búsqueda, pero corre el riesgo de obtener una

solución subóptima. En nuestros experimentos consideramos como heurística en cada nodo la probabilidad (convertida a costos como el negativo del logaritmo) que hubiera obtenido una variante del algoritmo IAMB denominada **greedy** de haberse ejecutado sobre el dominio $\mathbf{V}$, menos los predecesores del nodo en el árbol (básicamente una búsqueda voraz desde el nodo actual del árbol hasta la solución). Esto garantiza un cálculo eficiente de la heurística.

5 Resultados experimentales

Los resultados experimentales presentados en esta sección demuestran que el enfoque propuesto resulta en importantes mejoras en la calidad de los blankets obtenidos (medidos a través de la calidad de las estructuras que con ellos se aprendieron). Los experimentos comparan la calidad de estructuras Markovianas obtenidas con el algoritmo **GSMN**, **GSMNS-Greedy**, símil a GSMN pero usando variante greedy de GS (c.f. Sección 4), y **GSMNS** que utiliza GSS, nuestra variante de GS con búsqueda, en dos variantes **GSMNS-Costo-Uniforme** y **GSMNS-A*** que utilizan las estrategias de búsqueda de costo uniforme y A*, respectivamente. La comparación con GSMNS-Greedy es para mostrar la mejora de GSS frente a su propia heurística, i.e., GS-Greedy.

Para calcular la calidad de una estructura aprendida (por alguno de los tres algoritmos) la comparamos con la estructura del modelo subyacente. Para ello, utilizamos para el aprendizaje diversos datasets generados de muestrear (Gibbs sampler) redes Markovianas generadas aleatoriamente. De esta manera conocemos la red subyacente. Se generaron redes de $n = 8$, 16 y 20 variables, uniendo cada nodo de la red con $\tau = 1, 2, 4$ variables seleccionadas aleatoria y uniformemente de entre los nodos restantes. Para cada n y τ se generaron 10 redes, y para cada una de ellas un dataset. La calidad de la red generada es estimada midiendo la *distancia de Hamming normalizada* (o DHN) entre la red generada y la red real. La distancia de Hamming entre dos redes consiste en la cantidad de aristas existentes en una de las redes e inexistentes en la otra (y viceversa). Se normaliza dividiendo por la cantidad total de pares de variables (i.e., el valor máximo de la distancia de Hamming no normalizada), obteniendo un valor entre 0 y 1, correspondientes a redes iguales y redes totalmente discímiles, respectivamente. En las figuras reportamos este valor multiplicado por 100. Dado que la calidad de los tests (y de las estructuras generadas) dependen de la cantidad de datos en el conjunto de datos, corrimos el algoritmo para subconjuntos de D con un número $|D|$ creciente de renglones, también tomados aleatoriamente del conjunto de datos original.

La Fig. 2 compara resultados para $n = 8$, $\tau = 1, 2, 4$ de GSMNS-Costo-Uniforme y GSMNS-A*. La figura muestra el valor medio de la DHN sobre diez datasets para valores crecientes de $|D|$. Las barras de error denotan la desviación estandard. El objetivo de estos experimentos fue demostrar que GSMNS-A*, a pesar de ser subóptimo (debido a la heurística), recupera estructuras de calidad comparable a GSMNS-Costo-Uniforme, que sí es óptimo. Esta optimalidad genera un costo computacional tan alto que impidió experimentos para dominios superiores a $n = 8$. Los resultados confirman el excelente desempeño de GSMNS-

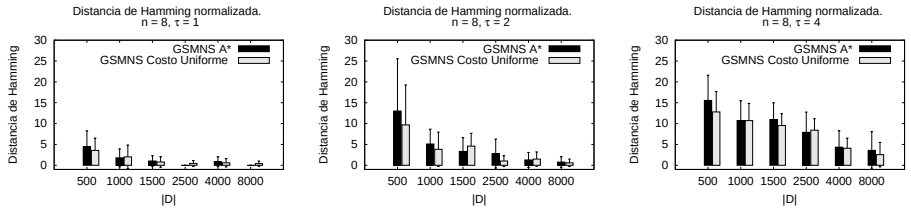

Fig. 2: Comparación de la distancia de Hamming normalizada de redes obtenidas por GSMNS-A*, y GSMNS-Costo-Uniforme (i.e., sin heurística), para $n = 8$ y $\tau = 1, 2, 4$.

A* para todos los casos, con valores medios a veces menores que GSMNS-Costo-Uniforme, pero dentro del error estadístico.

La Fig. 3 compara resultados para $n = 16, 20$ (renglones superior e inferior, respectivamente), y $\tau = 1, 2, 4$ de GSMNS-A*, y sus competidores GSMN y GSMNS-Greedy. La figura presenta la DHN sobre diez datasets para valores crecientes de $|D|$. Puede observarse que para todos los valores de n y τ, y para todos los algoritmos, la DHN disminuye para $|D|$ crecientes. Esto confirma que la calidad de los tests crece con $|D|$. Además, puede observarse también para todos los valores de n y τ, que la DHN de GSMNS disminuye más rápido que la de sus algoritmos competidores, i.e., tiene un porcentaje de error menor que el de sus competidores, registrándose disminuiciones del porcentaje de error frente a GSMN mayores al 10% en muchos casos y de hasta 20% frente a GSMNS-Greedy.

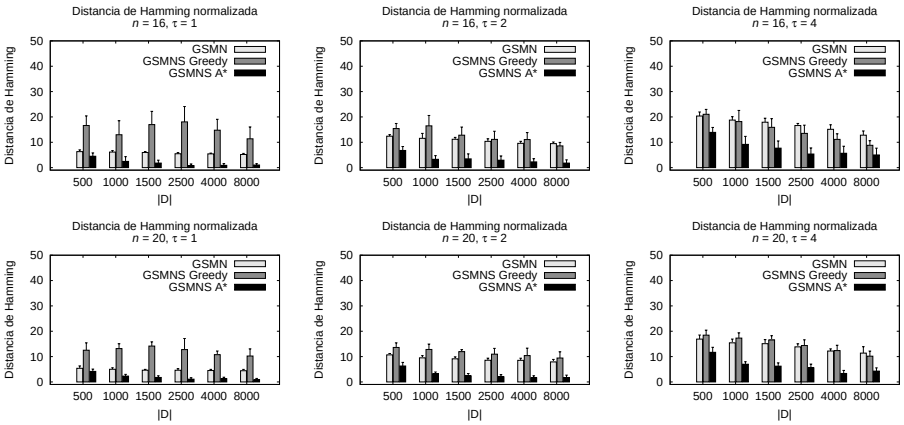

Fig. 3: Comparación de la distancia de Hamming normalizada de las redes obtenidas por GSMN, GSMNS y GSMNS-Greedy para tamaños crecientes del dataset de entrada, $n = 16$ (renglon superior) y $n = 20$ (renglón inferior), y $\tau = 1, 2, 4$ (columnas).

6 Conclusiones

En este paper se presenta GSS, una variante de GS para mejorar la calidad de Markov blankets y estructuras de independencia obtenidas con algoritmos basados en independencia. Evaluamos la calidad de los resultados a través de

datos muestreados, comparando la distancia de Hamming normalizada entre las redes obtenidas y las esperadas. El algoritmos GSMNS, utilizando la subrutina GSS, ha demostrado una reducción importante en los porcentajes de error respecto de sus competidores más cercanos, el algoritmos GSMN (que utiliza GS) y GSMN-Greedy (que utiliza la variante IAMB de GS). Si bien, en términos de costos computacionales, este algoritmo no es práctico, los resultados son altamente prometedores. Nuestro trabajo futuro se concentrará en continuar este camino hacia la búsqueda de heurísticas más óptimas o algoritmos de búsqueda más eficientes, con la intención de generar un algoritmo práctico que sacrifique mínimamente las mejoras en calidad obtenidas en este trabajo.

References

[1] A. Agresti. *Categorical Data Analysis*. Wiley, 2nd edition, 2002.

[2] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of Markov random fields for segmentation of 3D range data. *Proceedings of the CVPR*, 2005.

[3] F. Barahona. On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241–3253, 1982.

[4] J. Besag. Spacial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 1974.

[5] J. Besag, J. York, and A. Mollie. Bayesian image restoration with two applications in spatial statistics. *Annals of the Inst. of Stat. Math.*, 43:1–59, 1991.

[6] F. Bromberg and D. Margaritis. Improving the reliability of causal discovery from small data sets using argumentation. *JMLR*, 10:301–340, Feb 2009.

[7] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Computational Biology*, 7:601–620, 2000.

[8] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 1995.

[9] W. Lam and F. Bacchus. Learning Bayesian belief networks: an approach based on the MDL principle. *Computational Intelligence*, 10:269–293, 1994.

[10] S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[11] D. Margaritis. Distribution-free learning of Bayesian network structure in continuous domains. In *AAAI*, 2005.

[12] D. Margaritis and F. Bromberg. Efficient markov network discovery using particle filter. *Computational Intelligence*, 2009. In press.

[13] D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. In S. Solla, T. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 505–511. MIT Press, 2000.

[14] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.

[15] S. Russel and P. Norvig. *Artificial Intelligence. A Modern Approach.* 2nd Ed., 2002.

[16] S. Shekhar, P. Zhang, Y. Huang, and R. R. Vatsavai. In H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha, editors, *Trends in Spatial Data Mining*, chapter 19, pages 357–379. AAAI Press / The MIT Press, 2004.

[17] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Adaptive Computation and Machine Learning Series. MIT Press, 2000.

[18] I. Tsamardinos, C. Aliferis, and A. Statnikov. Algorithms for large scale markov blanket discovery. In *FLAIRS*, 2003.