

REDES BAYESIANAS

Eva Besada-Portas y Diane Oyen

A6.1 INTRODUCCIÓN

A continuación se describe el ejemplo utilizado en el capítulo 6 para ilustrar el proceso de aprendizaje de las redes bayesianas (RB).

El problema elegido es el modelado de las relaciones que se establecen entre diferentes partes del cerebro con una Red Bayesiana Dinámica (RBD). Su solución consiste en obtener automáticamente la RBD que mejor modela el comportamiento recogido en un conjunto de muestras.

En las siguientes secciones se describe el problema, los datos disponibles, los programas¹ de Matlab utilizados para el procesamiento de los datos, el aprendizaje de la RBD y los resultados obtenidos.

¹ Al final de este documento se presenta una tabla explicativa de todos los ficheros de Matlab proporcionados. Aquellos cuyo nombre empieza por “*Main*” se corresponden a los diferentes programas principales explicados a lo largo del documento.

A6.2 DESCRIPCION DEL PROBLEMA

Para estudiar el comportamiento del cerebro, éste puede ser dividido en un conjunto de regiones denominadas ROIs (*Regions of Interest*) en función de las tareas específicas realizadas en cada zona. Mientras el individuo piensa o realiza alguna actividad compleja, se produce la activación neuronal de algunas ROIs y el intercambio de información entre diferentes ROIs. En este momento, la comunidad científica sabe qué ROIs se activan al realizar múltiples actividades, pero todavía no entiende bien como las ROIs interactúan entre sí para realizar tareas o razonamientos complejos.

Por lo tanto, la obtención automática de modelos que representen las relaciones que se establecen entre las diferentes ROIs puede ayudar a comprender las interacciones que se dan entre las diferentes zonas del cerebro. Para poder llevar a cabo esta labor es necesario disponer de algún tipo de medidas de la actividad cerebral en las diferentes ROIs, elegir un tipo de modelo que permita establecer las relaciones existentes entre las mismas, y un algoritmo de aprendizaje que permita obtener los parámetros del modelo elegido.

Un tipo de medidas que permiten detectar qué ROIs se encuentran activas en un momento determinado son las imágenes de resonancia magnética funcional (fMRI). Durante los experimentos habituales se recoge una serie temporal de medidas de fMRI mientras se estimula al sujeto, habitualmente de forma auditiva o visual, y se le pide que responda al estímulo, habitualmente pulsando un botón. A partir de dichas imágenes se determina la actividad cerebral en diferentes instantes de tiempo en las diferentes ROIs.

Las RBD son una herramienta particularmente útil para modelar las interacciones funcionales que se observan entre las diferentes ROIs a la hora de realizar alguna tarea o razonamiento complejo. En este tipo de interacciones, también denominadas conectividad efectiva, no es necesario que las ROIs estén físicamente conectadas, sino únicamente que se comparta de alguna forma información entre dos ROIs, y que este hecho se refleje en la existencia de algún tipo de correlación entre las series temporales de actividad cerebral asociadas a las diferentes ROIs. Por lo tanto, la RBD es utilizada para modelar la dinámica de la influencia funcional que una ROI recibe de otra sin necesidad de asumir cómo se establecen las relaciones funcionales ni sus formas. Otros modelos de conectividad efectiva basada en fMRI necesitan asumir que las relaciones son lineales o que la influencia únicamente se produce entre parejas de ROIs.

La RBD utilizada en este problema es discreta, Markoviana y estacionaria. Cada una de sus variables representa el nivel de actividad cerebral de una ROI determinada. Las interconexiones se establecen entre los nodos asociados a las

ROIs en el instante $t-1$ y los nodos asociados a las ROIs en el instante t . No se consideran intraconexiones, ya que el objeto del estudio es medir la influencia funcional entre ROIs en instantes de tiempo sucesivos.

Para obtener los parámetros del modelo elegido, que en el caso de la RBD seleccionada son las interconexiones y las tablas de probabilidad asociadas a cada ROI, utilizaremos un algoritmo voraz basado en el Criterio de Información Bayesiana (BIC). Debido a las características de la RBD utilizada, en la que sólo hay conexiones entre las ROIs en un instante y el posterior, en este problema no es posible observar estructuras Markov-equivalentes. Por último, con el objeto de reducir el espacio de búsqueda, el número de nodos que pueden ser padres de cada nodo de la red, es decir el número de ROIs que pueden tener una influencia en la actividad cerebral de otras regiones en el siguiente instante de medida, se encuentra limitado.

A6.3 DATOS DEL PROBLEMA

Los datos que se utilizan para aprender la estructura de las RBD que modelan la conectividad funcional que se da entre diferentes ROIs son las medidas de fMRI originadas en la Competición de Interpretación de la Actividad Cerebral de Pittsburgh (<http://pbc.lrdc.pitt.edu/?q=2006-home>) en la que 2 sujetos se encontraban viendo 2 películas e indicaban qué estaban observando en las mismas. Por lo tanto, disponemos de 4 conjuntos de datos diferentes (2 sujetos x 2 películas) de los que podemos aprender 4 RBD y comparar si las conexiones que se establecen en los 4 casos son iguales o diferentes.

En los 4 casos se utilizan las 16 ROIs que se presentan en la tabla A6.1 y que se consideran relevantes a la hora de ver películas. En la figura A6.1 se representan el Hippocampus, la Insula y Striatum del lado izquierdo.

ROI	Izquierda (_L, "Left")	Derecha (_R, "Right")
<i>Frontal_Inf_Orb</i>	1	2
<i>Frontal_Latx_Orb</i>	3	4
<i>Frontal_Mid_Orb</i>	5	6
<i>Frontal_Sup_Orb</i>	7	8
<i>Hippocampus</i>	9	10
<i>Insula</i>	11	12
<i>ParaHippocampal</i>	13	14
<i>Striatum</i>	15	16

Tabla A6.1. Etiquetas numéricas asociadas a las diferentes ROIs

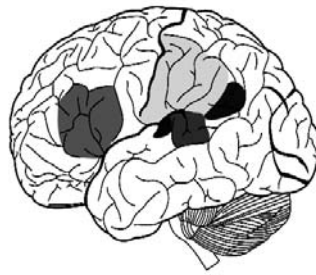


Figura A6.1. Tres ROIs del problema del lado izquierdo.

El grado de activación de cada una de las regiones durante múltiples instantes de tiempos sucesivos se obtiene a partir de las medidas de fMRI. Los valores de activación asociados a cada región y cada instante de tiempo son variables continuas.

Para cada uno de los 4 casos los datos se almacenan en los ficheros “*datasub2run1.csv*”, “*datasub2run2.csv*”, “*datasub3run1.csv*” y “*datasub3run2.csv*”. El primer número identifica al sujeto² (“*subject*”) y el segundo a la película (“*run*”).

A6.4 PROCESAMIENTO DE LOS DATOS

Para poder aprender la RBD que modela las relaciones que se establecen entre las diferentes ROIs de cada individuo mientras observa cada película es necesario: 1) cargar los datos originales, 2) discretizarlos, 3) aprender la estructura de cada RBD, 4) aprender las tablas de probabilidad correspondientes a cada una de las ROIs en cada una de las redes, y 5) validar los resultados. Todas estas labores, que se describen de forma detallada en las siguientes secciones, se llevan a cabo mediante el programa principal “*MainDBN.m*”

² Nótese que la numeración de los sujetos en los ficheros va del 2 al 3, mientras que nosotros hablamos del sujeto 1 y del sujeto 2. Esto se debe a que la numeración de los sujetos en los ficheros es la que aparece en la base de datos de Pittsburgh, de la que nosotros hemos elegido al sujeto 2 y 3 como segundo y primer sujeto de nuestros experimentos.

A6.4.1 Cargar los datos

La función genérica de Matlab “*importData.m*” permite leer el contenido de los ficheros “*datasub2run1.csv*”, “*datasub2run2.csv*”, “*datasub3run1.csv*” y “*datasub3run2.csv*”, y obtener una matriz *dataorig* de $T \times N$ elementos, donde T representa el número de instantes de tiempo en los que se han tomado las medidas, N el número de ROIs, y *dataorig*(i, j) el valor de la j -ésima ROI en el i -ésimo instante de tiempo.

A6.4.2 Discretización de los datos originales

Como las variables de activación originales toman valores continuos y las variables de la RBD con la que se desea modelar el comportamiento del sistema toman valores discretos es necesario implementar un algoritmo que discretice los valores de las variables de activación de forma automática.

Para llevar a cabo esta labor se utiliza un método simple de discretización en niveles distribuidos de forma uniforme entre el valor máximo y mínimo de actividad observada. Los valores de activación asociados a cada ROI se discretizan independientemente a partir del valor máximo y mínimo observado en cada ROI.

El método elegido se implementa en la función “*quantize.m*”, utilizando el mismo número de niveles para todas las ROIs. El número de niveles elegido, modificable en el programa “*MainDBN.m*” a través de la variable “*numBins*”, es igual a 4 (Muy Bajo=1, Bajo=2, Alto=3, Muy Alto=4) ya que observamos que producía buenos resultados en este experimento. La función “*quantize.m*” admite como parámetro de entrada los datos originales “*datosorig*” y devuelve los datos discretizados “*datosdis*”.

A6.4.3 Aprendizaje de la estructura de la RBD

Para aprender la estructura de la RBD utilizaremos un algoritmo de búsqueda voraz basado en la función $Score(\mathcal{G}, \mathcal{D})$ del BIC que se implementa en la ecuación (6.4).

Elegimos BIC porque mide lo bien que cada \mathcal{G} modela los \mathcal{D} y porque incluye un término que penaliza a los \mathcal{G} complejos. Por lo tanto, esta función $Score(\mathcal{G}, \mathcal{D})$ prefiere redes más sencillas, con pocos padres por nodo, a no ser que la mejora introducida en el modelado del comportamiento de los \mathcal{D} al

incrementar el numero de padres supere la penalización asociada a dicho incremento.

Como en la RBD elegida únicamente se permiten los enlaces que se establecen entre las ROIs en el instante $t-1$ y las ROIs en el instante t y BIC calcula de forma independiente la parte asociada a cada variable dados sus padres, se puede aprender de forma independiente los enlaces asociados a cada ROI en el instante t . Este hecho simplifica la implementación del método de aprendizaje de la estructura de la red.

La técnica voraz implementada en “*learn_dbn_greedy.m*” funciona iterativamente de la forma siguiente: para cada ROI en el instante t identificamos qué ROI en el instante $t-1$ produciría un valor más alto de BIC y por lo tanto cuál sería su mejor padre. Tras incluir dicha ROI como padre, identificamos la siguiente ROI del instante $t-1$ que produciría un mayor incremento de BIC. El proceso de inclusión de padres continúa hasta que no se produce ninguna mejora o hasta que se alcanza el máximo número de padres permitido para cada ROI. Este número es modificable en el programa “*MainDBN.m*” por medio de la variable “*max_fan_in*” (*max_fan_in*=2 en los resultados mostrados en este apéndice y en el capítulo 6).

En las siguientes figuras se presentan las 4 RBD aprendidas, identificando los nodos por medio de las etiquetas recogidas en la tabla A6.1. El programa principal “*MainDBN.m*”, además de aprender las estructuras de las RBD, también realiza la representación gráfica de las estructuras aprendidas por medio de una llamada a la función “*draw_dbn.m*”.

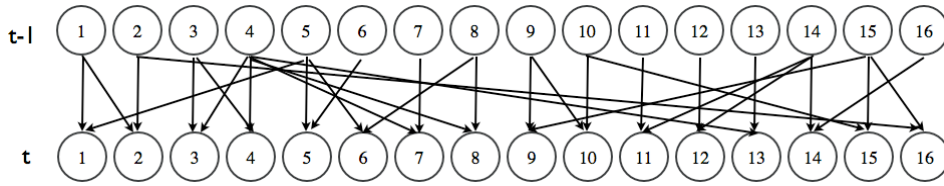


Figura A6.2. RBD asociado al sujeto 1 viendo la película 1

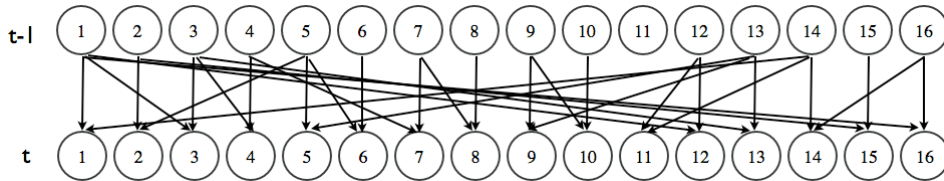


Figura A6.3. RBD asociado al sujeto 1 viendo la película 2

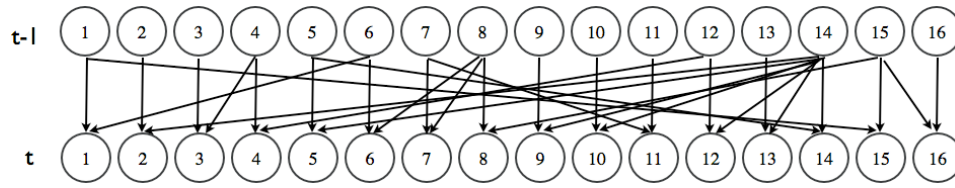


Figura A6.4. RBD asociado al sujeto 2 viendo la película 1

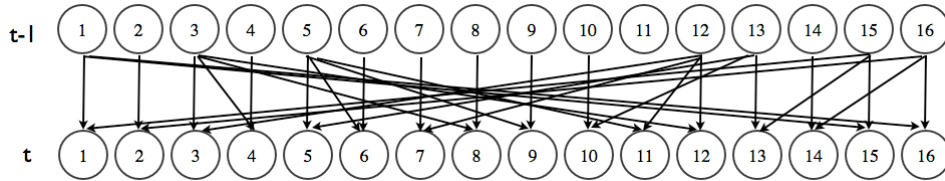


Figura A6.5. RBD asociado al sujeto 2 viendo la película 2

Los científicos están normalmente interesados en qué ROIs se encuentran influenciadas por otras. Por lo tanto, normalmente se muestran las familias (un nodo hijo y sus padres) en vez de la estructura completa. La función “*display_families.m*”, utilizada en el programa principal “*MainDBN.m*”, nos muestra las diferentes familias para los cuatro problemas. Los resultados se recogen en las cuatro tablas siguientes.

Hijo t	Padres $t-1$	Hijo t	Padres $t-1$
Frontal_Inf_Orb_L	Frontal_Inf_Orb_L Frontal_Mid_Orb_L	Frontal_Inf_Orb_R	Frontal_Inf_Orb_R Frontal_Inf_Orb_L
Frontal_Latx_Orb_L	Frontal_Latx_Orb_L Frontal_Latx_Orb_R	Frontal_Latx_Orb_R	Frontal_Latx_Orb_R Frontal_Latx_Orb_L
Frontal_Mid_Orb_L	Frontal_Mid_Orb_L Frontal_Mid_Orb_R	Frontal_Mid_Orb_R	Frontal_Mid_Orb_R Frontal_Sup_Orb_R
Frontal_Sup_Orb_L	Frontal_Sup_Orb_L Frontal_Latx_Orb_R	Frontal_Sup_Orb_R	Frontal_Sup_Orb_R Frontal_Latx_Orb_R
Hippocampus_L	Hippocampus_L Striatum_L	Hippocampus_R	Hippocampus_R Hippocampus_L
Insula_L	ParaHippocampal_R Insula_L	Insula_R	Insula_R ParaHippocampal_R
ParaHippocampal_L	ParaHippocampal_L Frontal_Latx_Orb_R	ParaHippocampal_R	Striatum_R ParaHippocampal_R
Striatum_L	Hippocampus_R Striatum_L	Striatum_R	Striatum_L Frontal_Inf_Orb_R

Tabla A6.2. Familias para sujeto 1 viendo película 1

Hijo t	Padres $t-1$	Hijo t	Padres $t-1$
Frontal_Inf_Orb_L	Frontal_Inf_Orb_L ParaHippocampal_R	Frontal_Inf_Orb_R	Frontal_Inf_Orb_R Frontal_Mid_Orb_L
Frontal_Latx_Orb_L	Frontal_Latx_Orb_L Frontal_Inf_Orb_L	Frontal_Latx_Orb_R	Frontal_Latx_Orb_R Frontal_Latx_Orb_L
Frontal_Mid_Orb_L	Frontal_Mid_Orb_L ParaHippocampal_L	Frontal_Mid_Orb_R	Frontal_Mid_Orb_L Frontal_Mid_Orb_R
Frontal_Sup_Orb_L	Frontal_Sup_Orb_L Frontal_Latx_Orb_R	Frontal_Sup_Orb_R	Frontal_Sup_Orb_R Frontal_Sup_Orb_L
Hippocampus_L	Hippocampus_L ParaHippocampal_L	Hippocampus_R	Hippocampus_L Hippocampal_R
Insula_L	ParaHippocampal_R Insula_R	Insula_R	Insula_R Frontal_Inf_Orb_L
ParaHippocampal_L	ParaHippocampal_L Frontal_Latx_Orb_L	ParaHippocampal_R	Striatum_R ParaHippocampal_R
Striatum_L	Frontal_Inf_Orb_R Striatum_L	Striatum_R	Frontal_Inf_Orb_L Striatum_R

Tabla A6.3. Familias para sujeto 1 viendo película 2

Hijo t	Padres $t-1$	Hijo t	Padres $t-1$
Frontal_Inf_Orb_L	Frontal_Inf_Orb_L Frontal_Mid_Orb_R	Frontal_Inf_Orb_R	Frontal_Inf_Orb_R ParaHippocampal_R
Frontal_Latx_Orb_L	Frontal_Latx_Orb_L Frontal_Latx_Orb_R	Frontal_Latx_Orb_R	Frontal_Latx_Orb_R Insula_R
Frontal_Mid_Orb_L	Frontal_Mid_Orb_L ParaHippocampal_R	Frontal_Mid_Orb_R	Frontal_Mid_Orb_R Frontal_Sup_Orb_R
Frontal_Sup_Orb_L	Frontal_Sup_Orb_L Frontal_Sup_Orb_R	Frontal_Sup_Orb_R	Frontal_Sup_Orb_R Striatum_L
Hippocampus_L	Hippocampus_L ParaHippocampal_R	Hippocampus_R	Hippocampus_R ParaHippocampal_R
Insula_L	Frontal_Sup_Orb_L Insula_L	Insula_R	Insula_R ParaHippocampal_R
ParaHippocampal_L	ParaHippocampal_L ParaHippocampal_R	ParaHippocampal_R	Frontal_Mid_Orb_L ParaHippocampal_R
Striatum_L	Frontal_Inf_Orb_L Striatum_L	Striatum_R	Striatum_L Striatum_R

Tabla A6.4. Familias para sujeto 2 viendo película 1

Hijo t	Padres $t - 1$	Hijo t	Padres $t - 1$
Frontal_Inf_Orb_L	Frontal_Inf_Orb_L Striatum_L	Frontal_Inf_Orb_R	Frontal_Inf_Orb_R Striatum_R
Frontal_Latx_Orb_L	Frontal_Latx_Orb_L Insula_R	Frontal_Latx_Orb_R	Frontal_Latx_Orb_R Frontal_Latx_Orb_L
Frontal_Mid_Orb_L	Frontal_Mid_Orb_L ParaHippocampal_L	Frontal_Mid_Orb_R	Frontal_Mid_Orb_R Frontal_Mid_Orb_L
Frontal_Sup_Orb_L	Frontal_Sup_Orb_L Insula_R	Frontal_Sup_Orb_R	Frontal_Sup_Orb_R Frontal_Latx_Orb_L
Hippocampus_L	Hippocampus_L Frontal_Mid_Orb_L	Hippocampus_R	Hippocampus_R ParaHippocampal_R
Insula_L	Frontal_Mid_Orb_L Insula_R	Insula_R	Insula_R Frontal_Latx_Orb_L
ParaHippocampal_L	ParaHippocampal_L Striatum_L	ParaHippocampal_R	ParaHippocampal_R Striatum_R
Striatum_L	Frontal_Inf_Orb_L Striatum_L	Striatum_R	Frontal_Inf_Orb_L Striatum_R

Tabla A6.5. Familias para sujeto 2 viendo película 2

A6.4.4 Aprendizaje paramétrico de la RBD

Una vez conocida la estructura de la RBD para cada uno de los cuatro casos, se pueden aprender las tablas de probabilidad asociada a cada nodo dados sus nodos padres. Como la estructura ha sido adquirida utilizando BIC, que no incluye ninguna información a priori sobre el valor de los parámetros, éstos deben ser aprendidos sin ese tipo de información.

Por lo tanto, utilizaremos la ecuación (6.2) en la función implementada para aprender las tablas de probabilidad “*learn_cpt.m*”.

Los resultados permiten a los científicos visualizar el tipo de relación (activación/inhibición) que se establece entre los diferentes miembros de la familia. Por lo tanto es conveniente utilizar una herramienta de visualización atractiva. Los mapas de calor resultan adecuados ya que permiten representar fácilmente la fuerza estadística de las relaciones, por lo que es el método utilizado en la función “*draw_cpt.m*”.

El número de mapas necesarios para cada uno de las variables de los 4 casos depende del tamaño de la familias (3 elementos en todos los casos) y del número de valores de cada variable (4 en todos ellos). Estos valores se pueden representar con 4 mapas, uno para cada combinación de valores del nodo hijo. Como cada caso tiene 16 variables, necesitamos un total de 4×16 mapas de calor

para cada uno de ellos. Debido a la existencia de los 4 casos, el total de mapas de calor necesarios es $4 \times 16 \times 4$. El programa los representa en grupos de 16. En la figura A6.7 se muestra uno de los grupos correspondientes al problema del sujeto 1 viendo la película 1. En cada fila se representa una variable hijo determinada y en cada columna uno de los valores de la variable hijo. La variable hijo y su valor aparecen en la parte superior de cada mapa de calor. Los ejes de abscisas y ordenadas de cada mapa se corresponden con los valores de sus variables padres. El negro representa una probabilidad nula del nodo hijo correspondiente a una combinación de valores de los padres y el blanco una probabilidad total. Por ejemplo, en el caso de la variable #1 y combinación de valores del padre #5 igual a 1 y del padre #1 igual a 4 (esquina superior derecha de los cuatro mapas de calor de la primera fila) se observa que la probabilidad es 1 cuando la variable #1 es igual a 1 y cero en los tres casos restantes. Esto indica que si en el instante $t-1$ la ROI #5 tiene una actividad muy baja y la ROI #1 un actividad muy alta, la actividad de la ROI #1 será muy baja en el instante t . El resto de las figuras se pueden obtener de forma automática mediante el programa “MainDBN.m”

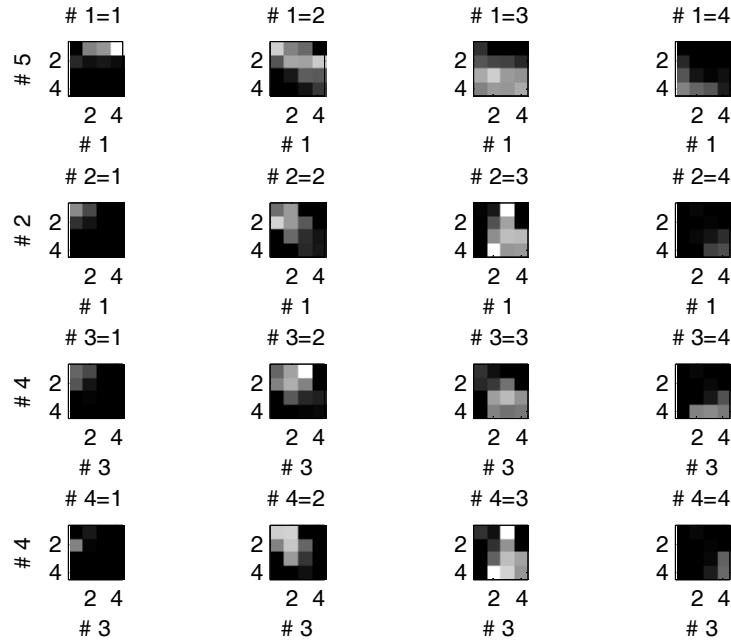


Figura A6.6. Representación gráfica de las tablas de probabilidad para el sujeto 1 y película 1 y las ROIs etiquetadas como #1, #2, #3 y #4.

A6.4.5 Validación de los resultados

Como en este problema se desconoce cuál es la red “real” que describe las relaciones entre los niveles de activación, la validación de los resultados se puede realizar comprobando si se obtienen redes similares en múltiples experimentos (combinaciones de sujeto-película) y si en ellas se observa la variabilidad de relaciones esperada: mayor variación en las relaciones entre sujetos que entre películas.

Este hecho puede ser verificado evaluando la función $Score(\mathcal{G}_i, \mathcal{D}_j)$ elegida, BIC, sobre las 4 estructuras \mathcal{G}_i aprendidas y las 4 bases de datos \mathcal{D}_j . Los resultados se obtienen al final del programa principal “*MainDBN.m*” por medio de la función “*calc_dbn_score.m*”. Los resultados se presentan en la tabla A6.6 donde cada fila se asocia con la \mathcal{G}_i aprendida en el caso mencionado, y cada columna con los datos \mathcal{D}_j asociados a cada caso. En la diagonal se presenta $Score(\mathcal{G}_i, \mathcal{D}_i)$, es decir el valor de la bondad de la DBN sobre el caso utilizado para entrenarlo, motivo por lo que es el mejor³ valor que se obtiene para una \mathcal{G}_i determinada respecto a todos los casos posibles. El segundo valor mejor es marcado con un asterisco *. En tres de los cuatro casos posibles, coincide con los datos del mismo sujeto observando una película diferente, hecho que pone de manifiesto el comportamiento esperado: mayor variación de actividad cerebral entre sujetos que entre películas.

	\mathcal{D} S1 P1	\mathcal{D} S1 P2	\mathcal{D} S2 P1	\mathcal{D} S2 P2
\mathcal{G} S1 P1	-13230	* -190290	-350650	-419560
\mathcal{G} S1 P2	* -377560	-12240	-840170	-489880
\mathcal{G} S2 P1	-302630	-260210	-13770	* -223140
\mathcal{G} S2 P2	* -269690	-427890	-416710	-12980

Tabla A6.6. BIC $Score(\mathcal{G}_i, \mathcal{D}_j)$

³ Cuanto mayor es un valor (mas pequeño negativamente) mejor es el $Score(\mathcal{G}_i, \mathcal{D}_j)$.

A6.5 LISTADO DE FUNCIONES Y PROGRAMAS

Finalmente se incluye el listado de funciones, programas y ficheros de datos suministrados, con una breve descripción de su utilidad. El nombre del programa principal ha sido resaltado en negrita.

MainDBN.m	<i>Programa principal que realiza todas las tareas necesarias para aprender las RBD y las tablas de probabilidad</i>
<i>bic_score_family.m</i>	<i>Función que calcula el valor de BIC asociado a una familia (un nodo y sus padres)</i>
<i>calc_dbn_score.m</i>	<i>Función que calcula BIC para una RBD y conjunto de datos</i>
<i>compute_counts.m</i>	<i>Función que calcula el número de elementos que se encuentran en las distintas configuraciones establecidas dentro de una familia</i>
<i>dataSub2run1.asv</i>	<i>Datos del sujeto 2 película 1</i>
<i>dataSub2run2.asv</i>	<i>Datos del sujeto 2 película 2</i>
<i>dataSub3run1.asv</i>	<i>Datos del sujeto 1 película 1</i>
<i>dataSub3run2.asv</i>	<i>Datos del sujeto 1 película 2</i>
<i>display_families.m</i>	<i>Función que muestra por pantalla cada una de las familias de una RBD</i>
<i>draw_cpt.m</i>	<i>Función que representa los mapas de calor</i>
<i>draw_dbn.m</i>	<i>Función que representa la estructura de la RBD⁴</i>
<i>learn_cpt.m</i>	<i>Función que aprende las tablas de probabilidad de la RBD</i>
<i>learn_dbn_greedy.m</i>	<i>Función que aprende la estructura de la RBD</i>
<i>Private</i>	<i>Directorio con funciones auxiliares</i>
<i>quantize.m</i>	<i>Función que realiza la discretización de las variables continuas</i>
<i>roiNames.txt</i>	<i>Fichero con los nombres de cada ROI</i>

Tabla A6.7. Ficheros suministrados.

⁴ La función “draw_dbn.m” utiliza parte del código proporcionado en Bayes Net toolbox de Matlab (<http://code.google.com/p/bnt/>). Esta herramienta que incluye múltiples algoritmos de aprendizaje y de inferencia, resulta muy útil para familiarizarse con las técnicas de inferencia y aprendizaje de las RB.