

Diagnostico y análisis de salidas:

El gráfico de corrida y el estadístico de Gelman-Rubin (ver más adelante), pueden dar idea de cuantas iteraciones iniciales deben ser descartadas y si se ha alcanzado un número suficiente de iteraciones para lograr convergencia de la cadena. Las distribuciones posteriores univariadas se pueden resumir o bien de forma tabular o gráfica. Con el botón “coda” en el menú de las muestras se pueden exportar las muestras a archivos texto que pueden ser analizadas con otro paquete (como “coda”, que es una librería de R). Con la herramienta Compare del menú Inference se pueden comparar las distribuciones posteriores de un conjunto de parámetros.

WinBUGS también computa el valor del criterio de información de la devianza (o DIC), que es usado para comparar el ajuste de dos o más modelos en competencia. Este cálculo lo realiza evaluando la log-verosimilitud en cada iteración del muestreador, y calcula el promedio sobre las iteraciones y el valor en la media posterior.

Estadístico Gelman Rubin

En la ventana de seguimiento de muestras, (Figura 3) se presenta un botón nombrado `bgr diag`. Este presenta un gráfico del estadístico Gelman Rubin, siempre que se hayan generado al menos 3 cadenas. Este estadístico es el cociente entre la variancia conjunta (pooled) de las cadenas con respecto al promedio de las variancias dentro de cada cadena. Estos son dos estimados de la variancia posterior, que cuando se ha alcanzado convergencia deberían ser iguales, por lo que el valor del estadístico debe converger a 1. WinBUGS grafica tres series: en color rojo se presenta el estadístico; en color azul el promedio de la amplitud del intervalo de probabilidad del 80% dentro de cada cadena, y en verde la amplitud del intervalo 80% de considerando todas las cadenas (pooled).

GUIA RÁPIDA

WinBUGS

Preparada por Isabel Llatas a partir del artículo “Review of WinBUGS 1.4 (Mary Kathryn Cowles), disponible en

<http://dga.jouy.inra.fr/dga/internet/ESeminaire/CowlesWinbugs14Amstat2004.pdf>

WinBUGS es un software de uso general para ajustar modelos bayesianos complejos usando métodos Monte Carlo de Cadenas de Markov que ha facilitado las aplicaciones de métodos de inferencia bayesiana en la práctica.

WinBUGS permite al usuario especificar un modelo bayesiano de dos maneras, una dibujando un grafo dirigido y otra usando un lenguaje con características similares a R. En función de esto, el software determina el kernel de transición de una cadena de Markov y genera muestras de la distribución conjunta posterior de las cantidades desconocidas o de interés, especificando el número de cadenas paralelas que se correrán, el número de iteraciones, las cantidades desconocidas que se monitorearán y algunos chequeos de convergencia. Los resultados finales vienen dados en resúmenes tanto numéricos como gráficos de las distribuciones marginales univariadas.

Inicio y salida:

La versión educacional puede ser obtenida en: <http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents>.

Esta versión tiene ciertas limitaciones en el número de nodos que pueden definirse; el usuario se debe registrar en línea para obtener el archivo “llave” (key) que transforma la versión de prueba en una versión sin restricciones.

Para iniciar el programa se puede usar el icono de WinBUGS en el escritorio o correr desde el menú Inicio-Programas. Al iniciar, el programa presenta una pantalla con aspectos de la licencia de uso.

Para ‘correr’ un problema en WinBUGS hacen falta tres componentes o archivos, que son: “el modelo”, “los datos” y “los valores iniciales de las cadenas”. Cada uno de estos puede estar definido en archivos separados que son abiertos desde el menú File de la interfaz gráfica del programa o pueden mantenerse en el mismo archivo.

Ayudas:

En el menú Help de la interfaz gráfica se puede acceder al manual de usuario y a varios ejemplos que están a disposición. Igualmente hay una guía rápida para especificar un modelo usando la facilidad gráfica “Doodle”.

Especificación de modelos:

Si el modelo es sencillo e involucra distribuciones de probabilidad estándar, se puede especificar un modelo directamente con “Doodle”, en la ventana gráfica se crean tantos nodos como nombres de variables se tiene. Para crear un nodo, simplemente apriete el botón izquierdo del ratón en algún lugar de la ventana que aparece luego que se escoge “Doodle”-“New”. Los arcos dirigidos se obtienen escogiendo el nodo “hijo” y pulsando el botón izquierdo del ratón con la tecla <ctrl.> presionada sobre el nodo padre. Al ser seleccionado un nodo, aparece en la parte superior de la ventana dos propiedades del nodo: name, para que se escoja un nombre, que puede

ser indexado, por ejemplo $x[i]$ y `type`, que puede ser de tipo “estocástico”, “lógico” o “constante”. Si el tipo es estocástico, hay que definir una distribución (por defecto, la normal) y los parámetros iniciales de esta distribución; si es lógico se puede usar una función de vínculo (identidad, logit, probit, cloglog, log), y el valor o definición de la función. El modo constante se usa, por ejemplo, si se trata de un nodo de covariables. Para crear una placa de repetición (un loop for), se aprieta el botón izquierdo con la tecla <ctrl.> presionada. Una vez completado el modelo se escribe el código del modelo desde el menú `Doodle`.

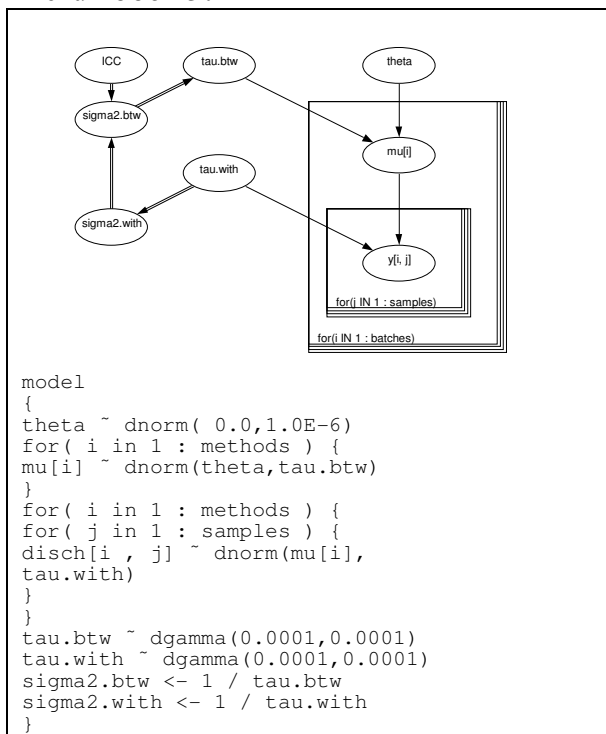


Figura 1: Grafo dirigido para el modelo de componentes de variancia

En la Figura 1 se presenta el grafo dirigido creado para un modelo de componentes de varianza y el código generado. Para definir

modelos más complejos lo que hay que recordar es que se inicia el modelo con la palabra `model`, las instrucciones van entre llaves y las asignaciones lógicas se hacen con “<-”, mientras que las asignaciones estocásticas son con “~”.

Datos y valores iniciales de las cadenas:

Los datos se pueden leer en dos formatos, uno tipo lista, como en R o S-Plus y otro en formato tipo tabla (filas y columnas). Las cantidades escalares deben ser definidas en el formato lista, por ejemplo para el modelo de componentes de varianza donde se tiene 4 métodos de medición y 6 muestras por método:

```

list(methods = 4, samples = 6,
  disch = structure(.Data=
c(0.583, 0.346, 1.109, 0.837, 1.323, 0.346,
0.954, 1.715, 1.463, 1.536, 1.691, 2.133,
2.512, 2.893, 3.122, 2.468, 3.134, 2.691,
4.141, 3.438, 3.309, 4.147, 3.788, 4.101),
.Dim = c(4,6)))

```

Para inicializar una cadena hay que proveer un valor inicial para cada parámetro del modelo en formato lista.

Compilando el modelo:

Esto se logra desde el menú `Model` con la herramienta de especificaciones que se presenta en la Figura 2.

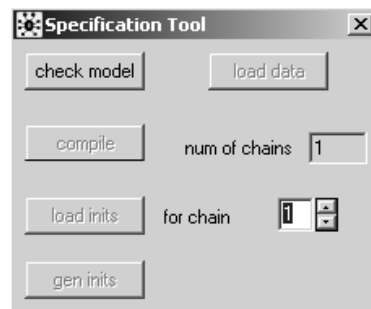


Figura 2: Menú para especificaciones del modelo- (a partir del menú `model`) Para chequear el modelo se escoge la palabra `model` con el ratón y se aprieta el botón

correspondiente. Para cargar los datos se escoge la palabra `list` y de nuevo se aprieta el botón correspondiente. En la esquina inferior izquierda aparecen mensajes si hay algún error. Una vez compilado y con los datos cargados se debe definir cuantas muestras generar, en el menú `Model-Update`.

Métodos de muestreo; seguimiento de cadenas:

WinBUGS usa el algoritmo de Gibbs para construir el kernel de transición de las cadenas de Markov. En la compilación, WinBUGS escoge el método de simulación para cada distribución condicional uniparamétrica completa, si la distribución es estándar se usan algoritmos conocidos, si no, se usa un método de muestreo de rechazo adaptativo, o el método de Metropolis.

En Winbugs hay que definir a cuales nodos se quiere hacer seguimiento, en el menú `Inference-Sample`, que se presenta en la Figura 3. Una vez definido el nodo, (con `set`), luego de generar una muestra de las cadenas, se puede revisar la traza, la densidad, la autocorrelación, en ventanas gráficas.

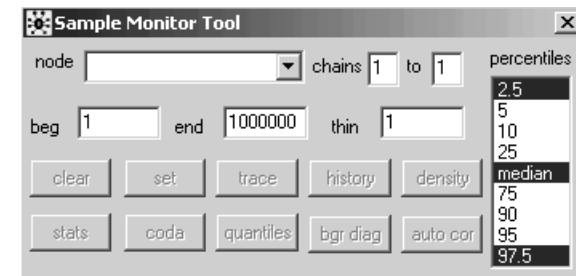


Figura 3: Menú para seguimiento de muestras de la distribución marginal posterior.