

Modelo de Red Bayesiana

Manual de Uso

Índice

1. Funcionalidades

1. Demostración prediseñada

2. Declaración de atributos del modelo

2.1. Declaración de Variables

2.2. Declaración de Valores

2.3. Declaración de Dependencia entre variables

2.4. Declaración de Hiper-Parámetro de Dirichlet

3. Importación de data externa

4. Observación de atributos del modelo

4.1. Observación de Cardinalidades

4.2. Observación del DAG

4.3. Observación de Data

5. Cálculos

5.1. Probabilidad Marginal

5.2. Probabilidad Conjunta

5.3. Probabilidad Condicional

5.4. Marginalización

5.5. Test de Independencia Condicional

5.5.1. Mutual Information

5.5.2. Pearson

5.6. Búsqueda de DAG óptimo

- 5.6.1. K2
- 5.6.2. Fuerza Bruta
- 5.6.3. Chow Liu
- 5.7. Medidas de Calidad
 - 5.7.1. K2
 - 5.7.2. MDL
 - 5.7.3. AIC
- 5.8. Inferencias
- 5.9. Entrenar

1. Introducción

El programa propuesto consiste en una línea de comandos para modelar redes bayesianas, permite diversas funcionalidades como asignar variables, leer archivos en formato csv, buscar el DAG optimo por algoritmos K2 y de Fuerza Bruta, etc.

Se tiene que ejecutar el programa mediante la linea de comandos. Se tiene que ir hasta la carpeta que contenga el programa y los datasets y ejecutar el comando “./output”

2. Funcionalidades

2.1. Demostración prediseñada

Se puede acceder a una prueba prediseñada de los comandos usando data que se incluye en los archivos “DemoData.csv”, “DemoData2.csv”, “DemoData3.csv” y “Asia.csv”.

Comando: “demo”

Input: demo1

Input: demo2

Input: demo3

Input: demo4

2.2. Declaración de atributos del modelo

2.2.1. Declaración de Variables

El programa propuesto permite al usuario declarar manualmente las variables que modelo mediante el comando “vars” seguido de los nombres de cada variable.

Comando: “vars”

Ejemplo: Declaración de variables A, B, C, D

Input: vars A B C D

Output:

+var(0): A

+var(1): B

+var(2): C

+var(3): D

2.2.2. Declaración de Valores

El programa propuesto permite al usuario declarar manualmente los valores de cada variable. Se requiere declara la variable antes de declara sus valores.

Comando: “vals”

Ejemplo: Declaración de valores para la variable A

Input: vals A Alto Medio Bajo

Output:

+val(0): Alto

+val(1): Medio

+val(2): Bajo

Ejemplo: Declaracion de valores para variable A(Discreta) y variable B(Continua).

Input: vals B -c

Output:

2.2.3. Declaración de Dependencia entre variables

El programa propuesto permite al usuario declarar manualmente dependencia entre variables. El primer elemento será reconocido como padre y aquellos elementos que le sigan serán reconocidos como hijos.

Comando: “arco”

Ejemplo: Declaración de arco de variable A hacia B y C.

Input: arco A B C

Output:

Arco creado: A->B

Arco creado: A->C

2.2.4. Declaración de Hiper-Parámetro de Dirichlet

El programa propuesto permite al usuario declarar manualmente el valor del hiper-parámetro de Dirichlet. Se puede declarar el valor de alpha como un porcentaje del tamaño total del dataset mediante el comando “-p”. De declarado manualmente, se asume el valor cero.

Comando: “alpha”

Ejemplo: Declaración de valor de alpha igual a 1

Input: alpha 1

Output:

+alpha: 1

Ejemplo: Declaración de valor de alpha igual al quince por ciento del tamaño total del dataset de 10 instancias.

Input: alpha -p 0.15

Output:

+alpha: 1.5

2.3. Importación de data externa

El programa propuesto permite al usuario importar un archivo “csv” que contenga la data a usar para el entrenamiento del modelo. Se puede usar el parámetro “-c” para declarar que el archivo contiene valores de cabecera que representan los nombres de las variables. Se requiere declara las variables y valores de variables antes de importar data.

Comando: “data”

Ejemplo: Importe de dataset sin cabecera y separado por comas

Input: data C://

Output:

+Cabecera: No

+Variables: 4

+Instancias: 100

Ejemplo: Importe de dataset con cabecera y separado por comas

Input: data C:// -c

Output:

+Cabecera: Si

+Variables: 4

+Instancias: 100

2.4. Observación de atributos del modelo

2.4.1. Observación de Cardinalidades

El programa propuesto permite al usuario observar las cardinalidades de cada variable.

Comando: “cards”

Ejemplo: Observación de cardinalidades del modelo con variables A, B y C cuyas cardinalidades son 2, 3 y 2

Input: cards

Output:

+A: 2

+B: 3

+C: 2

2.4.2. Observación del DAG

El programa propuesto permite al usuario observar el grafo de dependencias o DAG, por sus siglas en ingles, en forma de matriz.

Comando: “dag”

Ejemplo: Observación del DAG= $\{A \rightarrow B, A \rightarrow C\}$.

Input: dag

Output: A B C

0 0 0

1 0 0

1 0 0

2.5. Cálculos

2.5.1. Probabilidad Marginal

El programa propuesto permite al usuario calcular la probabilidad marginal de una variable. Se puede usar el parámetro “-v” para declara el valor de la variable.

Comando: “marginal”

Ejemplo: Calculo de probabilidad marginal de variable A

Input: marginal A

Output: A

+SI: 0.45

+NO: 0.55

Ejemplo: Calculo de probabilidad marginal de variable A

Input: marginal A -v SI

Output: A

+SI: 0.45

2.5.2. Probabilidad Conjunta

El programa propuesto permite al usuario calcular la probabilidad conjunta de un grupo de variables.

Comando: “conjunta”

Ejemplo: Calculo de probabilidad conjunta de variables A, B y C

Input: conjunta A B C

Output: A B C

SI BAJO SI: 0.083333333333

SI BAJO NO: 0.083333333333

SI MEDIO SI: 0.083333333333

SI MEDIO NO: 0.083333333333

SI ALTO SI: 0.083333333333

SI ALTO NO: 0.083333333333

NO BAJO SI: 0.083333333333

NO BAJO NO: 0. 083333333333
NO MEDIO SI: 0. 083333333333
NO MEDIO NO: 0. 083333333333
NO ALTO SI: 0. 083333333333
NO ALTO NO: 0. 083333333333
Distribución: 1

2.5.3. Probabilidad Condicional

El programa propuesto permite al usuario calcular la probabilidad condicional de un grupo de variables dado un grupo de padres.

Comando: “condicional”

Ejemplo: Calculo de probabilidad condicional de variables A, B con padre C

Input: condicional A B -p C

Output: A | C

SI SI: 0.25
SI NO: 0. 15
NO SI: 0. 20
NO NO: 0. 50
Distribucion: 2

2.5.4. Marginalización

El programa propuesto permite al usuario seleccionar un conjunto de variables para ser marginalizadas.

Comando: “marginalizacion”

Ejemplo: Marginalizar A dado el conjunto de variables A, B y C

Input: conjunta A B C

Output: A C B

0 0 0 0.0416667

0 0 1 0.0833333

0 0 2 0.0416667

Marginalizacion de B para A = 0 C = 0 Resultado: 0.166667

A C B

0 1 0 0.0833333

0 1 1 0.0416667

0 1 2 0.0833333

Marginalizacion de B para A = 0 C = 1 Resultado: 0.208333

A C B

1 0 0 0.0833333

1 0 1 0.0416667

1 0 2 0.0833333

Marginalizacion de B para A = 1 C = 0 Resultado: 0.208333

A C B

1 1 0 0.0416667

1 1 1 0.0416667

1 1 2 0.0416667

Marginalizacion de B para A = 1 C = 1 Resultado: 0.125

A C B

2 0 0 0.0416667

2 0 1 0.0416667

2 0 2 0.0416667

Marginalizacion de B para $A = 2$ $C = 0$ Resultado: 0.125

A C B

2 1 0 0.0416667

2 1 1 0.0833333

2 1 2 0.0416667

Marginalizacion de B para $A = 2$ $C = 1$ Resultado: 0.166667

2.5.5. Test de Independencia Condicional

2.5.5.1. Mutual Information

El programa propuesto permite al usuario realizar el test de independencia condicional Mutual Information

Comando: “mut”

Ejemplo: Calcular test de independencia condicional MI de A con B dado C

Input: mut A B -p C

Output: Grado de independencia MI: 6

2.5.5.2. Pearson

El programa propuesto permite al usuario realizar el test de independencia condicional Pearson

Comando: “pearson”

Ejemplo: Calcular test de independencia condicional Pearson de A con B dado C

Input: pearson A B -p C

Output: Grado de independencia Pearson: 3.68

2.5.6. Búsqueda de DAG óptimo

2.5.6.1. K2

El programa propuesto permite al usuario realizar una búsqueda del mejor DAG por medio del algoritmo K2 con el comando “k2”, el uso de las banderas es necesario(Buscar abajo en métricas de calidad).

Comando: “k2”

Ejemplo: Calcular el mejor DAG por K2

Input: k2

Output: Actual Puntaje: 102335

```
0 1 0 0
0 0 0 0
0 1 0 1
0 0 0 0
```

2.5.6.2. Fuerza Bruta

El programa permite al usuario realizar una búsqueda del mejor DAG por medio del algoritmo Fuerza Bruta “fuerzabruta”, el uso de las banderas es necesario(Buscar abajo en métricas de calidad).

Comando: “fuerzabruta”

Ejemplo: Calcular el mejor DAG por fuerzabruta

Input: fuerzabruta

Output: Mejor Puntaje: 102335

```
0 1 0 0
0 0 0 0
0 1 0 1
0 0 0 0
```

2.5.6.3. Chow Liu

El programa permite al usuario generar un el grafo del modelo por el algoritmo Chow Liu

Comando: “chowliu”

Ejemplo: Calcular el mejor DAG por Chow Liu

Input: chowliu

Output:

0 1 0 0
0 0 0 0
0 1 0 1
0 0 0 0

2.5.7. Medidas de Calidad

2.5.7.1. K2

El algoritmo K2 y Fuerza Bruta acepta la medida de calidad K2 por medio de la bandera “-k2”

2.5.7.2. MDL

El algoritmo K2 y Fuerza Bruta acepta la medida de calidad MDL por medio de la bandera “-mdl”

2.5.7.3. AIC

El algoritmo K2 y Fuerza Bruta acepta la medida de calidad AIC por medio de la bandera “-aic”

2.5.8. Inferencias

El programa permite realizar inferencias por medio de Variable Elimination

Comando: “inferencia”

Ejemplo: Inferir T dado B = yes, A = no, S = no y X = yes

Input: inferencia T B A S X -v yes no no yes

Output:

$P(T|A=no,)P(B=yes|S=no,)P(A=no)P(S=no|L,)P(X=yes|E,)P(L|E,)P(E|T,)$

Marginalizando: L

Marginalizando: E

Las probabilidades son:

Para T = yes -> 0.0396246

Para T = no -> 0.960375

2.5.9. Entrenar

El programa permite realizar inferencias por medio de Variable Elimination

Comando: “entrenar”

Ejemplo: Entrenar el modelo en base al DAG actual

Input: entrenar

Output:

$P(A)$

yes -> 0.00859656

no -> 0.991403

$P(S|L)$

yes yes -> 0.894578

yes no -> 0.475171

no yes -> 0.105422

no no -> 0.524829

$P(T|A)$

yes yes -> 0.0681818

yes no -> 0.00866935

no yes -> 0.931818

no no -> 0.991331

$P(L|E)$

yes yes -> 0.889785

yes no -> 0.000215889

no yes -> 0.110215

no no -> 0.999784

$P(B|S)$

yes yes -> 0.717521

yes no -> 0.299558

no yes -> 0.282479

no no -> 0.700442

$P(E|T)$

yes yes -> 0.978261

yes no -> 0.065954

no yes -> 0.0217391

no no -> 0.934046

$P(X|E)$

yes yes -> 0.991935

yes no -> 0.0436097

no yes -> 0.00806452

no no -> 0.95639

$P(D|B)$

yes yes -> 0.792238

yes no -> 0.134937

no yes -> 0.207762

no no -> 0.865063

Modelo Entrenado