



INSTITUTO SUPERIOR TÉCNICO
MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE
COMPUTADORES

PROGRAMAÇÃO ORIENTADA A OBJECTOS
Aprendizagem de redes dinâmicas de Bayes

Maria Margarida Dias dos Reis	n.º 73099
Ricardo Filipe Amendoeira	n.º 73373
David Romão Fialho	n.º 73530

Lisboa, 18 de Maio de 2015

Índice

1	Introdução	1
2	Decisões de Projecto	1
3	Estrutura do Projecto	1
4	Detalhes de Implementação	1
5	Testes Efectuados	1
6	Conclusão	4

1 Introdução

Com este projecto pretende-se utilizar redes dinâmicas de Bayes (DBN) para modelar uma série multivariante no tempo.

continuar
isto

2 Decisões de Projecto

Para se definir a estrutura do projecto optou-se, à partida, por tentar implementar uma estrutura modular e reutilizável, ou seja, algo que funcione não apenas com o que se pretende elaborar e com os requisitos a cumprir mas sim para casos genéricos. Assim, as *features* que foram projectadas tendo como base uma *framework* reutilizável são:

- *Bayesian Network* (BN) e *Dinamic Bayesian Network* (DBN);
- grafo;
- operações;
- *score*;
- critério de paragem do algoritmo GHC;
- número de pais de um dado nó.

É de referir que, apesar da rapidez de computação ser um critério relativamente importante para um programa deste género, decidiu-se que ter uma solução que providencia uma *framework* extensível e reutilizável para a aprendizagem de DBNs é ainda mais importante, quando se considera o âmbito da cadeira na qual o projecto está inserido.

3 Estrutura do Projecto

4 Detalhes de Implementação

explicar o
que e inter-
face, class,
etc

5 Testes Efectuados

explicar
como se faz
com as con-
tagens, etc

5.1 Aprendizagem da DBN

5.2 *Dataset*

explicar est
teste

5.2.1 Criação de DAGs

Ao longo de toda a execução do programa é importante garantir que o grafo que representa a BN é um grafo acíclico. Para testar a implementação desta verificação criou-se um grafo inicial com três nós não ligados e, de seguida, efectuaram-se três operações de adição de arestas que resultariam num grafo com ciclos.

As duas primeiras operações são realizadas com sucesso, mas a terceira, que iria gerar um ciclo nesse grafo, não chega a ser concluída, voltando o grafo ao seu estado de duas arestas apenas.

5.2.2 Mapeamento $j \leftrightarrow J$

Outro aspecto importante da aprendizagem da DBN é o mapeamento de $j \leftrightarrow J$. Começou-se por testar esta operação isoladamente, ou seja, inicializou-se um *array* correspondente aos *ranges* dos pais, outro com os valores dos pais e outro ainda com o valor de J que esses valores originariam. Uma vez verificado o correcto funcionamento do mapeamento este foi integrado no código do programa.

Face aos resultados obtidos para os testes explicados anteriormente assume-se que a aprendizagem da DBN está a funcionar.

5.3 Inferências

Inicialmente, para se verificar o funcionamento das inferências optou-se por verificar que a soma das probabilidades obtidas para os valores futuros possíveis da variável aleatória que se está a inferir dá próximo de 1, isto porque, obrigatoriamente, a variável aleatória toma no futuro um valor dos possíveis do seu *range*.

Este teste foi feito com recurso ao grafo da Figura 1, cuja criação se forçou na execução do programa.

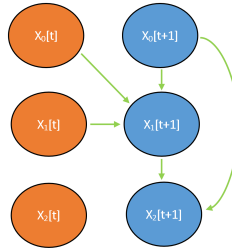


Figura 1: Grafo da rede de transição utilizada para testar as inferências.

Assumindo todas as variáveis aleatórias como binárias, os valores obtidos para as probabilidades das três variáveis do futuro para os seus dois valores possíveis apresentam-se na seguinte tabela.

Tabela 1: Probabilidades obtidas para os valores das variáveis aleatórias no futuro.

	$X_0[t+1]$	$X_1[t+1]$	$X_2[t+1]$
0	0.9999999999999949	0.9999999999999988	0.9999999999999982
1	1.000000000000005	0.9999999999999946	0.9999999999999969

Como se pode ver, todas as probabilidades têm um valor muito próximo de 1, sendo normal que algumas sejam de facto maiores, uma vez que são calculadas com recurso a estimativas.

Para verificar as inferências que o programa obtém optou-se por usar o teste #2 fornecido na página da cadeira como ficheiro de treino para se efectuar a aprendizagem da rede de Bayes. O ficheiro de teste construído corresponde aos valores do instante de tempo $t = 0$ do ficheiro de treino, sendo que, algumas das inferências, devem tomar valores próximos dos do instante de tempo $t = 1$ do ficheiro de treino:

```

1 A, B, C, D, E, F, G
2 3, 3, 1, 0, 1, 1, 2
3 2, 3, 0, 3, 3, 1, 0
4 3, 1, 0, 1, 1, 2, 3

```

Considerando a construção da rede de Bayes sem *random restarts* aquando da aplicação do algoritmo GHC tem-se a seguinte execução de programa para a inferência de todas as variáveis aleatórias. De notar que o teste apresentado de seguida foi executado no computador de um dos membros do grupo.

```

1 Parameters: train-data-2.csv test-data-2-TRAIN.csv LL 0
2 Building DBN: 28.934149246 seconds
3 Initial network:
4 === Structure connectivity
5 A : C

```

```

6 B :
7 C : D
8 D : G
9 E : A
10 F : A
11 G :
12 === Scores
13 LL Score: -4.754887502163469
14 MDL Score: -131.55188755985597
15 Transition network:
16 === Inter-slice connectivity
17 A : F G
18 B : F G
19 C : F
20 D : C F G
21 E : F
22 F :
23 G :
24 === Intra-slice connectivity
25 A : C
26 B : D
27 C : E D
28 D :
29 E : B D
30 F : E D G
31 G : A B E
32 === Scores
33 LL Score: -4233.21662434474
34 MDL Score: -33449.233345448425
35 Performing inference:
36 -> instance 1: 3, 3, 1, 0, 1, 1, 0
37 -> instance 2: 2, 4, 0, 3, 1, 1, 0
38 -> instance 3: 3, 2, 0, 2, 0, 1, 0
39 Inferring with DBN: 64.80672281400001 seconds

```

Analisando os valores pretendidos e aqueles que foram de facto obtidos verifica-se uma eficácia de 61.91%, um valor que se considera aceitável e, tendo em conta, que as somas das probabilidades já foram verificadas como estando próximas de 1 assume-se que as inferências estão correctas.

5.4 Performance Geral

Relembrando o teste anterior que analisa as inferências com base no ficheiro de treino, efectuaram-se repetições desse teste ainda no mesmo computador mas com a inclusão de *random restarts*.

De seguida apresenta-se o teste com o *score* LL, com 10 *random restarts* e com lista TABU.

```

1 Parameters: train-data-2.csv test-data-2-TRAIN.csv LL 10
2 Building DBN: 84.131070823 seconds
3 Initial network:
4 === Structure connectivity
5 A : C
6 B : D E A
7 C : F
8 D : C F A
9 E : C
10 F :
11 G : B F A
12 === Scores
13 LL Score: -4.754887502163468
14 MDL Score: -2647.679857454692
15 Transition network:
16 === Inter-slice connectivity
17 A : F G
18 B : F G
19 C : F
20 D : C F G
21 E : F
22 F :
23 G :
24 === Intra-slice connectivity
25 A : C
26 B : D
27 C : E D
28 D :
29 E : B D
30 F : E D G
31 G : A B E
32 === Scores
33 LL Score: -4233.21662434474
34 MDL Score: -33449.233345448425

```

```

35 Performing inference:
36 -> instance 1: 3, 3, 1, 0, 1, 1, 0
37 -> instance 2: 2, 4, 0, 3, 1, 1, 0
38 -> instance 3: 3, 2, 0, 2, 0, 1, 0
39 Inferring with DBN: 54.103460464 seconds

```

O teste com 50 *random restarts* apresenta-se em baixo.

```

1 Parameters: train-data-2.csv test-data-2-TRAIN.csv LL 50
2 Building DBN: 351.122175179 seconds
3 Transition network:
4 === Inter-slice connectivity
5 A : F G
6 B : F G
7 C : F
8 D : C F G
9 E : F
10 F :
11 G :
12 === Intra-slice connectivity
13 A : C
14 B : D
15 C : E D
16 D :
17 E : B D
18 F : E D G
19 G : A B E
20 === Scores
21 LL Score: -4233.21662434474
22 MDL Score: -33449.233345448425
23 Performing inference:
24 -> instance 1: 3, 3, 1, 0, 1, 1, 0
25 -> instance 2: 2, 4, 0, 3, 1, 1, 0
26 -> instance 3: 3, 2, 0, 2, 0, 1, 0
27 Inferring with DBN: 51.038501682 seconds

```

Como se pode ver, quantos mais *random restarts* houver, mais tempo se demora na aprendizagem da BN, tal como expectável. No entanto, verifica-se que o *score* obtido para a rede de transição não mudou, isto porque para se notarem diferenças é necessário ter pelo menos 500 ou 1000 *random restarts*.

Assim, optou-se por gerar uma série de teste com valores diferentes de *random restarts*, gerando gráficos dos *scores* obtidos.

6 Conclusão