Aprendizaje Automático

Modelos Gráficos Probabilísticos

Ramon Ruiz Dolz
Aitor Signes Cuco
4CO21-2017

Ramon Ruiz Dolz
Aitor Signes Cuco

1. Introducción

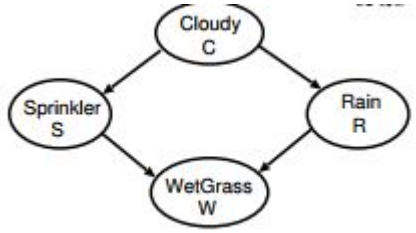
En esta práctica hemos aplicado los conceptos aprendidos en la teoría relativos a las redes bayesianas o BNT. A lo largo de la práctica hemos trabajado con distintas redes para poder observar mejor los conceptos teóricos aplicados en MATLAB. Nuestra experimentación se ha dividido en dos bloques principales, el primero en el cual hemos trabajado con datos completo e incompletos para aprender probabilidades de la BNT, y el segundo donde hemos partido del dataset de SPAM para aprender y clasificar las muestras mediante una BNT con mixturas de gaussianas.

2. Aprendizaje con datos completos e incompletos.

En este apartado hemos aprendido cómo funcionan las BNT en el ámbito del aprendizaje. Para poner en práctica estos conceptos hemos realizado dos experimentos, uno con una red bayesiana con las probabilidades de que el suelo esté mojado por un aspersor o por la lluvia, y otro con otra red que contiene las probabilidades de que un paciente tenga cáncer de pulmón en función de varios parámetros como que sea fumador, la polución en el ambiente, el resultado de los rayos x y si sufre de disnea.

- Sprinkler:

La red bayesiana que representa este set de datos es la siguiente:



Para realizar este primer experimento hemos lanzado el código modificando el parámetro *nmuestras*, es decir el número de muestras para el aprendizaje. En la tabla inferior tenemos los datos obtenidos:

nmuestras = 1000;
maxIter = 100;
#iter necesarias 25
 	
Cloudy
1 : 0.9996
2 : 0.0004
 	
Sprinkler
1 : 0.6969 0.3031
2 : 1.0000 0.0000
	nmuestras = 100;
	maxIter = 100;
	#iter necesarias 20
	
	Cloudy
	1 : 1.0000
	2 : 0.0000
	
	Sprinkler
	1 : 0.6676 0.3324
	2 : 0.9839 0.0161

```
Rain
1 : 0.4247 0.5753
2 : 0.0000 1.0000

Wet grass
1 1 : 0.9988 0.0012
2 1 : 0.2952 0.7048
1 2 : 0.1496 0.8504
2 2 : 0.0601 0.9399
```

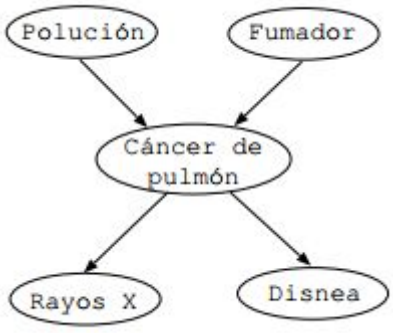
```
Rain
1 : 0.4096 0.5904
2 : 0.0000 1.0000

Wet grass
1 1 : 0.9996 0.0004
2 1 : 0.0340 0.9660
1 2 : 0.0071 0.9929
2 2 : 0.1329 0.8671
```

Como podemos observar, los valores probabilísticos cambian, no son igual con 100 muestras de aprendizaje que con 1000. Tampoco son necesarias el mismo número de iteraciones hasta obtener una solución como se puede observar, con 100 muestras con 20 iteraciones ya se ha alcanzado el resultado mientras que con 1000 son necesarias hasta 25 iteraciones. Pese a no ser enormes, se pueden apreciar cambios relevantes en las probabilidades, por ejemplo, la probabilidad de que el cielo esté nublado es 0 con 100 muestras mientras que con 1000 muestras existe probabilidad de que este y que no esté nublado aunque la probabilidad de la primera se acerque mucho a 0.

- Cáncer de pulmón:

La red bayesiana que representa este set de datos es la siguiente:



En este segundo ejercicio hemos realizado los experimentos entorno a esta red bayesiana. Primero hemos obtenido la probabilidad de que el paciente no tenga cáncer de pulmón si la radiografía ha dado un resultado negativo pero sufre disnea, es decir partiendo de estos dos datos observados, se ha modelado el script para obtener el valor de cáncer obteniendo como salida:

```
ans =

    0.0011
    0.9989
```

Siendo el primer valor la probabilidad de que SI tenga cáncer de pulmón y el segundo valor el de que NO. Por lo tanto la respuesta a la pregunta del ejercicio es que la probabilidad de que el paciente no sufra cáncer de pulmón sabiendo que la radiografía ha dado resultado negativo pero sufre disnea es de 0’9989, es decir una probabilidad muy elevada así que el paciente puede estar más o menos tranquilo por el momento.

Por otra parte, se nos pide obtener la explicación más probable de que un paciente sufra cáncer de pulmón. Esto quiere decir si C = 1, cuanto valdran las demás variables? Tenemos en cuenta que Polución(P) 1 = alto, 2 = bajo; Fumador(F) 1 = sí, 2 = no; Disnea(D) 1 = sí, 2 = no; Rayos X(R) 1 = positivo, 2 = dudoso, 3 = negativo; y Cáncer(C) 1 = positivo, 2 = negativo.
Tras lanzar nuestro script en MATLAB obtenemos la siguiente salida:

```
explMasProb =

  1×5 cell array

    [2]    [1]    [1]    [1]    [1]
       P             F             C             R             D

logVer =

   -5.0925
```

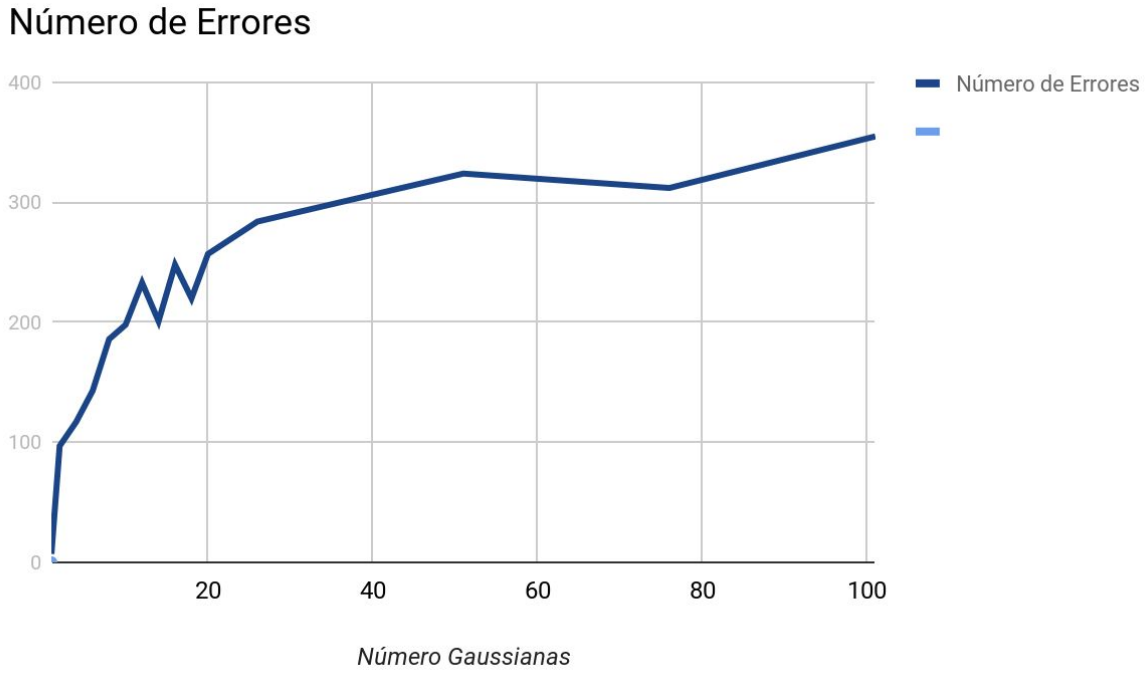
Estos datos se interpretan como que la explicación más probable de que un paciente sufra cáncer son que la polución sea baja, que sea fumador, que haya dado positivo en la prueba de rayos X y que sufra de disnea. Observando estas conclusiones de primeras puede parecer que no cuadre la primera, por qué la polución baja y no alta? Bien, esto se debe a que la probabilidad independiente de que la polución sea alta es tan baja que como explicación más probable prevalece la probabilidad de que la polución sea baja.

3. Aprendizaje y clasificación con mixtura de gaussianas.

En esta última parte, hemos entrenado nuestra BNT para clasificar un dataset de SPAM/ NO SPAM mediante es uso de mixtura de gaussianas. Grácias a las herramientas provistas en matlab, se ha aprendido la red y posteriormente se han clasificado los datos, obteniendo un error y su intervalos de confianza para cada prueba con cada número de gaussianas.

Para la obtención del error, una vez convergido el modelo con un número de gaussianas determinado se ha utilizado una variable error a modo de contador que se incrementa cada vez que la clase real de la muestra no coincide con la de mayor probabilidad a posteriori obtenida en el entrenamiento. Con el dataset de 1381 muestras se han obtenido los siguientes resultados:

Número de gaussianas	Num Errores	Error (%)	Intervalo de confianza
1	7	0.5%	+- 0.0037
2	97	7%	+-0.0000 + 5.0896i
4	117	8.4%	+-0.0000 + 6.1444i
6	143	10.4%	+-0.0000 + 7.5157i
8	186	13.4%	+-0.0000 + 9.7837i
10	198	14.3%	+-0.0000 +10.4166i
12	233	16.8%	+-0.0000 +12.2626i
14	201	14.5%	+-0.0000 +10.5748i
16	248	17.9%	+-0.0000 +13.0537i
18	220	15.9%	+-0.0000 +11.5769i
20	257	18.6%	+-0.0000 +13.5284i
26	284	20.5%	+-0.0037
51	324	23.4%	+-0.0213
76	312	22.5%	+-0.0221
101	355	25.7%	+-0.0230



En esta gráfica queda representado de forma visual los datos expuestos en la tabla anterior. Dibujando el número de errores total en función del número de gaussianas usadas para el entrenamiento, obteniendo el menor número de errores para la ejecución con 1 gaussiana. En el apéndice podemos observar el output de todos los experimentos realizados.

4. Apéndice

Output:

```
De 1 a 101 de 25 en 25

numVec =

        3220


dim =

    58


numClas =

     2


numGaus =

     1


numNodos =

     3

EM iteration 1, ll = -608876.6278
EM iteration 2, ll = -187136.9020
EM iteration 3, ll = -185876.4748
EM iteration 4, ll = -185876.4748


error =

    0.0051 (7)


ans =

        1381


confianza =

    0.0037


numGaus =

    26


numNodos =

     3

EM iteration 1, ll = -606928.3825
EM iteration 2, ll = -176101.0972
```

```
EM iteration 3, ll = -60191.9139
EM iteration 4, ll = 3404.9238
EM iteration 5, ll = 17758.9334
EM iteration 6, ll = 22536.5757
EM iteration 7, ll = 25092.6099
EM iteration 8, ll = 26810.3371
EM iteration 9, ll = 27723.9256
EM iteration 10, ll = 28159.6423
EM iteration 11, ll = 28457.7308
EM iteration 12, ll = 28078.6073
******likelihood decreased from 28457.7308 to 28078.6073!
EM iteration 13, ll = 28959.4389
EM iteration 14, ll = 29165.9648
EM iteration 15, ll = 29544.9348
EM iteration 16, ll = 29784.6383


error =

    0.2056 (284)


ans =

        1381


confianza =

    0.0213


numGaus =

    51


numNodos =

     3

EM iteration 1, ll = -606182.9708
EM iteration 2, ll = -174769.8127
EM iteration 3, ll = -55283.6010
EM iteration 4, ll = 21148.8045
EM iteration 5, ll = 39337.5898
EM iteration 6, ll = 43952.6542
EM iteration 7, ll = 46372.5923
EM iteration 8, ll = 47524.0365
EM iteration 9, ll = 48048.9288
EM iteration 10, ll = 48465.9121
EM iteration 11, ll = 48607.9408
EM iteration 12, ll = 48707.5045
EM iteration 13, ll = 48796.8124
EM iteration 14, ll = 48912.9032
EM iteration 15, ll = 48974.6255
EM iteration 16, ll = 49149.2668


error =

    0.2346 (324)


ans =
```

```
        1381


confianza =

    0.0223


numGaus =

    76


numNodos =

     3

EM iteration 1, ll = -606219.9122
EM iteration 2, ll = -173270.8049
EM iteration 3, ll = -50044.1183
EM iteration 4, ll = 30198.4330
EM iteration 5, ll = 49048.1803
EM iteration 6, ll = 52384.8127
EM iteration 7, ll = 53380.5339
EM iteration 8, ll = 53709.0087
EM iteration 9, ll = 53862.9410
EM iteration 10, ll = 53943.2002
EM iteration 11, ll = 54103.4389
EM iteration 12, ll = 54103.6181


error =

    0.2259 (312)


ans =

        1381


confianza =

    0.0221


numGaus =

   101


numNodos =

     3

EM iteration 1, ll = -606277.4163
EM iteration 2, ll = -174423.3151
EM iteration 3, ll = -44568.9268
EM iteration 4, ll = 36179.2602
EM iteration 5, ll = 55718.7840
EM iteration 6, ll = 58581.1432
EM iteration 7, ll = 59203.7173
EM iteration 8, ll = 59560.5319
EM iteration 9, ll = 59630.7287
```

```
EM iteration 10, ll = 59656.8321


error =

    0.2571 (355)


ans =

        1381


confianza =

    0.0230


------------------------------------------------
De 2 a 20 de 2 en 2

numVec =

        3220


dim =

    58


numClas =

     2


numGaus =

     2


numNodos =

     3

EM iteration 1, ll = -605892.6181
EM iteration 2, ll = -182788.6801
EM iteration 3, ll = -144348.1631
EM iteration 4, ll = -124713.9372
EM iteration 5, ll = -121314.6438
EM iteration 6, ll = -120274.9923
EM iteration 7, ll = -119356.6251
EM iteration 8, ll = -118618.1891
EM iteration 9, ll = -118368.2881
EM iteration 10, ll = -118261.7191


error =

    97


ans =

        1381
```

```
confianza =

   0.0000 + 5.0896i


numGaus =

     4


numNodos =

     3

EM iteration 1, ll = -606514.5091
EM iteration 2, ll = -180826.8167
EM iteration 3, ll = -116519.7539
EM iteration 4, ll = -86300.4549
EM iteration 5, ll = -78067.1747
EM iteration 6, ll = -73695.6603
EM iteration 7, ll = -69220.4917
EM iteration 8, ll = -65869.0995
EM iteration 9, ll = -63293.8668
EM iteration 10, ll = -61809.8661
EM iteration 11, ll = -60536.3899
EM iteration 12, ll = -59947.8182
EM iteration 13, ll = -59548.8363
EM iteration 14, ll = -59438.8908
EM iteration 15, ll = -59379.3812
EM iteration 16, ll = -59366.8652


error =

   117


ans =

        1381


confianza =

   0.0000 + 6.1444i


numGaus =

     6


numNodos =

     3

EM iteration 1, ll = -605718.6397
EM iteration 2, ll = -176735.5491
EM iteration 3, ll = -92790.1099
EM iteration 4, ll = -59619.2678
EM iteration 5, ll = -51056.8255
EM iteration 6, ll = -47605.9836
EM iteration 7, ll = -45514.2528
```

```
EM iteration 8, ll = -43634.6010
EM iteration 9, ll = -42473.9549
EM iteration 10, ll = -41494.2714
EM iteration 11, ll = -41006.0483
EM iteration 12, ll = -40664.2808
EM iteration 13, ll = -40049.6237
EM iteration 14, ll = -39836.3678
EM iteration 15, ll = -39467.1621
EM iteration 16, ll = -39301.6750


error =

   143


ans =

        1381


confianza =

   0.0000 + 7.5157i


numGaus =

     8


numNodos =

     3

EM iteration 1, ll = -605942.4133
EM iteration 2, ll = -176840.5418
EM iteration 3, ll = -82048.8382
EM iteration 4, ll = -46299.5469
EM iteration 5, ll = -34403.3601
EM iteration 6, ll = -29008.8705
EM iteration 7, ll = -25318.2912
EM iteration 8, ll = -22538.6246
EM iteration 9, ll = -20906.4908
EM iteration 10, ll = -19233.5289
EM iteration 11, ll = -17604.5067
EM iteration 12, ll = -16419.7923
EM iteration 13, ll = -15761.4266
EM iteration 14, ll = -15545.5967
EM iteration 15, ll = -15431.6369
EM iteration 16, ll = -15280.1445


error =

   186


ans =

        1381


confianza =
```

```
   0.0000 + 9.7837i


numGaus =

    10


numNodos =

     3

EM iteration 1, ll = -606159.7056
EM iteration 2, ll = -178800.0291
EM iteration 3, ll = -82212.0555
EM iteration 4, ll = -30568.8379
EM iteration 5, ll = -17797.0068
EM iteration 6, ll = -13711.3248
EM iteration 7, ll = -11201.6804
EM iteration 8, ll = -9815.6824
EM iteration 9, ll = -9091.6275
EM iteration 10, ll = -8602.5431
EM iteration 11, ll = -8108.0804
EM iteration 12, ll = -7468.2707
EM iteration 13, ll = -7008.3817
EM iteration 14, ll = -6843.3977
EM iteration 15, ll = -6476.9976
EM iteration 16, ll = -6302.2007


error =

   198


ans =

        1381


confianza =

   0.0000 +10.4166i


numGaus =

    12


numNodos =

     3

EM iteration 1, ll = -605676.1651
EM iteration 2, ll = -177563.1146
EM iteration 3, ll = -74199.6486
EM iteration 4, ll = -20128.5046
EM iteration 5, ll = -10252.2448
EM iteration 6, ll = -7093.2894
EM iteration 7, ll = -5067.1028
EM iteration 8, ll = -4289.1218
EM iteration 9, ll = -3758.3481
EM iteration 10, ll = -3125.7149
EM iteration 11, ll = -2673.8656
```

```
EM iteration 12, ll = -2043.2240
EM iteration 13, ll = -1440.5612
EM iteration 14, ll = -828.2807
EM iteration 15, ll = -477.0857
EM iteration 16, ll = -323.5320


error =

   233


ans =

        1381


confianza =

   0.0000 +12.2626i


numGaus =

    14


numNodos =

     3

EM iteration 1, ll = -605807.8633
EM iteration 2, ll = -176393.6791
EM iteration 3, ll = -72146.8763
EM iteration 4, ll = -18708.7642
EM iteration 5, ll = -5725.1008
EM iteration 6, ll = -1133.8440
EM iteration 7, ll = 1986.6496
EM iteration 8, ll = 3425.3245
EM iteration 9, ll = 4031.5903
EM iteration 10, ll = 4648.9200
EM iteration 11, ll = 5115.8122
EM iteration 12, ll = 5469.6366
EM iteration 13, ll = 5676.7056
EM iteration 14, ll = 5951.6393
EM iteration 15, ll = 6255.9800
EM iteration 16, ll = 6475.4407


error =

   201


ans =

        1381


confianza =

   0.0000 +10.5748i


numGaus =
```

```
    16


numNodos =

     3

EM iteration 1, ll = -606243.9250
EM iteration 2, ll = -177461.3223
EM iteration 3, ll = -67397.7856
EM iteration 4, ll = -13958.9917
EM iteration 5, ll = 1753.0433
EM iteration 6, ll = 6227.6163
EM iteration 7, ll = 7897.7603
EM iteration 8, ll = 9289.8824
EM iteration 9, ll = 10090.9555
EM iteration 10, ll = 10695.1913
EM iteration 11, ll = 11161.5406
EM iteration 12, ll = 11389.2193
EM iteration 13, ll = 11478.5101
EM iteration 14, ll = 11508.7446
EM iteration 15, ll = 11583.7775
EM iteration 16, ll = 11653.9628


error =

   248


ans =

        1381


confianza =

   0.0000 +13.0537i


numGaus =

    18


numNodos =

     3

EM iteration 1, ll = -607042.7657
EM iteration 2, ll = -176218.2175
EM iteration 3, ll = -66116.1604
EM iteration 4, ll = -7509.8876
EM iteration 5, ll = 4877.2459
EM iteration 6, ll = 9411.9968
EM iteration 7, ll = 11504.4745
EM iteration 8, ll = 12592.3955
EM iteration 9, ll = 13222.8660
EM iteration 10, ll = 13633.5796
EM iteration 11, ll = 14039.2747
EM iteration 12, ll = 14297.1645
EM iteration 13, ll = 14611.2084
EM iteration 14, ll = 14906.9883
EM iteration 15, ll = 15264.5744
```

```
EM iteration 16, ll = 15449.5824


error =

   220


ans =

        1381


confianza =

   0.0000 +11.5769i


numGaus =

    20


numNodos =

     3

EM iteration 1, ll = -607140.7927
EM iteration 2, ll = -176779.3388
EM iteration 3, ll = -69348.2054
EM iteration 4, ll = -11047.6050
EM iteration 5, ll = 4650.4969
EM iteration 6, ll = 10214.2813
EM iteration 7, ll = 12973.9772
EM iteration 8, ll = 14589.4506
EM iteration 9, ll = 15752.1238
EM iteration 10, ll = 16372.0129
EM iteration 11, ll = 16911.9704
EM iteration 12, ll = 17617.9027
EM iteration 13, ll = 18027.1113
EM iteration 14, ll = 18252.8195
EM iteration 15, ll = 18408.8515
EM iteration 16, ll = 18657.4460


error =

   257


ans =

        1381


confianza =

   0.0000 +13.5284i
```