

# Aprendizaje Automático

## Práctica 2: Modelos Gráficos Probabilísticos

El objetivo de esta práctica es experimentar con Modelos Gráficos Probabilísticos o Redes Bayesianas y para su realización utilizaremos el toolkit *BNT* para Matlab que ofrece una serie de funciones realmente útiles en la tarea que nos ocupa.

En primer lugar realizaremos un aprendizaje de las tablas de probabilidades de cada variable del ejemplo “*Sprinkler*” que aparece en el boletín para comprobar qué ocurre cuando el número de muestras varía. Así, partiendo de un grafo del ejemplo previamente implementado en Matlab y de la red bayesiana creada a partir del mismo los pasos serían los siguientes:

1. Generar 1000 muestras aleatorias completas a partir de la red bayesiana.

```
semilla = 0; rng(semilla);  
nMuestras = 1000;  
muestras = cell(N, nMuestras);  
for i=1:nMuestras muestras(:,i) = sample_bnet(redB); end
```

2. Crear una nueva red vacía pero con la misma estructura que la anterior.

```
redAPR = mk_bnet(grafo, tallaNodos);  
redAPR.CPD{C} = tabular_CPD(redAPR, C);  
redAPR.CPD{R} = tabular_CPD(redAPR, R);  
redAPR.CPD{S} = tabular_CPD(redAPR, S);  
redAPR.CPD{W} = tabular_CPD(redAPR, W);
```

3. Estimar las distribuciones de probabilidad de cada nodo a partir de las muestras aleatorias y mediante la función de *BNT* **learn\_params()**.

```
redAPR2=learn_params(redAPR, muestras);
```

Si mostramos la tabla de probabilidades de cualquier variable (por ejemplo la W) y la comparamos con la de la red aprendida con 100 muestras podemos observar que al aumentar el número de datos las probabilidades son más similares a las de la tabla original, es decir que aumenta la precisión.

- Aprendizaje con 100 muestras:

```
dispcpt(TPCaux{W})  
1 1 : 1.0000 0.0000  
2 1 : 0.0556 0.9444
```

```
1 2 : 0.0435 0.9565
2 2 : 0.0000 1.0000
```

- Aprendizaje con 1000 muestras:

```
dispcpt(TPCaux{W})
1 1 : 1.0000 0.0000
2 1 : 0.1085 0.8915
1 2 : 0.0905 0.9095
2 2 : 0.0250 0.9750
```

- Tabla original:

S	R	W	P(W   S,R)
1	1	1	1
2	1	1	0.1
1	2	1	0.1
2	2	1	0.01
1	1	2	0
2	1	2	0.9
1	2	2	0.9
2	2	2	0.99

También podemos realizar el experimento para una red que aprende de un conjunto de muestras incompletas. Para ello procederemos de la siguiente manera:

1. Copiamos las muestras y ocultamos el 50% de los valores de las variables aleatorias.

```
muestrasS = muestras;
semilla = 0; rng(semilla);
ocultas = rand(N, nMuestras) > 0.5;
[I,J] = find(ocultas);
for k=1:length(I) muestrasS{I(k), J(k)} = []; end
```

2. Creamos una nueva red vacía de manera idéntica al caso anterior.
3. Procedemos al aprendizaje por Esperanza Maximización mediante la función **learn\_params\_em()**

```
maxIter = 1000;
eps = 1e-4;
```

```

semilla = 0; rng(semilla);
[redEM2, trazaLogVer] = learn_params_em(motorEM, muestrasS, maxIter,
eps);
auxTPC = cell(1,N);
for i=1:N s=struct(redEM2.CPD{i}); auxTPC{i}=s.CPT; end

```

Si mostramos la tabla de probabilidades de cualquier variable (por ejemplo W) y la comparamos con la de la red aprendida a partir de 100 muestras y 100 iteraciones podemos observar que, aunque es menos precisa que en el caso de las muestras completas, la similitud con la tabla original es mayor cuanto mayor es la cantidad de muestras e iteraciones.

- Aprendizaje con 100 muestras y 100 iteraciones:  

```

dispcpt(TPCaux{W})
1 1 : 0.9996 0.0004
2 1 : 0.0340 0.9660
1 2 : 0.0071 0.9929
2 2 : 0.1329 0.8671

```
- Aprendizaje con 1000 muestras y 1000 iteraciones:  

```

dispcpt(TPCaux{W})
1 1 : 0.9988 0.0012
2 1 : 0.2952 0.7048
1 2 : 0.1496 0.8504
2 2 : 0.0601 0.9399

```

A continuación cambiaremos el ejemplo “*Sprinkler*” por el del cáncer de pulmón que aparece en el tema 5 de la asignatura. Nuestro objetivo es implementar la red bayesiana correspondiente para obtener información relevante acerca de las distribuciones de probabilidad. El script para la realización de este ejercicio está incluido en el fichero comprimido adjunto a la tarea pero los pasos a seguir serían los siguientes:

1. Definir el grafo asociado al ejemplo del cáncer del pulmón, que está formado por 5 nodos.

```

N=5; P=1; F=2; C=3; X=4; D=5;
grafo=zeros(N,N);
grafo([P F],C)=1;
grafo(C,[X D])=1;

```

2. Definir la información (discretitud y talla) de las variables y crear la red.

```

nodosDiscretos=1:N;
tallaNodos=[2 2 2 3 2];
redB=mk_bnet(grafo,tallaNodos,'discrete',nodosDiscretos);

```

3. Introducir las tablas de probabilidad correspondientes y crear el motor de inferencia.

```
redB.CPD{P}=tabular_CPD(redB,P,[0.1 0.9]);
redB.CPD{F}=tabular_CPD(redB,F,[0.3 0.7]);
redB.CPD{C}=tabular_CPD(redB,C,[0.08 0.03 0.05 0.001 0.92 0.97 0.95
0.999]);
redB.CPD{X}=tabular_CPD(redB,X,[0.7 0.1 0.2 0.1 0.1 0.8]);
redB.CPD{D}=tabular_CPD(redB,D,[0.65 0.3 0.35 0.7]);
motor = jtree_inf_engine(redB);
```

Una vez hecho esto, podemos obtener información acerca del sistema y predecir el valor de las variables dadas unas evidencias determinadas.

- Probabilidad de que un paciente no fumador no tenga cáncer de pulmón si la radiografía ha dado un resultado negativo pero sufre disnea:

```
evidencia = cell(1,N);
evidencia{X}=3;
evidencia{D}=1;
evidencia{F}=2;
[motor,logVerosim]=enter_evidence(motor,evidencia);
m=marginal_nodes(motor,C);
m.T
ans =
    0.0016
    0.9984
```

La probabilidad es del 99.84%.

- Explicación más probable de que un paciente sufra cáncer de pulmón:

```
evidencia=cell(1,N);
evidencia{C}=1
[explMaxProb,logVerosim]=calc_mpe(motor,evidencia)
explMaxProb =
    [2]    [1]    [1]    [1]    [1]
```

La explicación más probable es que la polución sea baja, el paciente sea fumador, que la radiografía haya dado positiva y que el paciente sufra disnea.

Por último, vamos a experimentar con un clasificador basado en mixturas de gaussianas para comprobar cómo varía su tasa de error dependiendo del número de gaussianas. Para ello, utilizaremos el conjunto de muestras “*Spam*” que vimos en la práctica 1 y las herramientas del toolkit *BNT*. La función que crea el clasificador basado en una red bayesiana, y que posteriormente calcula la probabilidad de error empírico, está incluida en

el fichero comprimido adjunto a la tarea. A continuación recogeremos en una tabla los resultados para diferentes números de gaussianas y posteriormente los analizaremos para determinar cómo influye (mejora o empeora) el clasificador dependiendo de este parámetro.

Número de gaussianas	Error empírico	Intervalos de confianza
2	0.0029	$P(0.0001 \leq p \leq 0.0057) = 0.95$
3	0	0
4	0.0029	$P(0.0001 \leq p \leq 0.0057) = 0.95$
5	0.0022	$P(0 \leq p \leq 0.0047) = 0.95$
6	0.0014	$P(0 \leq p \leq 0.0034) = 0.95$
7	0.0014	$P(0 \leq p \leq 0.0034) = 0.95$

En la tabla podemos observar que el mejor clasificador es el construido con 3 gaussianas porque no ofrece ningún error para el conjunto de muestras de prueba. Aún así, a partir de 4 y hasta las 6 gaussianas, la precisión del clasificador aumenta progresivamente hasta alcanzar el 99.86%.