Machine Learning 1 Homework 07 Theory Part

01.05.2016

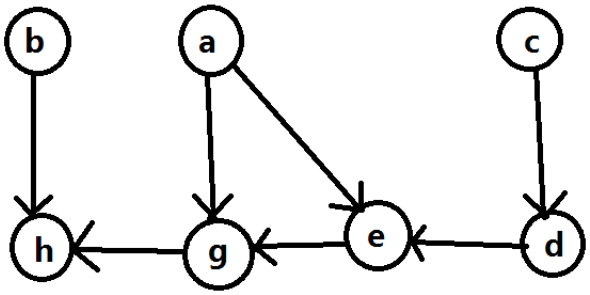
1. **Graphical Models and Conditional Independence** a) a and b are conditionally independent given the empty set.

   Check all the paths from a to b, we can see that they all pass through node g, then the path goes on to c,d or h. If it comes to c, then the arrows meet head-to-head at node g, it blocks the path. If it comes to h(or d), the arrows meet at node d(or h), which again blocks the path. This means all paths between a and b are blocked, so we can conclude that a and b are conditionally independent givent the empty set.

   b) a and b are not conditionally independent given h.

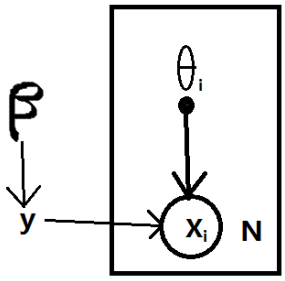
   Take a look at the path $a \to f \to g \to c \to b$, the arrows meet head-to-tail at f and c, neither f nor c is in set h. The arrows meet head-to-head at node g, but g has a descendant h, which is in the give set. So this path is unblocked. a and b are not conditionally independent given h.

   c)



2. **Graphical Models and Discriminants**

   a)



b)

$$\begin{aligned} P(y=k|\mathbf{x}) &= \frac{P(y=k,\mathbf{x})}{P(\mathbf{x})} \\ &= \frac{P(y=k)\Pi_{i=1}^{d} P(x_i|y=k)}{P(\mathbf{x})} \\ &= \frac{\beta_k \Pi_{i=1}^{d} \theta_{ik}^{x_i}(1-\theta)^{1-x_i}}{P(\mathbf{x})} \end{aligned}$$

In order to maximize $P(y=k|\mathbf{x})$, we have maximize $\beta_k \Pi_{i=1}^{d} \theta_{ik}^{x_i}(1-\theta)^{1-x_i}$

3. **Finding the Most Likely Solution** This is solved by programming. Here is the result. l,k,b,o stand for living room, kitchen, bedroom and office.

```
============================================================
======= the most likely path from l to l,k,b,o ==========
============================================================

in 1 steps:
l -> l
l -> k
l -> b
l -> o

in 2 steps:
l -> k -> l
l -> o -> k
l -> k -> b
l -> k -> o

in 3 steps:
l -> o -> k -> l
l -> k -> o -> k
l -> o -> k -> b
l -> k -> b -> o

in 4 steps:
l -> k -> o -> k -> l
l -> k -> b -> o -> k
l -> k -> o -> k -> b
l -> o -> k -> b -> o

in 5 steps:
l -> k -> b -> o -> k -> l
l -> o -> k -> b -> o -> k
l -> k -> b -> o -> k -> b
l -> k -> b -> o -> k -> o

in 6 steps:
l -> o -> k -> b -> o -> k -> l
l -> k -> b -> o -> k -> o -> k
l -> o -> k -> b -> o -> k -> b
l -> k -> b -> o -> k -> b -> o

the optial path is:  l -> k -> b -> o -> k -> b -> o
```

In [ ]: