Mục lục

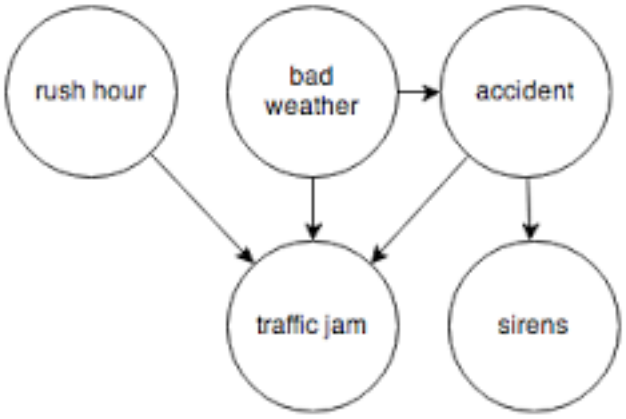
1 Khái quát về mạng Bayes

Là một đồ thị có hướng phi chu trình mà trong đó:

- Các nút biểu diễn các biến ngẫu nhiên
- Các cạnh biểu diễn các phụ thuộc xác suất giữa các biến ngẫu nhiên tương ứng

Nếu có một cạnh từ nút A tới nút B, thì B phụ thuộc trực tiếp vào biến A, và A được gọi là cha của B, còn B là con của A.

- Nếu biến được thể hiện bởi một nút được quan sát thì nó được gọi là nút chứng cứ. Nếu không, nút đó được gọi là nút ẩn.
- Nếu X_i không có cha, ta nói rằng phân phối xác suất của nó là không có điều kiện.



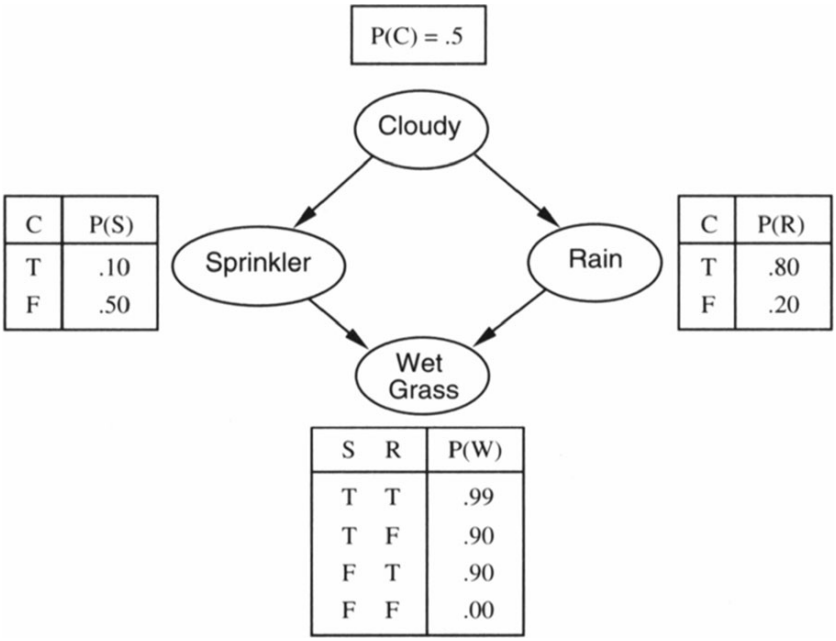
Hình 1: Một mạng Bayes đơn giản

2 Tính chất của mạng Bayes

- Mạng Bayes là một đồ thị định hướng phi chu trình.
- Về mặt định lượng, các nút được đính kèm một bảng phân phối xác suất phụ thuộc vào các nút cha mẹ của nó (CPD - conditional probability distribution) (nếu có).
- Nếu với mỗi biến X_i, tập hợp các biến cha được ký hiệu bởi $parents(X_i)$ thì phân phối có điều kiện đồng thời bởi các biến đó được thể hiện bởi công thức:

$$Pr(X_1, ..., X_n) = \prod_{i=1}^{n} Pr(X_i | parents(X_i))$$

3 Ví dụ



Hình 2: Ví dụ về mạng Bayes

$$\begin{aligned} P(C,S,R,W) &= P(W|S,R,C)P(S|R,C)P(R|C)P(C) \\ &= P(W|S,R)P(S,C)P(R|C)P(C) \end{aligned}$$

$$\begin{aligned} P(R=T|W=T) &= \frac{P(R=T,W=T)}{P(W=T)} \\ &= \frac{\sum_{S,C} P(R=T,S,C,W=T)}{\sum_{S,R,C} P(W=T,S,R,C)} \end{aligned}$$

4 Học mạng Bayes

- Công việc xây dựng mạng phức tạp với con người. Do vậy phải thực hiện công việc học tập cấu trúc và tham số mạng từ dữ liệu.
- Vấn đề này được gọi là BN learning (học mạng Bayes), có thể được mô tả như sau:
  - Cho bộ dữ liệu và thông tin liên quan.

  - Tìm cấu trúc của mô hình mạng Bayes đồng thời xác định định lượng các thông số của nó.

- Độ tốt của giải thuật learning: Đưa ra hàm tính điểm tương ứng với mạng Bayes tìm được.

4.1 Sơ lược về Học cấu trúc

- Giả sử dữ liệu được sinh ra từ một mạng Bayes và tất cả các biến là quan sát được, việc tối ưu hóa dựa trên phương pháp tìm kiếm có thể được dung để tìm hiểu cấu trúc mạng.

- Một cách tiếp cận khác là sử dụng chiến lược tìm kiếm địa phương để tìm một cấu trúc tại địa phương, tối ưu hóa điểm số của nó một cách cục bộ, rồi tìm cách mở rộng địa phương ra hướng về toàn cục.

4.2 Sơ lược về Học tham số

- Đối với mỗi biến X, cần chỉ ra phân bố xác suất X theo điều kiện thông tin từ các cha của X.

- Các phân bố có điều kiện này bao gồm các tham số chưa biết, phải được ước lượng từ dữ liệu.

- Một trong những cách ước lượng là phương pháp MLE, và EM

4.3 Bốn trường hợp BN learning

- Maximum – likelihood estimation

- EM (or gradient ascent), MCMC

- Search through model space

- EM + search through model space

Bảng 1: Bốn trường hợp của học mạng Bayes

Case	BN structure	Observability	Proposed learning method
1	Known	Full	Maximum-likelihood estimation
2	Known	Partial	EM (or gradient ascent), MCMC
3	Unknown	Full	Search through model space
4	Unknown	Partial	EM + search through model space

Maximum likelihood estimation (MLE)

- Sử dụng khi biết cấu trúc mạng và có thể quan sát toàn bộ dữ liệu
- Còn được gọi là phương pháp hợp lý cực đại
- Tập dữ liệu gồm m trường hợp độc lập nhau.
- Cho tập dữ liệu huấn luyện: $\Sigma = \{x_1, x_2, ..., x_m\}$, trong đó $x_i = (x_{i,1}, ..., x_{i,n})$ Và tập tham số $\Theta = (\theta_1, ..., \theta_n)$, trong đó θ_j là vector tham số của biến ngẫu nhiên X_j.
- Hàm tính điểm cho kết quả của tham số là hàm likelihood:

$$logL(\Theta|\Sigma) = \sum_{i=0}^{m}\sum_{j=0}^{n} logP(x_{i,j}|\pi_j, \theta_j)$$

EM (or gradient ascent), MCMC

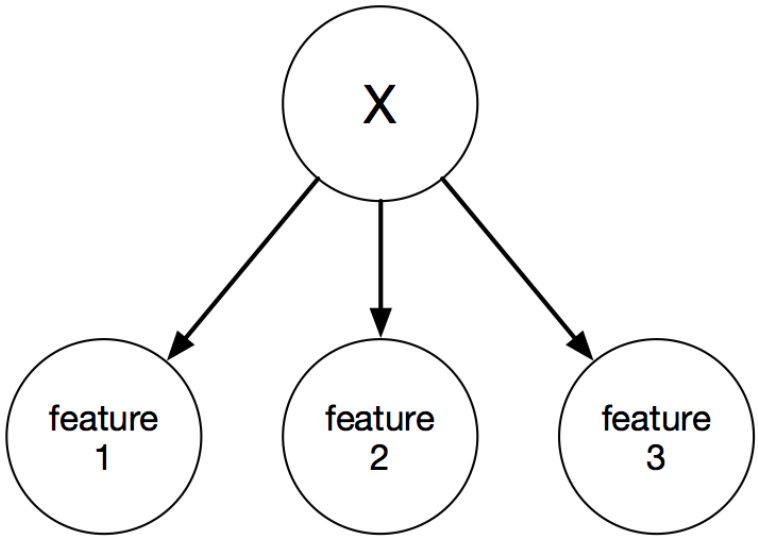
- Cho trước cấu trúc mạng, nhưng chỉ quan sát được một phần
- Sử dụng tối đa hóa kỳ vọng (expectation maximization)
- Thuật toán này thực tế là tìm kiếm một MLE địa phương của các tham số.
- MCMC (Markov chain Monte Carlo) là một cách tiếp cận khác đã được sử dụng để ước lượng tham số của mô hình BN.

Search through Model Space

- Đây là trường hợp không biết cấu trúc mạng nhưng quan sát được toàn bộ.
- Mục tiêu là tìm được một cấu trúc mạng có thể giải thích tốt nhất mối quan hệ giữa các biến.
- Vấn đề tìm kiếm đòi hỏi một hàm tính điểm và một chiến lược tìm kiếm
- Đây là một vấn đề NP – khó vì để duyệt toàn bộ cấu trúc định hướng phi chu trình (DAG) đòi hỏi thời gian cấp siêu lũy thừa đối với số biến.
- Một cách tiếp cận được đề xuất là xây dựng mạng Naïve BN.

Naïve BN

- Những biến độc lập được nhóm vào một Class, trong đó có một nút cha duy nhất, các nút còn lại đều là con của nó.
- Sử dụng mạng Naïve BN trong thực nghiệm Cho thấy kết quả đem lại khá khả quan trong Nhiều trường hợp thực tế.



Hình 3: Naïve BN

EM + search through model space

- Kết hợp hai phương pháp xây dựng cấu trúc lẫn tham số trong trường hợp có ít thông tin nhất (không biết cấu trúc mạng, quan sát được một phần)
- Để có thể sử dụng hàm tính điểm trong việc tìm kiếm cấu trúc, ta phải loại bỏ các nút ẩn và các tham số đi kèm nó.
- Điều này thường là không giải quyết được nên thường sử dụng một phép xấp xỉ tiệm cận gọi là tiêu chuẩn thông tin (BIC) hay mô tả tối thiểu.

Bayesian information criterian (BIC)

- Xem xét hai thông số là thông số likelihood và thông số penalty.
- Trong đó likelihood đánh giá độ hợp lý tham số, còn penalty đánh giá độ phức tạp của mô hình
- Hai thông số này cần được cân bằng để có thể đưa ra điểm số tốt cho thuật toán.

5 Học mạng Bayes bằng Python

Sử dụng thư viện pgmpy để thao tác trên mạng Bayes.

5.1 Học tham số

- Maximum likelihood estimation: Tính trực tiếp xác suất dựa trên mẫu bằng cách coi tỉ lệ các thành phần trong mẫu như tỉ lệ trong thực tế.

- Bayesian Parameter Estimation
  - Bdeu
  - K2
  - Dirichlet

Maximum likelihood estimation

	fruit	size	tasty
0	banana	large	yes
1	apple	large	no
2	banana	large	yes
3	apple	small	yes
4	banana	large	yes
5	apple	large	yes
6	banana	large	yes
7	apple	small	yes
8	apple	large	yes
9	apple	large	yes
10	banana	large	yes
11	banana	large	no
12	apple	small	no
13	banana	small	no

fruit	apple		banana	
size	large	small	large	small
tasty				
no	1.0	1.0	1.0	1.0
yes	3.0	2.0	5.0	0.0

fruit	fruit(apple)	fruit(apple)	fruit(banana)	fruit(banana)
size	size(large)	size(small)	size(large)	size(small)
tasty(no)	0.25	0.3333333333333333	0.16666666666666666	1.0
tasty(yes)	0.75	0.6666666666666666	0.8333333333333334	0.0

Bayesian Parameter Estimation

- Diriclet: Đưa ra một pseudo_counts, là một danh sách các số được dùng để cộng vào mẫu khảo sát của bộ dữ liệu huấn luyện một cách trực tiếp trước khi ước lượng.

  Ví dụ: Pseudo_counts ta chọn cho bộ dữ liệu là [1,2]

- K2: Giống với Diriclet, có điều pseudo_counts có các phần tử là 1.

- Bdeu: Cũng dựa trên Diriclet, có điều các phần tử của Pseudo_counts được tính dựa trên một tham số là equivalent_sample_size và khi đó, các phần tử của pseudo_counts có giá trị là:

$$\frac{equivalent\_sample\_size}{NodeCardinality * np.prod(ParentsCardinality)}$$

5.2 Học cấu trúc

Để học cấu trúc, ta thường có ba hướng:

- Một là sử dụng Score-base structure learning
- Hai là sử dụng Constraint-base structure learning
- Ba là kết hợp cả hai: Hybird structure learning

5.3 Score – base

- Sử dụng các hàm tính điểm để đánh giá độ tốt của cấu trúc mạng.
- Các hàm tính điểm trong thư viện pgmpy:
  - BdeuScore
  - K2Score
  - BicScore
- Chiến lược tìm kiếm: Với mạng có rất ít nodes, sử dụng exhaustive search, với mạng có nhiều hơn các node cần phải sử dụng heuristic search. Một trong những giải thuật tìm kiếm đề xuất là HillClimbSearch (thực hiện một tìm kiếm địa phương bằng giải thuật tham lam)

```
bic = BicScore(data)
hc = HillClimbSearch(data, bic)
best_model = hc.estimate()
print(best_model.edges())
```

```
[('DPQ', 'RD'), ('DPQ', 'DFT'), ('DI', 'C'), ('DI', 'DPQ')]
[Finished in 89.8s]
```

```
bdeu = BdeuScore(data, equivalent_sample_size=0.05)
hcbdeu = HillClimbSearch(data, bdeu)
best_model1 = hcbdeu.estimate()
print(best_model1.edges())
```

```
[('DPQ', 'RD'), ('DPQ', 'DI'), ('DI', 'C'), ('DFO', 'DPQ')]
[Finished in 120.2s]
```

```
k2 = K2Score(data)
hck2 = HillClimbSearch(data,k2)
best_model1 = hck2.estimate()
print(best_model1.edges())
```

```
[('DPQ', 'RD'), ('DPQ', 'DI'), ('DI', 'C'), ('DFO', 'DPQ')]
```

Constraint-Base

Gồm hai bước:

- Bước 1: Xác định các node độc lập với nhau.
  - Sử dụng hàm test_conditional_independence(X,Y,Zs)
- Bước 2: Xây dựng một mẫu mạng dựa trên các node độc lập đã xác định, gồm 3 bước:
  1. Xây dựng một bộ khung mạng vô hướng bằng hàm estimate_skeleton()
  2. Định hướng các cạnh của mạng để đạt được một mạng có hướng phi chu trình địa phương bằng hàm skeleton_to_pdag()
  3. Mở rộng mạng địa phương để đạt được mạng toàn cục pdag_to_dag()

```
est = ConstraintBasedEstimator(data)
print(est.estimate(significance_level=0.01).edges())
```

```
[('DPQ', 'DFO'), ('DFT', 'DI'), ('TQ', 'DFO'), ('RD', 'DFT'), ('RD', 'DI'), ('OU', 'DFO')]
[Finished in 5721.2s]
```

Hybrid Structure Learning

Gồm hai bước:

- Bước 1: Học một khung đồ thị vô hướng sử dụng thủ tục constraint-based construction MMPC.
- Bước 2: Định hướng các cạnh sử dụng tối ưu score-based (BdeuScore + modified hill – climbing).