베이지안 네트워크에서의 효율적인 탐색 기법에 관한 연구

황성철° 이일병
연세대학교 컴퓨터과학과
{franz82°, yblee}@csai.yonsei.ac.kr

The study of Efficient Learning Method for Bayesian Network

Sungchul Hwang°, Yillbyung Lee
Dept. of Computer Science, Yonsei University

요 약

불확실성이 존재하는 대용량의 데이터에서의 추론과 각 특성들 간의 상관관계를 파악하기 위해서 사용되는 기법이 베이지안 네트워크 학습 방법이다. 본 논문에서는 베이지안 네트워크 학습 방법에서 발생할 수 있는 NP-Hard문제를 해결하게 위한 효율적인 탐색 기법을 구현하여 실제 네트워크 학습에서 적용시키고, 어떻게 개선되는지 알아본다.

1. 서 론

베이지안 네트워크 학습은 최근 대용량의 데이터를 통한 추론을 위해 다양하게 사용되고 있다. 실세계로부터 얻어지는 데이터들의 인과관계와 서로 어떠한 관련성을 가지고 발생되는지 그 관계를 추론하고 표현할 수 있게 해준다. 이처럼 데이터의 실제적인 모델을 찾기 위한 구조의 학습 과정에서는 주어진 데이터에 기반하여 모델을 선택하는 과정을 필요로 한다. 이러한 구조 학습에 있어서의 방법론은 크게 두 가지로 구분할 수 있다. 첫 번째로, 구조 학습을 CSP(Constraint Satisfaction Problem)의 형태로 다루는 제약기반접근법(Constraint-Based Approach)이 있다. 이것은 데이터간의 상호 의존성을 찾아내고자 하는 알고리즘으로 주로 통계학적 가설검증 방법을 사용한다. 두 번째 방법은 구조 학습을 최적화 문제(Optimization Problem)의 측면에서 다루는 탐색과 스코어링에 기반한(Searching and Scoring-Based Approach) 방법이다. 여기서의 구조 학습은 최적화의 문제로서 스코어를 최대로 하는 구조 S_{opt}를 찾는 것이 문제이다. 이 알고리즘에서의 스코어링은 각각의 모델이 주어진 데이터에 대해 얼마나 적합한 형태를 갖는지에 따라 높은 점수를 부여한다. 한편, 데이터의 크기가 큰 경우 이러한 상황에서 쉽게 가장 최적화된 네트워크 구조를 찾는 것은 NP-Hard에 속하는 문제이다[1]. 따라서 이러한 탐색에서의 문제를 해결하기 위해서는 탐색 공간의 크기를 최소화하여 효율적으로 빠른 시간에 문제를 풀어가기 위한 기법이 요구된다. 특정 노드에서 다른 모든 나머지 노드 n-1개에 대해 부모노드로서의 연관성을 가지고 있는지를 테스트하게 되면 너무 많은 탐색이 필요해지기 때문에, 이러한 부분에서의 탐색을 줄이는 것이 요구된다. 본 논문에서는 이와 같은 관점에서의 가장 효율적인 네트워크 탐색 기법을 다룰 것이다.

"본 연구는 과기부 뇌신경정보학사업으로터 부분적인 지원을 받아 수행되었음."

2. 베이지안 네트워크(Bayesian Network)

베이지안 네트워크(Beyesian Network)는 $B=(G,\theta)$ 로 나타낼 수 있는 확률 그래프 모델(Probabilistic Graphical Model)이다[2]. 여기서, $G=(\nu,\varepsilon)$ 는 이러한 베이지안 네트워크의 구조를 표현하는 DAG(directed acyclic graph)이다. ν는 그래프에서의 노드 집합을 나타내며, ε은 간선 집합을 나타낸다. ν에 속하는 노드 X에 대해, X로의 직접적인 링크가 있는 것을 부모노드라고 하며, 이를 $pa(X)$로 표기한다. 전체 베이지안 네트워크(BN)에는 N개의 변수 $X_i(1 \le i \le N)$가 존재한다.

조건부확률 $\theta_{ijk}=P(X_i=k|pa(X_i)=j)$ 는 X_i의 부모노드 $pa(X_i)$가 j 의 값을 가질 때, X_i가 k의 값을 가질 확률을 나타낸다. 한편, X_i가 루트 노드일 경우에는 θ_{ijk}가 X_i의 marginal probability에 해당한다. θ는 모든 모수 θ_{ijk}의 집합을 나타낸다[3].

2.1 모수 추정(Parameter Estimation)

MLE(Maximum Likelihood Estimation)가 베이지안 네트워크를 구성하는 데이터에 잘 맞는 것처럼 보이지만, MLE의 단점은 그것이 특정 데이터에만 Overfitting 되는 경향이 있다는 것이다. 적은 양의 데이터에 대해서만 학습을 시키고 추론을 하게 될 경우에 MLE접근법으로는 보다 현실적인 모델과는 다를 수 있다는 것이다. 이러한 문제점을 해결하기 위해서는 모수 추정에 있어서 베이지안 접근법을 사용하는 것이 보다 효율적이다. 베이지안 접근법에서는 기존의 측정된 데이터에 초기 사전분포 $P(\theta)$를 가정하여 모수 추정을 시행하기 때문에 MLE의 문제점을 보완할 수 있는 장점이 있다. 새로운 데이터가 있을 때, 기존의 $P(\theta)$와의 조합으로 사전 분포를 업데이트시켜가면서 초기의 분포를 점차 실제 모델에 맞게 변화되는 과정을 거치게 된다. 이렇게 업데이트 된 분포를 사후분포 $P(\theta|D)$라고 한다[2].

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

여기에서 P(D)는 주변우도(marginal likelihood) 라고 하며, 데이터에서 모든 가능한 모수들의 평균을 나타낸다. 하지만 이것은 정규화 상수(Normalizing Constant)로, θ와는 독립적이기 때문에 네트워크 점수를 계산하는 데에는 반영하지 않는다.

사전 분포 $P(\theta)$는 경우에 따라 균등분포(Uniform Distribution) 또는 디리클레 사전분포(Dirichlet Prior)를 사용할 수 있다. 균등분포는 모든 경우의 확률적 분포가 동등하게 가정되는 것이며, 디리클레 분포의 경우 사전 분포를 다음과 같이 나타낼 수 있다[3].

$$P(\theta) = \text{Dirichlet}(\alpha_{x^1|u}, \ldots, \alpha_{x^K|u}) \sim \prod_{x^i|u} \theta_{x^i|u}^{\alpha_{x^i|u}-1}$$

위의 식에서 사용되는 모수(Hyperparameter) $\alpha_{x^i|u}$는 각각의 $x^i \in Val(x)$ 에 대응되는 것이다. 이는 실험 데이터 D를 적용하기 이전에 가상의 모수로 여기는 것으로, 부모 노드 $Pa_x = u$ 를 가지는 $X = x$가 있을 경우의 수(Pseudo Count)에 해당하는 것이다.

디리클레 사전분포는 전역적 모수 독립성(Global parameter independence)과 지역적 모수 독립성(Local parameter independence)을 만족하는 성질을 가지며, 이 두 가지 독립성이 모두 만족될 때, 사전분포 $P(\theta)$가 모수 독립성을 만족한다고 할 수 있다.

만약 이처럼 $P(\theta)$가 디리클레 분포를 따른다면, 사후분포 $P(\theta|D)$는 Sufficient Statistics M[x,u]에 의해 다음과 같이 표현될 수 있다.

$$P(\theta|D) = Dirichlet(\alpha_{x^1|u} + M[x^1, u], \ldots, \alpha_{x^K|u} + M[x^K, u])$$

$\alpha^* = \sum_{x^i|u} \alpha_{x^i|u}$ 는 효율적인 샘플 크기를 나타내며, 이것이 클수록 사전분포에 대해 강한 확신을 가지고 있고, 반대로 D가 클수록 사전분포에 대한 확신이 적다는 것을 나타낸다. M[x,u]는 D에서 X_i가 x의 값을 가지고 $pa(X_i)$가 u의 값을 가지는 경우의 수를 나타낸다.

2.2 구조 학습(Structure Learning)

주로 구조 학습을 위해 사용되는 방법은 Greedy-hill climbing과 Simulated Annealing방법 등이 있으며, 이러한 방법들은 가장 많은 스코어를 기록하는 네트워크를 찾기 위한 탐색 기법들로 사용되고 있다. 각각의 노드에서 다른 노드로의 간선(edge)의 추가, 역방향, 삭제 등의 지역적인 변화에 따라 최대 스코어를 갖는 네트워크를 찾아나가는 방식이다. Greedy-hill climbing의 경우 지역 국소에 빠지는 경향이 있지만, 이러한 문제점을 해결하기 위해 본 논문에서는 Random restart방식을 사용하여, 문제를 최소화 하였다. 구조학습에서의 탐색을 효율적으로 하기 위해서 부모노드의 수를 제한하는 방법을 적용시킬 수 있다. 전체 N개의 노드 중, 현 노드를 제외한 N-1개의 모든 노드를 부모노드로 고려하여 구조 학습을 진행해 나가는 것은 효율성 측면에서도 좋지 않으며, NP-Hard에 해당하는 문제로서 해결하기가 쉽지 않다. 따라서, 현재 노드에서의 가능한 부모노드 집합을 구하기 위해 각 노드들 간의 상관관계를 파악하는 과정이 필요하게 된다. 이 과정에서 사용하는 개념이 Mutual Information이다.

2.2.1 BDe *(Bayesian Dirichlet Equivalence)*

네트워크 학습에서 사용되는 스코어링 척도는 *BDe(Bayesian Dirichlet Equivalence)*를 사용하였다[5]. 사전 확률 분포를 디리클레 분포로 가정할 때의 스코어링 척도는 다음과 같은 형태를 가진다.

$$Score_B(G, D) = \sum_i FamScore_B(X_i, pa(X_i) : D)$$

여기서 G는 전체 베이지안 네트워크를 나타내며, D는 주어진 모든 데이터를 나타낸다. 위의 식에서 나타난 것처

럼 전체 네트워크의 스코어가 각각의 노드와 관련된 (Family)부분에 대한 스코어로 분할하여 계산 가능한 것은 최적의 네트워크 모델을 찾는 데에 있어서 매우 효율적이다. FamScore는 각각의 변수 X_i에 대한 스코어를 나타내며, 다음과 같이 나타낸다.

$$FamScore_B(X_i, Pa(X_i):D)$$
$$= \log\left[\prod_{u\in Pa(X_i)} \frac{\Gamma(\alpha_{x_i|u})}{\Gamma(\alpha_{x_i|u}+M[u])} \prod_{x_{ij}\in X_i} \frac{\Gamma(\alpha_{x_{ij}|u}+M[x_{ij},u])}{\Gamma(\alpha_{x_{ij}|u})}\right]$$

여기서의 모수(Hyperparameter)값은 다음과 같이 결정된다.

$$\alpha_{x_i|u_i} = MP(x_i, u_i)$$

2.2.2 Mutual Information Percentage

두 개의 노드 X, Y가 있을 때, 이들이 서로 얼마나 큰 영향을 주고 있는지 측정하기 위해 사용되는 것이 Mutual Information이다. 이것은 간단히 X, Y의 관계를 U(Y)와 U(X|Y)를 계산함으로서 구할 수 있다. 엔트로피 U(X)는 다음과 같이 나타낸다[4].

$$U(X) = \sum_{x_i} P(x_i)\log_2 \frac{1}{P(x_i)}$$

한편, 이를 이용한 Mutual Information은 다음과 같이 나타낼 수 있다.

$$MI(X, Y) = U(Y) - U(Y|X)$$
$$MI(X, Y) = \sum_{x,y} P(x,y)\log_2\left(\frac{P(x,y)}{P(x)P(y)}\right)$$

하지만, 위에서의 Mutual Information은 X와 Y의 관계만을 나타낼 뿐, X에서 Y로의 영향, 또는 그 반대의 경우를 나타낼 수 없기 때문에, 부모노드인지 자식노드가 될지 결정할 수 없다. 따라서 이러한 문제를 해결하기 위해 부모노드의 후보를 선택하는 과정에서 Mutual Information Percentage를 적용하여 어떠한 노드가 영향을 주는지를 백분율로 쉽게 나타낼 수 있다. 이러한 방법을 사용하여, Y에 부모노드로서 영향을 주는 노드 X의 집합을 구할 수 있다.

$$MI\%(X, Y) = \frac{MI(X, Y)}{U(Y)} \cdot 100$$
$$= \frac{U(Y) - U(Y|X)}{U(Y)} \cdot 100$$

3. 실험 결과 및 향후과제

실험은 Alarm Network 데이터를 사용하였으며, Intel OpenPNL C++ library를 사용하여 구현하였다. 학습방법은 Greedy Hill-Climbing Random Restart를 사용하였고, 스코어링 척도로 BDe를 사용하였다.

Method	반복횟수	시간(초)	KL
MI% 10	1	4.5	3.789
	2	6.7	1.563
	3	8.3	0.967
MI% 20	1	5.0	3.276
	2	7.3	1.456
	3	8.9	0.798
MI 10	1	6.5	2.574
	2	8.2	0.512
	3	9.6	0.183
MI 20	1	7.7	1.945
	2	9.3	0.423
	3	10.8	0.067

표 1. 실험 결과

위의 결과는 학습에서 MI%와 MI의 가능한 부모노드의 수에 제한(10~20)을 두어 적용시킨 결과를 나타낸다. 표 1에서 볼 수 있듯이 본 논문에서 사용된 방법을 학습에서의 부모노드 선정에서 적용한 결과 학습 시간에서의 큰 변화가 있었으며 학습의 효율성 증진에 효과가 있음을 알 수 있다.

참고문헌

[1] D.M. Chickering, Learning Bayesian networks is NP-Complete. In Learning from Data, Artificial Intelligence and Statistics, 1996

[2] K. Sivakumar, R. Chen, and H. Kargupta, Learning Bayesian Network Structure from Distributed Data. In Proceedings of the 3rd SIAM International Data Mining Conference, pp. 284-288, 2003

[3] D. Heckerman. A tutorial on learning with Bayesian networks. In M. I. Jordan, editor, Learning in Graphical Models, 1998

[4] Shannon, C., Warren, W., The Mathematical Theory of Communication, University of Illinois Press, Urbanaand Chicago, 1949

[5] Shulin Yang, Kuo-Chu Chang, Comparison of Score Metrics for Bayesian Network Learning, IEEE Transactions on Systems, Man and Cybernetics-Part A:Systems and Humans, VOL.32, No.3, 2002