진화 알고리즘을 이용한 베이지안 네트워크 병합의 최적화

김경중, 조성배
연세대학교 컴퓨터과학과
{kjkim, sbcho}@cs.yonsei.ac.kr

Optimization of Bayesian Networks Aggregation Using Genetic Algorithm

Kyung-Joong Kim, Sung-Bae Cho
Dept. of Computer Science, Yonsei University

요 약

베이지안 네트워크 병합은 여러 개의 베이지안 네트워크를 하나의 네트워크로 합치는 것을 말한다. 일반적으로 사용되는 병합 알고리즘은 병합 순서에 따라 최종결과 네트워크의 복잡도가 달라지는 문제를 갖고 있고, 최종 병합 네트워크의 에지 수를 최소화 하는 병합 순서를 찾는 것은 NP-hard라고 증명되었다. 본 논문에서는 최적의 병합 순서를 결정하기 위해 진화 알고리즘을 사용하는 방법을 제안한다. 해공간 분석을 통해 permutation index 표현방법이 해탐색에 유리함을 보이고 이를 이용한 진화 알고리즘을 제안한다. 실험결과, 기존의 휴리스틱과 greedy 탐색 방법에 비해 제안한 방법이 우수한 성능을 보였다.

1. 서론

베이지안 네트워크는 불확실한 상황을 확률적으로 표현하기 위한 그래프 모델이다. 네트워크의 구조는 변수들 사이의 확률적인 의존관계를 나타내며 각 노드는 조건부 확률 파라미터를 가지고 있다. 베이지안 네트워크의 구조는 데이터로부터 학습이 가능하며 전문지식이 있을 경우 전문가가 직접 구조를 설계하기도 한다.

동일한 도메인에 대해 한 명 이상의 전문가가 각각 베이지안 네트워크를 설계하는 경우 서로 다른 지식이 분산되어 통합적으로 관리되기 어렵다. 이러한 이유로 여러 개의 베이지안 네트워크를 병합하는 연구가 진행되어 왔다. 두 개의 베이지안 네트워크를 하나의 구조로 만드는 알고리즘이 제안되었으며[1], 여러 개의 네트워크를 결합할 때 결합된 네트워크의 에지 수를 최소화 하는 것은 NP-hard라는 것이 증명되었다[2]. 여러 개의 베이지안 네트워크가 주어졌을 때 결합된 네트워크가 최소의 에지 수를 가질 수 있도록 하는 결합 순서를 탐색하기 위해 진화 알고리즘을 사용하는 것을 제안한다. 해공간 분석을 통해 탐색공간의 특성을 파악하고 permutation index에 기반한 염색체 표현방법을 제안한다.

결합 네트워크의 에지수를 최소화 하려는 것은 복잡도를 감소시켜 일반성을 높이고, 입력해 주어야 하는 조건부 확률의 개수를 줄이려는 것이다. 제안하는 방법의 유용성을 평가하기 위해 임의로 생성한 네트워크를 사용하였으며, 동일한 네트워크를 대상으로 휴리스틱과 greedy 방법과 성능을 비교한다.

2. 진화 알고리즘을 이용한 베이지안 네트워크 병합

베이지안 네트워크를 결합하는 가장 손쉬운 방법은 합집합, 교집합 연산자를 사용하는 것이다. 교집합 연산자의 경우 모든 네트워크에 공통적으로 구조가 병합 네트워크에 사용되어진다. 반면, 합집합 연산자의 경우 각 네트워크의 모든 변수와 에지가 병합 네트워크에 들어간다. 합집합 연산자의 경우, 사이클이 생기는 문제가 있을 수 있다. 여러 개의 네트워크를 합집합 하는 과정에서 새로운 에지를 추가함으로 인해 사이클이 생길 수 있으며 사이클을 막기 위해 해당 에지를 무시할 경우 정보의 손실이 발생할 수 있다. 이러한 문제를 해결하기 위해 reverse 연산자를 사용한다(그림 1). 이 연산자를 사용하면 사이클을 만드는 에지의 방향을 바꾸면서 기존 변수들 사이의 의존관계를 그대로 유지할 수 있다.

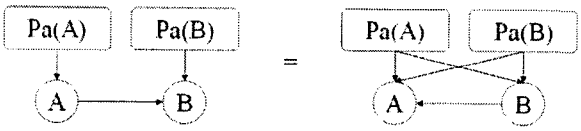


그림 1. Reverse 연산자

N이 베이지안 네트워크의 개수일 때, 하나의 모델로 합쳐져야 하는 베이지안 네트워크의 집합을 $S=\{B_1, B_2, \ldots, B_N\}$이라고 하고, 결합은 점증적으로 이루어진다. 최초에 B_1과 B_2를 결합하여 B_{12}가 만들어지고, B_3와 B_{12}가 결합하여 B_{123}가 만들어진다. 이와 같은 과정을 마지막 베이지안 네트워크까지 반복하면 최종 결과는 $B_{123\ldots N}$이 된다.

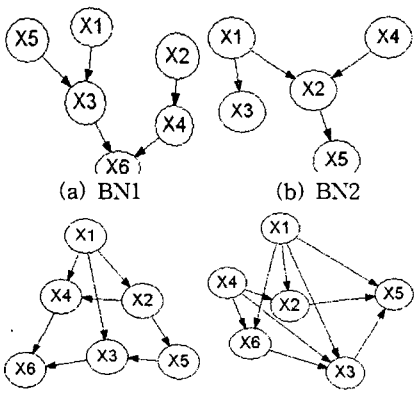
본 논문에서 해결하려는 문제는 베이지안 네트워크를 결합해 나가는 순서를 결정하는 것이다. 순서에 따라 최종 병합 베이지안 네트워크의 에지 수가 달라진다. $B_{12345\ldots N}$과 $B_{21345\ldots N}$는 동일하지 않다. N개의 베이지안 네트워크에 대해 $N!$ 개의 결합방법이 존재한다. 두 개의 베이지안 네트워크를 결합하는 자세한 방법은[1]을 참조하기 바란다.

B_1과 B_2를 결합하여 B_{12}를 만든다고 했을 때 B_1은 원본 네트워크(source network)라고 불리고 B_2는 목표 네트워크(target network)라고 불린다. 원본 네트워크의 구조를 그대로 두고, 목표 네트워크의 구조를 합치는 과정으로 볼 수 있다. 합치는 과정에서 목표 네트워크의 에지들은 DIR, REV, EQ의 세 가지 그룹으로 분류가 되며 그룹에 따라 서로 다른 방식으로 원본 네트워크에 삽

입된다. 원본 네트워크에 속한 모든 변수들의 위상값(topological value)을 계산하고 이 값을 토대로 목표 네트워크의 변수들을 세 그룹으로 분류한다. 변수 x_i의 위상값은 원본 네트워크의 위상그래프(topological graph)의 최상위 노드로부터 해당 변수까지의 최장거리 패스(path) 길이로 정의한다. 낮은 위상값을 가진 변수에서 높은 위상값을 가진 변수로의 연결을 추가하면 사이클을 만들지 않지만 반대의 경우에는 사이클이 생긴다.

DIR은 원본 네트워크에 그대로 삽입할 수 있는 목표 네트워크의 에지를 말한다. REV는 reverse 연산을 해주어야만 하는 에지를 말한다. EQ는 에지를 구성하는 두 변수의 위상값이 동일한 경우로 원본 네트워크에 삽입할 경우 변수들의 위상값에 변화가 생겨 다시 계산해 주어야 한다.

알고리즘은 다음과 같은 6단계로 이루어진다. 1) 원본 네트워크의 변수들의 위상값을 계산한다. 2) 목표 네트워크의 에지들을 DIR, REV, EQ 그룹으로 분류한다. 3) REV에 있는 각 에지들에 대해 reverse 연산을 목표 네트워크에서 수행하고 새로 추가된 에지들을 위의 세 그룹으로 분류한다. 4) DIR에 있는 에지를 원본 네트워크에 추가한다. 5) EQ에 있는 각 에지들을 원본 네트워크에 추가하고 위상값을 다시 계산해 준다. 이 경우 EQ에 속한 에지 중 일부가 DIR로 바뀌게 된다. 6) 세 그룹에 있는 에지를 모두 처리하였으면 종료한다. REV와 EQ에 속한 에지들의 처리 우선순위는 [1]을 참조하였다.



(a) BN1 (b) BN2

(c) BN1, BN2 (d) BN2, BN1

그림 2. 결합 순서의 효과

그림 2는 결합 순서에 따라 결과가 달라지는 것을 보여준다. (a)와 (b)는 원본 네트워크와 목표 네트워크가 되고, (c)와 (d)는 결합 순서에 따른 결과를 보여준다. 베이지안 네트워크가 15개라면 결합 방법은 1.3×10^{12}가지가 존재한다. 모든 결합 순서에 대해 결합을 수행해 보고 가장 최소의 에지를 가지는 결합 순서를 찾는 것은 불가능하다. 왜냐하면 결합하는 과정에 많은 계산량이 소요되고 베이지안 네트워크의 개수가 추가되면 후보의 개수가 기하급수적으로 늘기 때문이다.

탐색해야 하는 해 공간에 대한 정보를 얻기 위해, 임의로 베이지안 네트워크를 생성하였다. 각 베이지안 네트워크의 변수는 3개에서 40개 사이에서 결정되었다. 변수들은 $X=\{x_1,x_2,\ldots,x_{50}\}$에 속한다. 베이지안 네트워크의 변수 수가 37로 결정되었다면 X에 속한 변수 중에서 37개가 선택된다. 만약 베이지안 네트워크의 개수가 9 이상이라면 모든 결합 순서에 대해 결과를 계산해서 해 공간을 그리는 것은 시간적인 문제로 불가능하다. 그림 3은 8개의 베이지안 네트워크를 결합하는 경우의 해공간을 보여주는데, 최적해는 16561 ~17281사이에 존재한다. x축은 8개 베이지안 네트워크의 permutation index를 나타낸다. $B_{12345678}$은 1이고 $B_{87654321}$은 마지막인 40,320이다 (그림 4).

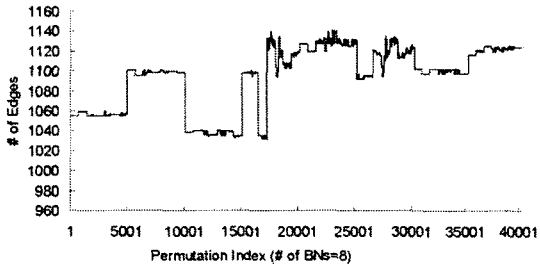


그림 3. 8개의 베이지안 네트워크를 결합한 경우

Permutation Index	결합 순서							
0	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	8	7
2	1	2	3	4	5	7	6	8
3	1	2	3	4	5	7	8	6
4	1	2	3	4	5	8	6	7
5	1	2	3	4	5	8	7	6
...	...	...	...	...	...	...	...	...
40318	8	7	6	5	4	3	1	2
40319	8	7	6	5	4	3	2	1

그림 4. 8개 BN에 대한 permutation index

```
1: /* N : # of Bayesian networks for combination */
2: /* POP: Population */
3: /* order[] : Array with length N */
4: /* Permutation(permutation index): return the order of
   N items given the inputted permutation index */
5: /* fitness(j) : fitness of jth individual */
6: /* Initialization(population): Each individual is initialized
   as a value from 0 to N!-1 */
7: /* θ: random variable */
8: Initialization(POP);
9: For i=1 to MAX_GEN {
10:    For j=1 to |POP| {
11:       order = Permutation(POP(j));
12:       fitness(j) = Fusion (order); }
13:    POP=Selection(POP, fitness); // selection
14:    For j=1 to |POP|/2// crossover
15:    {Parent(); // select two indexes
       POP(children)=
       θ×POP(parent1)+(1-θ)×POP(parent2);}
16:    For j=1 to |POP| // mutation
17:       POP(j)=Random(POP(j), 0, N!-1)
18:}
```

그림 5. 진화적 병합 알고리즘의 의사코드

그림 5는 진화적 병합 알고리즘의 의사코드를 보여준다. 진화 알고리즘은 permutation index를 염색체 표현으로 사용하였다. Permutation index를 염색체 표현으로 사용하면 교차나 돌연변이 연산의 구현이 상대적으로 쉬워진다. 또한 해공간이 permutation index를 기준으로 보았을 때 연속성이 있기 때문에 해탐색에도 유리하다.

3. 실험 및 결과

진화 알고리즘의 파라미터는 집단 크기 20, 최대 세대 50, 교차율 0.8, 돌연변이율 0.01로 하였다. 총 16개의 베이지안 네트워크를 임의로 생성하였다. 진화 알고리즘의 성능은 다음의 전략과 비교하였다.

- 휴리스틱#1: 에지의 수가 적을수록 결합 순서가 빠름
- 휴리스틱#2: 에지의 수가 많을수록 결합 순서가 빠름
- Greedy#1: 결합할 경우에 에지의 수가 최소화 되는 네트워크 순으로 정렬
- Greedy#2: 결합할 경우에 에지의 수가 최대화 되는 네트워크 순으로 정렬

표 1. 모든 결합 순서를 나열해서 성능을 평가한 경우 (숫자는 최종 병합 네트워크의 에지 수)

BN의 개수	결합 순서의 개수	평균	최대	최소
2	2	609.0±135.7645	705	513
3	6	917.5± 61.7243	994	855
4	24	926.1± 53.6475	995	858
5	120	970.5± 62.6244	1056	880
6	720	982.6± 61.3339	1065	883
7	5040	1056.0± 39.3754	1119	984
8	40320	1093.4± 31.0356	1140	1032

표 2. 전역해(에지의 수를 최소화하는 결합 순서)의 비율 및 나열식 평가 시간

BN의 개수	전역해의 개수	전역해의 비율	총 평가시간
2	1	50.0	-
3	2	33.3	-
4	6	25.0	-
5	18	15.0	-
6	90	12.5	-
7	5	0.1	1h 40m
8	96	0.23	22h

표 3. 각 결합전략별 성능 비교

BN의 개수	휴리스틱 #1	휴리스틱 #2	Greedy #1	Greedy #2	제안한 방법
2	513	705	513	705	513
3	855	989	855	904	855
4	858	990	990	905	858
5	880	1005	1041	932	880
6	883	1005	1046	933	883
7	1014	1093	1105	993	986
8	1055	1118	1126	1039	1034

표 1과 2는 BN의 개수를 2개부터 8개까지 바꾸어 가면서 가능한 모든 결합 순서에 대해 결합을 해보고 결과를 분석한 것이다. 표 3은 각 결합전략별 성능비교를 보여준다. 제안한 방법이 2개~8개의 네트워크를 결합할 때 가장 좋은 성능을 보여주고 있음을 보여준다. 그림 6은 상대적으로 가장 좋은 성능을 보여주었던 휴리스틱 #1과의 비교결과를 보여주고 있다. 제안한 방법이 8개 이상의 네트워크를 결합할 때에도 우월한 성능을 보임을 알 수 있다. 그림 7은 진화과정을 분석한 도표이다.

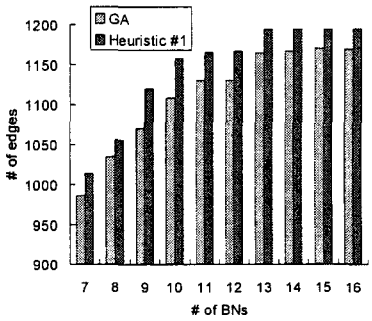


그림 6. 제안한 방법과 휴리스틱 #1과의 비교

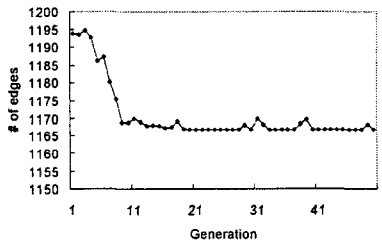


그림 7. 평균적합도의 변화 (네트워크의 개수=14)

4. 결론 및 향후연구

본 논문에서는 베이지안 네트워크를 결합하는 알고리즘이 결합 순서에 따라 서로 다른 결과를 내기 때문에 최종 결과를 최적화하는 결합 순서를 발견하는 유전자 알고리즘을 제안하였다. 해공간 분석을 통해 permutation index기반 진화 알고리즘을 제안하였고 본 방법이 기존의 전통적인 방법보다 우수한 성능을 보였다.

감사의 글

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(IITA-2005-(C1090-0501-0019))

참고문헌

[1] I. Matzkevich, and B. Abramson, "The topological fusion of Bayes nets," *Proceedings of the 8th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 191-198, 1992.

[2] I. Matzkevich, and B. Abramson, "Some complexity considerations in the combination of belief networks," *Proceedings of the 9thConference on Uncertainty in Artificial Intelligence*, pp. 159-165, 1993.