Journal of the Korean
Data & Information Science Society
2001, Vol. 12, No. 1, pp. 61 ~ 69

불완전한 지식에서 정리증명을 위한 확률추론

김진상 [1] · 신양규[2]

요 약

본 논문은 논리문장으로 표현된 지식을 처리하는 정리증명 과정에서 증명이 완료되기 전에 잠정적 결론을 유도하는 확률추론 기법을 제시한다. 정리증명 과정 중에 베이지안 해석을 이용하여 지식을 갱신하는 방법을 제시하고, 의사결정 방법을 사용하여 시간에 민감한 사안에 대해 신속하게 대처할 것인지 아니면 고의로 미룰 것인지를 결정하는 방법을 밝힌다.

주제어: 확률추론, 베이지안 분석, 의사결정, 불확실한 지식, 정리증명.

1. 서론

정리증명은 논리적이고 결정론적인 처리과정이다. 그러나 문제 해결에 사용될 지식과 풀이과정의 불확실성을 고려하게 되면 정리증명이나 혹은 수학 분야의 많은 내용에는 상당한 변화를 가져올 수 있다. 수학자 George Polya는 수학자들의 직관과 노력을 향상시키는 '근사추론'(plausible reasoning)의 중요성을 강조하였다. 특히 그는 수학의 정리증명에서 유사성과 불확실성의 중요함에 대해 다양하게 연구하였다[8,9]. 수학적 관점에서 참인 문장은 관련된 개념에 관한 기존 증명과 귀납적 사고의 결과에서 유도되는 경우가 많다. 예를 들어 논리적 지식과 귀납적 사고를 바탕으로 '페르마의 마지막 정리'[3]가 1994년 실제 그 증명이 발견되기 전까지 약 360년 동안 많은 수학자들은 그 정리를 참으로 믿고 있었다[11].

본 논문에서는 제한된 계산능력을 가지고 정리증명을 수행하는 동안 불확실한 지식의 분석을 이용한 근사추론의 특성을 형식화한다. 특히 정리증명 동안 탐색공간의 크기 정보와 목표를 향해 진행하는 정보를 이용하여 명제 문장의 진위에 대한 베이지안 해석 방법을 기술한다. 베이지안 해석은 최근 정보통신 분야의 지식표현, 근사추론, 기계학습, 정보추출 등 다양한 영역에서 중심기술로 활용되고 있다[1,5,6].

본 논문은 또한 의사결정론적 추론과 확률적 메타추론을 이용하여 다음에 취할 최선의 행동과 부분적 정보만으로 어떤 행동을 취하기 전에 신중하게 고려해야 할 사실들을 어떻게 결정하는지를 설명한다. 나아가 시간에 민감한 경우 의사결정 시스템이 얼마나 오랫동안 추론을 계속할 수 있는지를 결정한다고 할 때 정리증명 처리기를 위한 계산량의 기대치를 특성화시키는 방안을 제시하고자 한다.

2. 의사결정론적 메타추론

[1] 대구시 달서구 신당동 계명대학교 컴퓨터 전자공학부교수
[2] 경북 경산시 점촌동 경산대학교 정보과학부교수

[3] n이 2 보다 큰 자연수에 대해 방정식 $x^n + y^n = z^n$은 x, y, z 모두가 0이 아닌 정수해의 짝을 갖지 않는다.

복잡한 계산 절차에서 어떤 결과가 나올지를 미리 예측하는 것은 거의 대부분의 경우 불가능하지만 예상되는 결과에 대한 요약된 설명을 이용하여 계산 결과의 특징을 기술하는 것은 가능하다. 예를 들어, 질적인 측면에서 다양한 성질을 정의할 수 있을 것이며 자원의 분배를 통해 여러가지 측면들이 어떻게 세분화되는지 예측할 수 있다. 컴퓨터의 경우 계산 중인 결과를 세분하기 위해 할당할 시간과 기억장치의 질적인 면을 추론할 때 계산 결과에 관한 불확실성을 구체적으로 고려해야만 한다. 계산기대치 λ는 현재 계산 결과에 사용될 자원의 기대치와 추가로 자원을 할당했을 때 얻을 수 있는 계산 결과에 사용되는 자원의 기대치와의 차이를 나타낸다[10]. 전체계산기대치 Λ는 계산기대치와 지연된 행동에 관한 비용을 합한 것이다.

유연한 계산절차란 계산에 필요한 시간이나 기억장치와 같은 여러가지 자원의 할당이 증가할 때 계산 결과의 속성을 연속적으로 세분화할 수 있는 계산방법을 말한다[3]. 유연한 계산절차가 가지는 특징은 계산에 필요한 자원이 충분히 제공될 때 이상적인 결과에 수렴한다는 점이다. 이러한 특징은 추론 시스템에도 적용될 수 있으며, 이때 추론 시스템은 계산량이 증가될 수록 이상적인 결과에 수렴한다고 볼 수 있다.

유연한 계산절차는 주어진 문제 사례 I를 계산에 필요한 자원의 양 r만큼 사용하여 부분적 결과 $\delta(I)$로 변환시킨다. 부분적 결과에 관한 비용을 추론에 사용된 비용 u_i와 결과의 메타수준이 아닌 대상수준[4] 비용 u_o로 구분할 때, 문제 사례 I를 세분화하기 위해 유연한 계산 절차 S_i에 할당한 자원의 양 r에 관한 전체계산기대치 Λ는 다음과 같다.

$$\Lambda(S_i, I, r) = \int_{\delta(I)} u_o(\delta(I)) \times p(\delta(I)|S_i, I, r) - u_o(I) - u_i(r)$$

현재의 문제 사례 I는 미리 분배된 자원을 이용하여 유연한 계산절차에 의해 계산된 부분적 결과 $\delta^o(I)$일 수도 있다. 이때 어떤 부분적 결과 값의 차원이나 성질이 불확실할 수 있고, 대상수준의 자원을 부분적 결과의 성질들로 대응시키는데 사용된 함수 u_i와 u_o도 불확실할 수 있다. 이러한 경우를 해결하기 위한 방안은 위 식을 부분적 결과의 서로 다른 성질에 대한 불확실성과 함수들의 불확실성에 대해 모두 더해주면 된다.

3. 명제논리

여기서는 확률추론의 방법을 적용할 명제논리에 관해 간략히 언급한다[7]. 명제논리의 문법적 구성은 명제 알파벳에서 시작되고 이는 다음과 같이 구성된다: (1) 두 개의 괄호 '('와 ')', (2) 단위 문장을 나타내는 명제 변수들 $P, Q, R, \ldots$들의 집합, (3)논리 연결어들의 집합.

명제 논리에서의 식 (혹은 논리식)은 다음과 같이 정의한다: (1) 모든 명제는 식이다, (2) 만약 ϕ와 ψ가 식이면, $\neg\phi$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$ 그리고 $(\phi \leftrightarrow \psi)$ 들도 식이다, (3) 모든 식은 (1)과 (2)의 규칙을 유한번 적용시켜 만들어진 것이라야 한다.

주어진 명제 알파벳에 대해, 명제언어란 그 알파벳의 기호들로 만들 수 있는 가능한 모든 식들의 집합을 가리킨다. 해석 함수 (혹은 해석)은 각 명제에 진리값 하나를 부여하는 어떤 함수다. 이 함수의 정의역은 명제 집합이다. 진리표의 각 열은 하나의 해석이 되며 해석 중에서 참인 것을 그 식의 모델(model)이라 한다. 어떤 식 F가 가능한 모든 해석에서 항상 참일 때 그 식을 항진이다 (혹은 정당하다)고 하고, $\models F$로 나타낸다. 식 F가 거짓 (혹은 만족 불가능) 일 필요충분 조건은 그 식이 모든 가능한 해석에서 거짓일 경우다. 식 F가 어떤 해석 v에서 참인 경우 v는 F를 만족한다고 하고 v가 F의 모델이 된다.

리터럴(literal)은 하나의 단순 명제 혹은 그것의 부정이다. 두개의 리터럴 L과 $\neg L$를 서로 반대(opposite)라 한다. 또한 리터럴들이 논리합으로 연결된 형태 $L_1 \vee \ldots \vee L_n$을 절(clause)이

[4]대상수준(object level)이란 실제 추론 자체의 대상이 되는 지식 및 객체들의 차원을 가리키고, 메타수준(meta-level)이란 대상수준의 특징이나 성질을 언급하는 다른 차원을 가리킨다.

라 한다. 식 P가 $P_1 \wedge ... \wedge P_n (n \geq 1)$의 형태이고 각 P_i가 리터럴들의 논리합일 때 P를 논리곱 정규형이라고 한다. 또한 식 P가 $P_1 \vee ... \vee P_n (n \geq 1)$와 같은 형태이고 각 P_i가 리터럴들의 논리곱일 때 P를 논리합 정규형이라고 한다.

식 P가 식들의 집합 $\{Q_1, ..., Q_n\}$의 논리적 결론일 필요충분조건은 $Q_1 \wedge ... \wedge Q_n$을 참으로 하는 모든 해석 v가 P역시 참으로 할 때이다. 이 경우 $Q_1, ..., Q_n \models P$로 나타낸다.

4. 정리증명 방법

여기서는 제한된 자원을 가진 경우에 정리증명이 어떻게 이루어 지는지에 대해 살펴본다. 특히 명제논리로 표현되는 지식을 세분화하고 정리증명을 제어하기 위해 앞에서 언급한 유연한 계산절차와 부분적 결과에 초점을 맞춘다. 특히 정리증명의 경우 "전제로 주어지는 논리식의 집합이 결론에 해당하는 하나 혹은 여러 개의 논리식을 함축하는가?"에만 한정한다. 자동 정리증명 처리기는 바로 이러한 질문에 대한 답을 계산하는 시스템이라 볼 수 있다.

전제와 항진명제에서부터 무한히 많은 정리들이 연역될 수 있다. 따라서 전제에다 정당한 추론 규칙을 연속적으로 적용하여 결론을 유도해 낸다는 것은 매우 비현실적이다. 한 가지 효과적인 방법은 전제들의 논리곱 T와 결론의 부정 $\neg C$에서 모순을 유도하는 것인데, 이때 만약 모순을 발견하게 되면 C가 T에서 함축된다. 그런데 만약 전제들 자신이 모순이라면 임의의 문장이 함축될 수 있기 때문에 전제들의 집합이 모순이어서는 안된다.

비교흡수를 이용한 반증법과 같이 정리증명을 위한 다양한 방법들이 있지만 이들 방법은 증명 과정이 언제 끝날지에 관한 정보를 제공하지는 못한다. 임의의 단계에서 생성된 정리 집합의 크기도 집합 그 자신이 모순이 없다는 정보는 제공할 수 없다.

한편 Bibel이 제안한[2] 행렬 방법을 이용한 명제논리 증명 처리기는 문장들의 집합에 관한 일관성을 검사하기 위해 참과 거짓의 진리값을 배정하는 전체 상태 공간을 탐색하게 한다. 이 행렬 방법은 증명의 종료 시점에 관한 예상 거리 정보를 제공할 수 있다는 특징을 가지고 있다.

행렬 방법은 $(T \wedge \neg C)$ 내의 절들을 만족하는 진리값의 배정을 탐색함으로서 모순을 증명한다. 만약 참을 배정할 수 있는 가능성이 없다면 이것은 모순인 상황을 의미하므로 결론이 전제로부터 유도됨을 나타내는 것이다. 자세한 처리 과정은 다음과 같다: 우선 집합 $(T \wedge \neg C)$ 내의 모든 식들의 논리곱을 논리곱 정규형으로 변경시킨다. 이렇게 변경하고 나면 논리합들의 집합 $C_1, C_2, ..., C_n$이 생성되는데 이것은 각 집합이 묵시적인 $\wedge$로 연결된 것으로 보면 된다. 이 절들의 집합에서 경로 x란 리터럴들의 집합 $L_1, ..., L_n$이 되는데, 여기서 각 L_i는 논리합 C_i 내에 나타나는 것이어야 한다. 따라서 $L_i (1 \leq i \leq n)$는 어떤 단일 명제 P이거나 그것의 부정이 된다. 어떤 경로가 '열려있을' 필요충분 조건은 모든 $1 \leq i, j \leq n$에 대해 $L_i \neq \neg L_j$이다. 반면에 경로가 '닫혀있을' 필요충분 조건은 어떤 $1 \leq i, j \leq n$에 대해 $L_i = \neg L_j$ 이다.

진리값 배정 v가 경로 x에 있는 모든 리터럴을 만족할 필요충분 조건은 x가 열려있을 때이다. 식들의 집합 $(T \wedge \neg C)$이 절들의 집합 $C_1, C_2, ..., C_n$와 논리적으로 합동이므로 $(T \wedge \neg C)$을 만족하는 진리값 배정이 존재할 필요충분 조건은 절들을 따라가는 열린 경로가 존재할 때이다.

행렬 방법에서 열린 경로를 찾는 과정은 트리 구조에서 깊이우선 탐색을 행하여 해결할 수 있다. 이때 첫 번째 논리합의 리터럴들은 루트노드의 자식이되고, 임의의 레벨의 노드들은 다음 논리합의 리터럴들을 자식 노드로 갖는다. 자식노드를 향해 아래로 내려가는 탐색은 해당 경로에 모순이 발견되면 즉시 종료하며, 그 아래의 모든 부속 경로들도 취소한다. 이러한 탐색의 결과로 만약 모든 경로들이 모두 닫히게되면 참이되는 진리값의 배정이 없다는 뜻이므로 전제에서 결론이 유도됨을 알 수 있다.

행렬 방법을 이용한 정리증명 절차는 탐색해야 할 공간의 크기가 리터럴들의 수에 대한 절들의 수의 지수승으로 커진다는 점을 유의해야 한다. 즉, 행렬에서 n개의 절이 있고 각 절은 m개의 리터럴로 구성될 때 전체 m^n 개의 경로가 존재한다.

5. 확률과 증명

여기서는 정리증명과 확률 사이의 관계에 대해 논한다. 행렬을 이용한 정리증명은 문제에 나타난 리터럴들의 집합에다 2^k 가지의 진리값 배정 각각이 $(T \wedge \neg C)$의 원소들을 만족시킬 수 있는지를 검사하는 것과 같다. 행렬을 이용한 정리증명에서 탐색공간은 가능한 모델들의 공간과 동일하지는 않다.

절들 $C_1, C_2, ..., C_n$을 통한 각 부분 경로 x는 $1 \leq i \leq n$ 사이의 어떤 i에 대해 $L_i = T$이면 $v(T) = true$인 진리값 배정 집합 v의 합집합과 같다. 문법적인 경로들과 가능한 진리값 배정들 사이에는 일-대-다 대응관계가 성립하므로 문법적 경로 집합을 탐색하는 것은 진리값 배정과 같은 진리표 탐색 보다는 훨씬 빨리 끝낼 수 있다. 또한 특정 행렬을 이용해 주어진 길이의 경로 각각은 항상 같은 수의 서로 다른 리터럴들을 가지며 따라서 같은 수의 진리값 배정이 존재한다. 결과적으로 전체 탐색공간의 일부분에 대한 해석은 진리값 배정의 다양한 조합으로 생기는 공간에서 행렬 방법에서 탐색될 결로 공간으로 확장될 수 있다.

행렬을 이용한 정리증명 동안 n_j라는 번호를 가진 절에서 현재의 부분경로를 따라 가면서 리터럴 L_i를 추가할 때 만약 모순이 발생한다면 (즉, 이미 리터럴 $\neg L_i$가 나타난 경우) 이미 탐색한 초기 탐색공간에서 경로의 수 m^n은 m^{n-n_j} 만큼 더 증가하게 된다. 이렇게 증가된 수는 따라갈 부분경로를 포함하는 전체 경로의 수를 의미한다.

닫힌 경로가 나타나면 이를 기록해 두어야 하는데, 이때 닫힌 경로의 빈도 수를 방문한 전체 탐색공간에서의 함수로 나타낼 수 있다. 또한 이때 열린 경로가 발견되기 전에 탐색한 탐색공간에 관한 확률적 정보도 저장할 수 있다. 이러한 정보를 이용하여 증명이 완료되기 전에 진리값의 확률을 계산하고자 한다.

주어진 전제들이 있을 때 α를 "그 전제에서 어떤 결론이 유도된다"는 메타이론적 주장이라고 가정한다. 여기서 해결하고자 하는 것은 전체 탐색공간의 일부가 열린 경로 없이 탐색되었다는 정보를 이용하여 행렬 방법이 α의 진위를 결정할 확률이다. 이를 해결하기 위해 우선 S를 α에 대한 증명을 발견하지 못하고 탐색한 탐색공간의 부분을 나타낸다고 가정한다.

진행된 탐색 정보와 진리값에 관한 사전확률을 근거로 α가 참일 확률을 베이즈 정리로 계산할 수 있다. 즉,

$$p(\alpha|S,\sigma) = \frac{p(S|\alpha,\sigma)p(\alpha|\sigma)}{p(S|\alpha,\sigma)p(\alpha|\sigma) + p(S|\neg\alpha,\sigma)p(\neg\alpha|\sigma)} \tag{1}$$

여기서 $p(\alpha|S,\sigma)$는 탐색공간의 S가 열린경로를 발견하지 못한채 탐색을 하였다는 사실과 상황 σ에 관한 이면 정보가 주어질 때 메타이론적 주장 α가 참일 확률이다. 또한 $p(S|\alpha,\sigma)$는 α가 참일 때 S의 확률이고, $p(S|\neg\alpha,\sigma)$는 α가 거짓일 때 S의 확률이며, $p(\alpha|\sigma)$는 α가 참일 사전확률이다. 메타이론적 주장 α의 사전확률은 입력의 크기나 구조와 같은 정보를 기반으로 정리증명 처리기에 주어지는 질문의 집합을 토대로 계산한다. α가 거짓일 확률 $p(\neg\alpha|\sigma)$은 α의 사전확률의 보수로 계산한다.

만약 α가 참이고, 열린경로가 발견되지 않았으며, 정리증명 처리기가 전체 탐색공간을 탐색하기 전에 멈춘다면 식 (1)을 더욱 간략히 할 수 있다. 즉 탐색의 모든 부분들이 1보다 작으므로 $p(S|\alpha,\sigma) = 1$이다. 따라서 $p(\alpha|S,\sigma)$는 $p(\alpha|\sigma)$와 $p(S|\neg\alpha,\sigma)$을 이용하여 표현할 수 있게 된다.

$$p(\alpha|S,\sigma) = \frac{p(\alpha|\sigma)}{p(\alpha|\sigma) + p(S|\neg\alpha,\sigma)(1 - p(\alpha|\sigma))} \tag{2}$$

따라서 α의 진리값에 대한 사전확률을 알고 α가 실제로 거짓일 때 α가 거짓이라는 증명을 찾지 못하면서 탐색하는 탐색공간의 부분이 증가될 확률을 안다면 정리증명 처리기가 멈추기 전에 어떤 명제의 진리값을 계산할 수 있게된다.

확률분포 $p(S|\neg\alpha,\sigma)$의 예상 형태가 문제로 남는다. 주어진 문제에 관한 정보가 없다면 문제에 나타난 리터럴들의 집합에 대한 가능한 진리값 배정 각각이 $(T \wedge \neg C)$의 원소들을 만족시킬 확률이 동등하다고 가정한다. 이러한 가정은 2^k 개의 가능한 진리값 배정 각각이 모두 $\neg\alpha$와 모순되지 않을 것이란 사실을 나타낸다. 탐색공간의 각 경로가 서로 독립이라고 가정하고 행렬에는 l개의 열린경로가 있다고 가정한다. 그러면 전체 탐색공간의 s 부분만큼 탐색한 후 $\neg\alpha$의 증명을 찾지 못할 확률은

$$p(S = s|\neg\alpha,\sigma) \approx \prod_{i=0}^{sm^n-1} 1 - \frac{l}{m^n - i} \tag{3}$$

여기서 m^n은 행렬에 존재하는 경로의 전체 수를 나타낸다.

이 논문에서는 열린경로들의 예상 수 l를 이용하여 실제 계산하는 과정을 설명하지는 않는다. 대신 여기서는 $p(S|\neg\alpha,\sigma)$에 관한 자료가 어떻게 이용될 수 있는지에 대해서만 언급하기로 한다. 그렇지만 일반적인 경우에는 문제 사례의 특성에 관한 자료에서 열린경로의 수에 관한 확률분포를 진리값 배정 공간의 크기에 관한 함수로서 정할 수 있다.

절들의 수와 길이, 그리고 알파벳의 크기가 열린경로의 수에 영향을 미친다. 열린경로의 수에 관한 불확실성은 문제 사례들의 속성 I_i에 대해 열린경로의 수에 관한 정보를 나타내는 확률분포 $p(l|I_1, I_2, ..., I_n, \sigma)$를 이용해 식 (3)을 수정하여 설명할 수 있다. 트리에서 경로들은 조상 노드들을 공유할 수 있기 때문에 행렬에서 모든 경로들은 서로 독립적일 수는 없다. 그러나 만약 대량의 경로가 주어진다면 리터럴들의 집합에 관한 경로들을 단일 경로로 치환함으로서 최소한의 종속성을 유지할 수 있다.

6. 증명 전의 행동

지금까지 살펴본 문장의 진위에 관한 지식을 이용하여 증명에 관한 행동와 연결시키기로 한다. 우선 에이전트가 취할 행동의 기대치는 행렬방법을 이용하여 증명할 수 있는 문장의 진리값에 의존한다고 가정한다. 열린 경로가 존재할 확률을 에이전트가 추론하면 논리적 해석이 끝날 때 까지 기다리지 않고서도 그 에이전트는 정리증명의 부분적 결과만을 이용하여 명제논리 지식베이스에서 행동을 취할 수 있다.

6.1 마감시간에서 즉각적인 행동

정리증명 처리기를 가진 에이전트는 기대하는 효용성을 극대화 시킬수 있도록 행동해야 한다. 하나 이상의 명제 식들의 진리값의 가능성을 바탕으로 한 서로 다른 행동들 A_i의 기대치를 계산하기 위해 모든 행동 A_i와 식 α_j에 대해 행동 결과의 효용성 $u(A_i, \alpha_j)$와 $u(A_i, \neg\alpha_j)$를 고려한다. 그런 다음 효용성의 기대치 U를 극대화 시킬 수 있는 행동을 선택한다. 이를 위해 행동 A_i를 행할 때 U는 다수 개의 식 α_j의 진리값에 따라 달라지며 이들은 서로 배타적이라고 가정한다. 이 경우에 행동 A_i를 취할 효용성의 기대치 $U(A_i)$는 다음과 같다.

$$U(A_i) = \sum_j p(\alpha_j|S,\sigma)u(A_i,\alpha_j) \tag{4}$$

그림 1: 즉각적인 행동 대 계속적인 추론에 관한 의사결정

최선의 행동이 어떤 식 α의 진리값에 관한 지식에 의해 결정된다면, 최선의 행동 A^*는 다음과 같을 것이다.

$$A^* = argmax_A[p(\alpha|S,\sigma)(u(A_i,\alpha) - u(A_i,\neg\alpha)) + u(A_i,\neg\alpha)] \tag{5}$$

여기서 $argmax_A\Phi(A_i)$는 $\Phi(A_i)$ 값이 최대가 되는 A의 원소 A_i를 나타낸다.

예를 들어, 두 개의 행동 A_1과 A_2가 각각 "택시를 탄다"와 "버스를 탄다"일 때 확률 $p(\alpha|S,\sigma)$와 임계확률 p^* 사이의 관계를 고려하여 더 좋은 행동을 결정할 수 있다. 여기서 임계확률 p^*는 두 개의 행동이 동일한 효용성의 기대치를 가진다고 할 때 식 α의 진리값의 확률이고 이는 다음과 같다.

$$p^* = \frac{u(A_2,\neg\alpha) - u(A_1,\neg\alpha)}{u(A_2,\neg\alpha) - u(A_1,\neg\alpha) + u(A_1,\alpha) - u(A_2,\alpha)} \tag{6}$$

이와 같이 둘 중의 하나를 선택해야 하는 의사결정에서는 $p(\alpha|S,\sigma)$이 p^*보다 크다면 행동 A_1이 더 좋은 것이고, 그렇지 않다면 A_2가 더 나은 행동이 된다.

6.2 즉각적인 행동 대 추론

여기서는 행렬 방법을 이용한 논리추론 계산의 기대치 $\tilde{\Lambda}$을 유도한다. $\tilde{\Lambda}$은 행동을 미룰 때 소요되는 비용과 더 나은 의사결정을 얻기 위해 추가 계산을 할 것인가에 관한 상호간의 반대급부 정보를 얻을 수 있다. 우선 탐색공간 S의 일부인 s를 증명을 찾지 못한채 탐색하였다고 가정한다. 이때 추가로 j개의 경로를 더 탐색할 때 소요되는 비용을 계산하고자 한다.

그림 1은 즉각적인 행동 대 추가로 더 심사숙고하기 위해 행동을 미루는 경우에 관한 의사결정 트리를 나타낸다. 만약 정리증명 처리기가 행렬 방법에서 추가로 x개의 경로를 더 조사할 수 있다면 두 가지 결과가 나올 수 있다. 첫째는 시스템이 α의 증명을 찾은 후 멈추는 경우이고, 둘째는 멈추기 않고 계속 진행하는 경우다. 만약 정리증명 처리기가 멈추지 않는다면 S를 S'로 갱신하고 α의 진리값에 관한 지식 $p(\alpha|S',\sigma)$을 식 (2)에 의해 변경한다.

시간에 민감한 상황에서 한 가지 이상의 결과에 대한 효용성은 행동이 취해지기 전에 지연되는 시간의 길이에 종속된다. 여기서는 시간에 종속된 효용성을 행동이 일어나기 전에 지연된 시간 t를 고려한 효용성의 변화까지 감안하여 식 (5)에서 사용한 결과의 효용성 표현을 확장한 $u(A_i,\sigma_j,t)$로 나타낸다.

증명을 찾지 못하고 추가로 j개의 경로를 더 탐색한 경우 효용성의 기대치를 계산한다. 확률 $p(\alpha|S,j,\sigma)$은 탐색공간 S의 일부를 이전에 탐색한 후에도 증명을 찾지 못하고 추가로 j개의 경로를 탐색할 때 α의 확률을 나타낸다. 또한 $t(j)$는 j개의 경로를 검색하는데 소요된 시간의 양을 나타낸다고 가정한다. 먼저 증명을 찾지 못한채 j개의 경로를 탐색한 후 즉각적인 행동을 할 기대치는 다음과 같다. 만약 $V(S|j)$가 증명을 찾지 못한채 j개의 경로를 추가로 탐색한 후 최선의 행동을 실행할 때 기대할 수 있는 효용성의 기대치라면,

$$V(S,j) = max_A[p(\alpha|S,j,\sigma)(u[A_i,\alpha,t(j)] - u[A_i,\neg\alpha,t(j))]) + u[A_i,\neg\alpha,t(j))])] \tag{7}$$

와 같다. 이때 함수 $max_A\Phi(A_i)$는 A의 원소 A_i에서 $\Phi(A_i)$의 최대값을 나타낸다.

우선 하나의 경로를 더 추론할 것인지 결정하기 위해 $\tilde{\Lambda}$를 구한다. 그러기 위해서는 정리증명 처리기가 다음에 찾을 경로 중 열려 있으면서 멈출 경로를 찾을 확률과 멈추지 않을 확률을 알아야 한다. 확률 $p(H|S,j,\sigma)$를 j번째 추가로 탐색한 경로 상에서 멈춘 후 α가 거짓이라고 결정할 확률이라고 둔다. 그렇다면 하나의 경로에 대한 $\tilde{\Lambda}$은 다음과 같다.

$$\tilde{\Lambda}(S,1) = p(H|S,1,\sigma)max_A u[A_i,\neg\alpha,t(1)] + [1 - p(H|S,1,\sigma)]V(S,1) - V(S,0) \quad (8)$$

이때 $V(S,0)$는 추가 탐색을 하는 대신 즉각적인 행동을 취할 때 기대되는 효용성을 가리킨다.

시간에 관한 비용함수가 주어질 때 한 단계만의 분석으로는 더 오랫동안 계산하는 것이 좋을지 혹은 그렇지 않을지 판단하는 것이 불가능하다. 따라서 여러 개의 경로를 탐색하는데 필요한 $\tilde{\Lambda}$를 구할 필요가 있다. 남은 탐색공간 내에서 앞으로 탐색할 임의의 수의 경로 x에 대한 $\tilde{\Lambda}$에 대한 공식은 x개의 모든 경로들이 탐색되기 전에 각각 서로 다른 시간에 시스템이 종료할 확률로 표현된다.

$$\tilde{\Lambda}(S,x) = \sum_{j=1}^{x} p(H|S,j,\sigma)max_A u[A_i,\neg\alpha,t(j)] + [1 - \sum_{j=1}^{x} p(H|S,j,\sigma)]V(S,j) - V(S,0) \quad (9)$$

두 개 중 하나를 선택해야 하는 의사결정의 경우 식 (9)에서 최대값 계산은 확률$p(\alpha|S,\sigma)$가 p^*보다 큰지만 비교하면 된다.

식 (8)과 (9)를 풀기위해 필요한 확률분포는 앞에서 설명한 정리증명 처리기의 성능에 관해 수집된 자료에서 바로 구할 수 있다. 그러나 식 (3)에서 설명한 대로 자료에서 나타난 관계들을 설명할 수 있는 확률 모델을 사용할 수 도 있다. 특히 j번째 새로 탐색되는 경로 상에서 멈출 확률의 근사치를 구하는 것도 가능하다. 만약 α가 참이라면 탐색이 종료되기 전에 j번째 탐색 경로에 추가될 어떤 가지에서도 확률 $p(H|\alpha,S,j,\sigma)$은 0이 된다. 왜냐하면 α가 참이면 정리증명 처리기는 전체 탐색공간을 완전히 탐색하기 전까지는 멈추지 않기 때문이다. 따라서 α가 거짓일 때 정리증명 처리기가 멈출 확률만 다음과 같이 구하면 된다.

$$p(H|S,j,\sigma) = p(H|\neg\alpha,S,j,\sigma)p(\neg\alpha|S,j,\sigma) \quad (10)$$

만약 식 (3)의 확률 모델을 구할 때 가정했던 독립성을 적용한다면 식 (10)의 첫째 항은 다음과 같은 근사치를 가지게 된다.

$$p(H|\neg\alpha,S,j,\sigma) \approx \left[\frac{l}{r-(j+1)}\right] \sum_{i=0}^{j-2} 1 - \frac{l}{r-i} \quad (11)$$

단, 위 식 (11)은 j개의 경로를 추가로 탐색한다고 가정하고 전체 탐색공간 중에서 r개의 경로는 탐색하지 않는 경우에 해당한다. 확률 $p(\neg\alpha|S,j,\sigma)$는 식 (2)에서 설명한 대로 베이즈 정리로 계산할 수 있다.

7. 결론

이 논문에서는 유연한 계산과 한정된 자원에서 지식과 행동을 처리할 때 세밀하게 고려해야 할 정도를 제어하는 방법을 제시하였다. 정리증명 처리기의 관점에서 정리증명 활동의 효용성에 초점을 맞추었으며 환경에 대한 행동을 어떻게 취할지에 한정하여 의사결정 방법을 기술하였다. 이 논문은 불확실한 지식을 가진 상황에서 확률을 이용한 추론 방법과 제한된 계산 자원을 가진 정리증명 처리기 사이의 연결에 이론적인 발판이 될 수 있을 것

으로 본다. 추후 연구과제로는 일차논리를 위한 정리증명 처리기에도 본 논문에서 제시한 명제논리의 확률적 기법이 확장되어 적용되어야 할 것이다. 특히 확률적 기법은 일차논리 증명처리기가 증명을 하는 동안 생성되는 탐색공간의 크기를 유도하는데 효율적인 도구가 될 것으로 본다.

참 고 문 헌

1. Alsinet, T. and Godo, L. and Sandri, S. (1999). On the semantics and automated deduction for PLFC, a logic of probabilistic uncertainty and fuzziness. In Proceedings of Fifteenth Conference on Uncertainty in Artificial Intelligence.

2. Bibel, W. (1987). Automated Theorem Proving. Braunschweig: Vieweg, Reading, Massachusetts.

3. Dean, T. and Wellman, M. (1991). Planning and Control. pages 353–363. Morgan Kaufmann Publishers, San Mateo, California.

4. Doyle, J. (1990). Rationality and its roles in reasoning. In Proceedings of the Eigth National Conference on Artificial Intelligence, pages 1093–1100. American Association for Artificial Intelligence, AAAI Press / The MIT Press, Cambridge, MA.

5. Heckerman, D. *et al* (2000). Dependency networks for inference, collaborative filtering, and data visualization, Journal of Machine learning research, pages 49–75.

6. Monti, S. and Cooper, G. (1998). A Bayesian network classifier that combines finite mixture model and a naive Bayes model, Technical report ISSP-98-01, University of Pittsburgh.

7. Nilsson, N. (1998). Artificial Intelligence, chapter 13 Propositional Calculus, pages 217–230. Morgan Kaufmann Publishers, San Mateo, California.

8. Polya, G. (1954). Mathematics and Plausible Reasoning: Induction and analogy in mathematics, vol. 1. Princeton University Press, Princeton, New Jersey.

9. Polya, G. (1954). Mathematics and Plausible Reasoning: Patterns of Plausible Inference, vol. 2. Princeton University Press, Princeton, New Jersey.

10. Russell, S. (1990). Fine-grained decision-theoretic search control. In Proceedings of Sixth Conference on Uncertainty in Artificial Intelligence, Cambridge, MA. Association for Uncertainty in Artificial Intelligence, Mountain View, CA.

11. van der Poorten, A. (1995). Notes on Fermat's Last Theorem. Wiley Inter-science, New York.

A Probabilistic Reasoning in Incomplete Knowledge for Theorem Proving

Jinsang Kim[5] · Yangkyu Shin[6]

Abstract

We present a probabilistic reasoning method for inferring knowledge about mathematical truth before an automated theorem prover completes a proof. We use a Bayesian analysis to update beleif in truth, given theorem-proving progress, and show how decision-theoretic methods can be used to determine the value of continuing to deliberate versus taking immediate action in time-critical situations.

Key Words and Phrases: Probabilistic reasoning, Bayesian analysis, Decision making, Uncertain knowledge, Theorem proving.

[5]Faculty of Computer and Electronics Engineering, Keimyung University, Taegu, Korea
[6]School of Information Science, Kyungsan University, Kyungpook, Korea