

에이전트를 이용한 자동화된 협상에서의 전략수립에 관한 연구

The Strategy Making Process For Automated Negotiation System Using Agents

전진*, 박세진**, 김성식**

*고려대학교 부설 정보통신기술공동연구소

**고려대학교 산업공학과

Abstract

Due to recent growing interest in autonomous software agents and their potential application in areas such as electronic commerce, the autonomous negotiation become more important. Evidence from both theoretical analysis and observations of human interactions suggests that if decision makers have prior information on opponents and furthermore learn the behaviors of other agents from interaction, the overall payoff would increase. We propose a new methodology for a strategy finding process using data mining in autonomous negotiation system: ANSIA(Autonomous Negotiation System using Intelligent Agent). ANSIA is a strategy based negotiation system. The framework of ANSIA is composed of following component layers : 1) search agent layer, 2) data mining agent layer and 3) negotiation agent layer. In the data mining agent layer, that plays a key role as a system engine, extracts strategy from the historic negotiation is extracted by competitive learning in neural network. In negotiation agent layer, we propose the autonomous negotiation process model that enables to estimate the strategy of opponent and achieve interactive settlement of negotiation. ANSIA is motivated by providing a computational framework for negotiation and by defining a strategy finding model with an autonomous negotiation process.

1. 서론

1.1 연구 배경

협상(negotiation)과 그 결과로서의 거래(bargaining)는 경제활동에 있어서 가장 기본적인 행위이다(Lewicki[6]).

일반적으로 제품의 가격이 거래 가격이 낮거나, 수량이 적은 경우는 우리는 확정가(fixed price)를 가지고 거래를 한다. 하지만 높은 가격이나 다량의 거래 시에는 협상(negotiation)이라는 문제가 대두되게 된다.

하지만 인간의 협상과정에서는 흔히 오류, 또는 견해차이(conflict)가 존재하게 되며, 이에 대한 극복을 위하여 여러 분야에서 협상과정에 대한 모델링(modeling)과 정량화(quantitative research)가 이루어지고 있다.

Fisher[4]에 따르면 일반적인 인간의 협상 과정에서는 다음과 같은 3가지 오류가 존재한다.

- 감정(emotions)
- 잘못된 추론(mis-perception)
- 의사소통부족(lack of communication)

위와 같은 오류로 인해 발생하는 협상과정문제는 크게 다음과 같은 2가지로 나누어진다.

- 협상행위의 복잡성 (complexity of activity)
- 선호구조 파악의 난해함 (difficulty of understanding preference structure)

여기서 협상행위의 복잡성이란 인간의 협상행위는 협상에서 나타나는 수많은 요소에 대한 고려를 파악이 어려워, 결과적으로 협상에서의 타결(settlement)이 어렵고, 보다 나은 결과를 고려하지 못한다는 점이다.

예를 들어 Rangaswamy[10]의 연구에 따르면 4가지 고려사항(issue)이 각각 4가지의 대안을 가지고 있을 경우에는 $4^4 = 256$ 개의 가능한 결과가 생성이 되어 결과적으로 34번의 협상 중 4건만이 타결이

된다는 결과를 보여주고 있다.

두 번째로 선호구조 파악의 난해함이란 협상에서 보다 나은 결과를 낳기 위해서는 자신의 선호구조 뿐만 아니라 상대방의 선호구조를 파악하여야 하는 문제가 대두되는데, 인간의 협상에 있어서는 주관적인 추론이 객관적이고 논리적으로 상대방의 선호구조의 계산을 방해한다는 점이다.

본 논문에서는 이러한 오류를 극복하고 객관적인 자료에 의해 협상을 이끌어 나가기 위해서 정성적(qualitative)인 과거 협상 데이터를 분석하여 정량적인 협상 전략으로 바꾸어 이를 협상 지식으로 활용하는 모델을 제시한다.

1.2 기존 연구 고찰

자동협상모델에 관한 관련 연구로는 크게 2가지 범주의 연구가 진행되고있다. 첫 번째 범주는 협상을 포함한 온라인 구매모델에 관한 연구이고, 두 번째 범주는 협상을 위한 상대방의 이익구조에 관한 분석에 관한 연구가 있다. 이러한 온라인 구매모델을 구현한 모델에는 Maes[7]의 *Tête-à-Tête*, Chavez[3]의 *Kasbah*등이 있고, 협상을 위한 상대방의 이익구조에 관한 연구는 Richard[11]의 연구와 Zeng[14]의 *Bazaar*가 있는데 이중 *Tête-à-Tête*, Richard[11]의 연구, *Bazaar*에 대하여 간략히 설명하고 그 한계를 살펴보자.

1.2.1 온라인 구매모델

온라인 구매모델(online shopping framework)에 관한 연구는 기존의 상업활동을 온라인 상에서 진행할 경우에 필요한 활동과 이중 협상이 어떠한 형태로 진행되어야 하며, 협상 이전에 선행되어야 하는 조건에 관한 연구이다.

Terpsidis[12]에 따르면 일반적인 온라인 구매모델은 다음과 같이 일반적으로 6단계로 나누어진다.

① 구매제품의 확정(Need Identification) : 이 단계는 구매자가 판매자가 제시한 여러 종류의 제품을 서로 비교하거나 시뮬레이션을 통하여 자신이 구매할 제품을 확정하는 단계이다.

② 제품 중계(Product Brokering) : 이 단계는 구매자가 확정한 제품에 대하여 여러 판매자에게 정보를 요청·수집하는 단계로서, 결과로 제품에 대한 구매후보 셋(considerable set)을 예측하게 된다.

③ 판매자 중계(Merchant Brokering) : 이 단계는 구매후보 셋에 들어 있는 제품을 판매하는 판매자에 대한 정보를 수집하여 구매자에게 알려주는 단계이다. 이 단계에서 구매자가 제시한 조건에 따라 구매자를 필터링(filtering)하기도 한다.

④ 협상(Negotiation) : 이 단계는 제품과 판매자가 확정되었을 때 어떠한 방법으로 매매를 할 것인가에 관한 문제이다.

⑤ 구매와 배송(Purchase and Delivery) : 이 단계는 협상종료의 신호에 따라서 제품 가격에 대한 지불이나, 소비자가 원하는 형태로의 배송 문제가 있다.

⑥ 사후관리와 평가(Product Service and Evaluation) : 이 단계는 제품·구매자에 대한 사후관리와 모든 구매처리 과정에 대한 평가를 내리는 과정이다.

Tête-à-Tête

Maes[7]가 제안한 대표적 온라인 쇼핑 프레임워크로, Tête-à-Tête는 판매자와 구매자 모두 각자의 이익을 협동적으로 구현하는 것을 목표로 하고 있다. Tête-à-Tête의 특징은 다음과 같다.

· 통합적인 온라인 쇼핑 프레임워크이다.

Tête-à-Tête는 위의 온라인 구매모델 중 ①~④부분을 통합하여 다룬 모델로서 온라인 상에서 판매하는 여러 쇼핑물의 가격을 비교한다거나, 경매(auction)형태를 띄는 다른 구매모델들과는 다르게 XML(eXtensible Markup Language)를 사용하여 통합적인 협상언어(integrative negotiation protocol)를 사용함으로써 다수의 온라인 판매자와 구매자가 공통된 탐색공간 및 협상언어를 갖게 하였다.

· Bilateral Argumentation

Tête-à-Tête는 구매제품과 판매자들이 정해진 경우에 양쪽 에이전트의 협상의 방법으로 bilateral argumentation형태의 협상방법을 취하고 있다.

구매 에이전트는 요청한 제품에 대하여 판매 에이전트들이 가장 가까운 제품에 대한 특성 및 사양에 대해 전해주는 제안(proposal)들에 대하여 구매자의 선호도와 비교를 하여 평가를 하게 된다.

협상은 이러한 선호도의 변화를 따라서 진행되게 되며, 이러한 선호도의 변화는 평가(critique)의 형태로 다시 판매 에이전트들에게 전해지게 된다. 평가의 결과를 보고 다시 구매 에이전트들은 수정된 제안(counter-proposal) 또는 거부(withdraw-proposal)를 하는 형태로 협상이 진행된다.

· 제품·판매자 중계 의사결정 시스템

Tête-à-Tête시스템 내부에는 구매자의 제품선택의 난해함을 도와주기 위한 다중-속성 유틸리티 함수(multi-attribute utility theory)와 "stereotyping"이라고 불리는 보완적 실행 촉진 기능(complementary bootstrapping mechanism)을 수

행하는 의사결정 엔진을 탑재하고 있어 구매자가 자신의 이익구조를 가지고 쉽게 제품이나 판매자를 선택할 수 있도록 해주는 기능을 가지고 있다.

하지만 이러한 많은 특징에도 불구하고 Tête-à-Tête는 여전히 사용자가 끊임없이 자신의 이익구조를 고쳐 나가거나, 새로운 제안을 하여야 하는 수동적이라는 문제점을 갖는다.

1.2.2 상대방의 이익구조에 관한 연구

Back-Propagation

Richard[11]은 구매와 판매의 형태(buy/sell pattern)인식의 방법으로서 신경망 이론 중 역전파 알고리즘을 이용하여 구매와 판매에 영향을 끼치는 요소를 18개의 기본 변수(fundamental variable)로 사용하여 판매결과 변수와의 연관관계를 해석하는데 사용하였다. 이 연구는 이익구조에 미치는 속성들을 패턴화 하여 판매/구매와의 연관성을 찾아보았다는 의의를 가지고 있다.

Bazaar

Zeng[14]은 협상에서 가장 중요한 요소는 상대방의 내정가(Reservation Price, 이하 RP)라고 가정하였다. 여기서 내정가란 판매자의 판매 하한가와 구매자의 구매 상한가를 의미한다.

Bazaar모델에서의 협상의 초점은 바로 상대방의 내정가를 예측하는 과정이다. 이는 상대방의 내정가를 알 수 있다면, 내정가에 근접하여 제안(offer)을 함으로서, 보다 나의 이익(payoff)을 증가시키려는 방법이다.

Zeng[14]은 이러한 상대방의 내정가를 학습(learning)하는 과정을 "Bayesian belief updating process"라고 정의하고 사후확률을 이용하여 상대방의 RP값을 계산하였으며, 이를 이용한 협상과정을 포함하여 Bazaar라고 명명하였다.

하지만 현실적으로 사후확률을 얻는 것이 불가능하며, 상대방의 RP예측치에 관한 근거가 존재하지 않아 실제적으로 모델을 사용하기는 적절하지 않다.

따라서 본 논문에서는 위에서 제기된 협상자동화의 문제를 극복하기 위해 어떠한 지식구조가 존재하여야 하며, 이러한 지식을 가지고 어떻게 상대방의 이익구조를 예측해 나가며 협상을 이끌어 나갈 것인가에 관하여 논의를 한다.

1.3 논문의 구성

본 논문의 구성은 다음과 같다. 2장에서는 에이전트를 이용한 자동협상모델에 관한 프레임워크와 각각의 하부모듈을 나누어 설명하며, 특히 에이전트의 지식발견 모델에 관하여 자세히 설명한다. 3장에서는 다양한 척도로 자동협상모델을 위한 지식발견 모델의 수행도를 평가하며 마지막으로 4장에서는 본 논문의 의의를 제시한다.

2. 본론

2.1 에이전트를 이용한 자동 협상 모델

전장에서 언급한 바와 같이 일반적인 인간의 협상 과정에서는 빠지기 쉬운 오류가 존재한다. 이러한 오류를 극복하고 객관적인 자료에 의해 협상을 이끌어 나가는 자동화된 모형의 제시 협상에서의 자동화를 구현하기 위해서는 인간의 지식을 모방할

수 있는 에이전트의 지식을 구축하는 과정과 전체 시스템에 관한 논의가 필요하다. 이를 위해 본 논문에서는 에이전트의 지식 구축 과정으로서의 협상 전략 발견 모델과 이를 이용한 에이전트를 이용한 자동협상 모델(Autonomous Negotiation System using Intelligent Agent:이하 ANSIA)을 제안한다.

2.1.1 ANSIA의 구조

ANSIA는 그림 1과 같이 3계층의 구조를 가지고 있다.

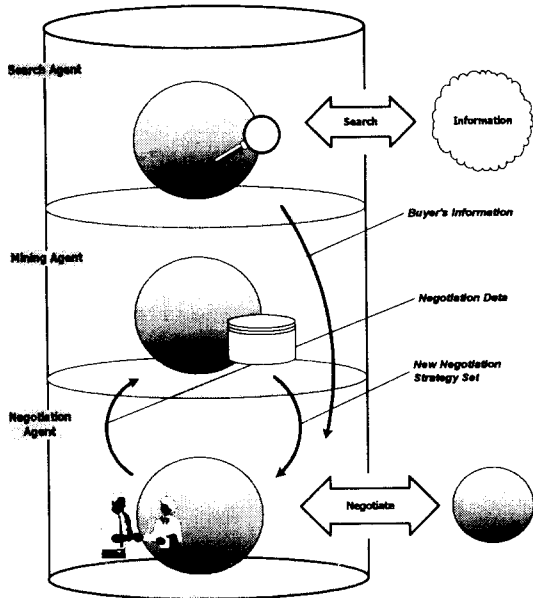


그림 1. 에이전트를 이용한 자동협상모델

(1) 검색 에이전트 계층

협상을 하기 위한 가장 기초 데이터를 수집하는 에이전트로 제품에 대한 고시가격(posted price: 판매자가 가장 초기에 제품에 가격으로서 고시한 가격으로 협상의 여부로서 값의 하락이 일어날 수 있는 가격)을 검색하여, 고시가격 중 가장 낮은 가격과 판매자를 선택하여 하위 에이전트인 마이닝 에이전트에게 넘겨주는 역할을 수행한다.

(2) 지식발견 에이전트 계층

과거 협상 데이터로부터 협상 속성에 따른 협상 전략을 찾아내어 협상 에이전트의 협상 전략 공간(negotiation strategy dimension)구성을 수행한다. 지식발견 에이전트 계층에서 사용되는 방법론으로 본 논문은 협상전략 발견 모델을 제안한다.

(3) 협상 에이전트 계층

실제로 판매자 또는 구매자 에이전트와 협상을 진행하는 에이전트로 검색에이전트가 넘겨주는 협상에 관한 정보와 협상 속성에 맞는 협상 전략 셋(negotiation strategy set)을 협상 전략 공간으로부터 가져와 전략에 따라 판매자와 협상을 하는 역할을 수행한다.

2.1.2 가정 및 시스템의 범위

(1) 시스템 범위

전체 시스템의 관점은 구매자의 시점에서 문제를 모델링 한다. 하지만 판매자의 입장에서 본다면 본 시스템의 프로세스는 변하지 않는다.

· 데이터 마이닝

본 논문에서는 데이터 마이닝의 범주 중 데이터의 속성(attribute)사이의 상관관계(correlation)를 발견하기 위하여 군집 모델을 사용하며, 상관관계를 바탕으로 의사결정트리(decision tree)를 구성하기 위하여 분류 모델을 사용하기로 한다. (표 1)

사용범위	데이터의 분류	의사결정트리 구축
데이터마이닝의 범주	군집모형	분류모형
학습형태	자율학습	지도학습
기타	설명모델	예측모델

표 1. 본 논문에서 사용한 데이터 마이닝 기법

· 신경망 이론

데이터의 속성 중 상관관계를 발견하고, 과거 협상데이터의 패턴화를 위하여 신경망 이론 중 자율 학습방법으로 경쟁학습(competitive learning)방법을 사용하도록 한다.

· 전자상거래

본 논문에서 제안하고 있는 자동협상모델은 Jim[5]이 제안한 전자상거래의 프레임워크를 따르면 다음과 같이 분류된다.

① 협상유무: 협상을 하지 않는 바자(bazaar)와는 다른 협상(negotiation)모델

② 상대방의 수: 복수의 상대방(multilateral)이 아닌 단수의 상대방(bilateral) 모델

③ 논의의 대상 수: 논의의 대상이 복수(multi-issue)가 아닌 대상을 가격만으로 한정된 모델(single issue)

④ 의사결정 주체: 모델이 사용자의 의사결정을 돕는(suggest)모델이 아닌 결정을 자신이 내리는(autonomous)한 모델

(2) 시스템 가정

· 협상에 참여하는 주체는(판매자, 소비자)모두 합리적이고 일반적인 경제논리에 의해 의사결정을 한다.

예1) 각 주체의 예약가 사이에서만 매매 계약(struck bargain)이 일어난다. 즉 그림 2와 같이 $RP_s < RP_b$ 가 되어 협상가능 구간(the zone of agreement)이 생성되었을 경우에만 매매가 성립된다. (Raiffa[9])

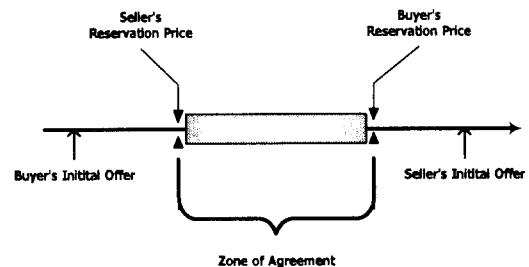


그림 2. 협상 가능 구간

예2) 그림 3의 경우, 즉 $RP_s > RP_b$ 인 경우에는 협상이 성립되지 않는다.

예3) 거래가 진행될수록 가격 과장 비율(inflated price rate)은 줄어든다.

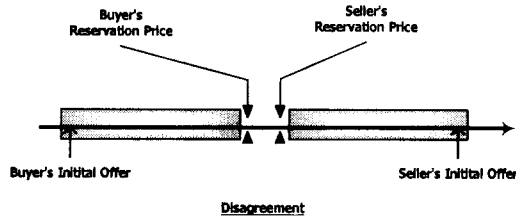


그림 3. 협상 가능 구간이 존재하지 않는 경우

- 판매자의 RP_s 값과 고시가격의 차이의 비율인 가격 과장 비율(inflated price rate)은 초기 17%로 가정한다. (Zeng[14])
- 각 주체별 예약가의 예측은 에이전트의 탐색에 의해 고시 가격 중 최저가에 속하는 가격을 초기 가정(null hypothesis : H_0)으로 지정한다.
- 판매자의 제안과 구매자의 제안의 차가 일정 % 내에 들어오게 되면 거래가 성립되었다고 간주할 수 있는 가격의 차인 거래허용치(allowable gap): $10\% \text{ of } |RP_b - RP_s| > |O_s^n - O_b^n|$ 이면 거래성립

where

- O_s^n : 판매자의 n 번째 제안
- O_b^n : 구매자의 n 번째 제안

2.2 협상전략 발견 모델

(1) 협상전략 발견 모델

협상 에이전트가 협상에 참여하여 상대방의 제안을 평가, 예측하고 새로운 제안을 생성하기 위해서는 에이전트가 지식을 가지고 있어야 한다.

이러한 협상 에이전트가 가져야할 지식을 협상전략(negotiation strategy)라고 하며, 이는 과거의 협상데이터로부터 지식발견 에이전트가 특정 속성 조합에 대한 대표적 협상패턴(representative negotiation pattern)을 추출·구조화하여 새로운 협상전략공간을 구성하게 된다.

이를 협상전략 발견 모델(model of finding negotiation strategy)이라고 하며, 협상패턴을 추출·구조화를 위해 경쟁학습(competitive learning)을 이용하여 의사결정트리(decision tree)를 구축한다. 구축된 트리를 본 논문에서는 NPT(Negotiation Pattern Tree: 이하 NPT)라고 하며 NPT는 속성에 의해서 분류되어진 협상패턴을 그 결과 값으로 갖게되며, 결과 값은 다시 협상전략공간으로 분류되어 저장되게 된다.

협상전략 발견 모델의 절차는 다음과 같다.

① 데이터 변환(Data Transformation)

과거의 협상데이터를 일정한 규칙으로 정규화하는 과정이다.

② 전처리 과정(Preprocess)

경쟁 학습망(competitive learning: 이하 CL)을 이용 협상 데이터 중 수치형 데이터에 대한 임계값(threshold)을 찾아 이산화 하는 과정이다.

③ 협상패턴트리(Negotiation Pattern Tree)

CL을 이용하여 협상 데이터 중 순차적으로 NPT구축에 필요한 결정노드를 선정하고 트리를 펼쳐나가는 확장(expansion) 단계와 구축한 의사결정트리를 가지고 시스템의 민감도를 조정하며 트리를 재구성하는 병합(merge)단계의 2단계 시스템을 거쳐 협상패턴트리를 구성한다.

④ 후처리 과정(Post-process)

발견 1, 2단계를 거쳐 구성된 의사결정트리를 이용하여, 협상전략공간을 구성하게 된다.

(2) 경쟁 학습

본 연구에서는 데이터의 군집화와 각 군집의 중심값을 산출하기 위해서 신경망 이론 중 입력층과 경쟁층의 두 개의 층(layer)으로 구성된 경쟁 학습(competitive learning: 이하 CL)을 이용하였다.

2.2.1 데이터 변환(Data Transformation)

본 논문의 협상전략 발견 모델에는 입력데이터로 과거의 협상데이터가 사용되게 된다. 하지만 하위 단계들에서 사용되는 CL의 입력데이터로 사용하기 위해서는 코딩(coding)·분리(dividing)·정규화(normalizing)등의 작업을 거쳐야 한다. 따라서 모델의 첫 단계에서 데이터에 대한 변환(transformation) 작업이 필요한 것이다. (Adriaans[2])

데이터의 형식에 따른 분류는 다음과 같다.

- 이산형(discrete 혹은 nominal) 데이터 : 속성 값 중, 정수 값으로 사상(mapping)될 수 있는 데이터이며 코딩(coding) 과정이 적용된다.

예: 판매자 {철수, 영희} → 판매자 {1, 2}

- 수치형(numeric 혹은 continuous) 데이터 : 속성 값 중, 연속형 실수 값을 가진 데이터로서 데이터이며 클래스를 나누는 작업인 분리(dividing)가 적용된다.

예: 가격 {1000, 1500, 1300, 27000, 23000} →

Price class A {1000, 1300, 1500}, Price class B {27000, 23000}

입력데이터 속성의 종류와 특성, 그리고 변환 형태는 다음과 같다.

- 협상대상물의 속성(attribute of negotiation item): 협상대상물에 관련된 속성으로 제품일 경우는 제품의 산지(the place of origin), 수량, 가격대 등이다.

- 협상의 속성(attribute of negotiation): 협상 대상 외적인 요소로서, 판매자ID, 협상시기, 등 협상대상 외적인 요소가 속한다고 할 수 있다.

- 협상진행의 패턴(negotiation pattern): 과거 협상과정의 진행형태를 나타내는 데이터로 가격을 판매자 내정 가격(reservation price of seller: 이하 RP_s)에 대한 비율로 변환시키는 과정을 정규화(normalizing)이라고 정의한다.

예: $RP_s = 10000$ 일 때, 예제 데이터를 표 2 와 같이 정규화한다.

S	0	1	2				
예제 데이터	O_s^u	O_b^u	O_s^l	O_b^l	O_s^z	O_b^z	거래 가격
	11700	9800	11500	10500	10800	10800	10800
정규화된 데이터	O_s^u	O_b^u	O_s^l	O_b^l	O_s^z	O_b^z	거래 가격
	17	-2	15	5	8	8	8

표 2. 협상진행의 패턴

S (Negotiation_ID) : 한 협상에서의 판매자와 구

매자의 제안의 쌍(pair)을 단계라고 하며, $S(Negotiation_ID)$ 는 협상의 결국 협상의 진행과정을 의미한다.

$N(Negotiation_ID)$: 협상이 시작되어서 합의가 되어 협상이 타결될 때까지의 협상횟수를 말한다. 이를 협상도수(negotiation frequency)라고 한다.

$O_s^{(S(Negotiation_ID))}$: 협상단계 $S(Negotiation_ID)$ 에서 판매자가 제의한 가격

$O_s^{(S(Negotiation_ID))}$: 협상단계 $S(Negotiation_ID)$ 에서 판매자가 제의한 가격을 RPs를 기준으로 정규화한 값으로 $O_s^{(S(Negotiation_ID))} = \frac{\hat{O}_s^{(S(Negotiation_ID))} - RPs}{RPs}$ 을 이용하여 구할 수 있다.

• 거래가격(Payoff)

특성: 협상이 타결된 가격에 대한 데이터로, 협상 진행의 패턴정규화와 같이 RPs에 대한 비율로 정규화한다(표 2에서 거래가격 참조).

2.2.2 전처리 과정(Preprocess)

전처리 과정에서는 연속형 데이터를 속성값으로 가진 속성을 몇 개의 집합으로 클러스터링을 할 것인가를 정하는 단계로 이 결과는 2.2.3단계에서 필요한 CL의 뉴런의 개수를 계산하는데 사용되며, 이는 또한 트리 확장의 기준(split criterion)이 된다.

본 논문에서는 트리 확장의 기준으로 혼잡도(entropy)를 정의하고, 이를 이용하여 클러스터링 임계값을 계산하는 방법을 제시한다. 혼잡도는 ID3의 gain ratio, ID5R의 E-score와 같은 역할을 한다.

(1) 클러스터링 임계값

이산형 데이터는 직관적으로 클러스터링이 가능하며 클러스터링의 개수를 대표하는 CL의 뉴런(neuron) 개수 지정이 명확하다. 하지만 연속형 데이터인 제품의 가격 $\in \{1000, 1200, 1340, 1500, 1290, 2200, 1950\}$ 과 같은 연속형 속성값의 경우에는 몇 개로 클러스터링을 하는 것이 가장 속성값의 혼잡도(entropy)를 낮추면서 클러스터링을 해야 하는지의 문제가 생기게 된다.

본 논문에서는 가장 혼잡도가 낮은 데이터의 클러스터링을 위하여 혼잡도를 정의하고 실험을 통하여 이러한 속성값에 대한 혼잡도를 가장 낮출 수 있는 클러스터의 수로서 $N-cluster$ 를 정의하고 이를 구하는 방법을 제안한다.

• $N-cluster$: 클러스터링 임계값은 n 개의 연속형 데이터를 가진 벡터에 대해 혼잡도를 가장 낮추면서 클러스터링을 할 수 있는 CL의 뉴런 수이다.

• 혼잡도(entropy) : n 개의 속성값을 가진 벡터를 k 개의 클러스터로 나누었을 때 CL에서 각 클러스터를 구성하는 가중치(weight vector)와 해당 클러스터의 각 원소사이의 거리의 합으로 정의한다.

$$Entro(V_n^*) = \sum_{i=1}^n \sum_{j=1}^k \frac{\sqrt{(w_i^j - v_i^j)^2}}{k}$$

where

$Entro(V_n^*)$: n 개의 속성값을 가진 벡터 V 를 k 개의 클러스터로 나누었을 때의 혼잡도

V_n^* : 벡터 V 의 n 개의 속성 값 $v_1, v_2, v_3, \dots, v_n$ 을 k 개의 군으로 분류한 데이터

k : n 개의 속성값을 나누는 클러스터의 개수

m^k : n 개의 속성값을 k 개의 군으로 나누었을 때 k 군에 속하는 속성값의 개수

v_i^k : 벡터 V 의 속성 값 중 k 군에 i 번째 속하는 속성 값

(2) 절차

속성값에 대한 혼잡도를 가장 낮출 수 있는 클러스터의 수를 결정하는 $N-cluster(V)$ 를 구하는 절차는 다음과 같다.

Step1. 협상데이터중 연속형 데이터를 구별한다.

Step2. 속성값을 CL의 입력으로 구성한다.

For $i = 1; i++ \{$

CL을 이용하여 i 개의 neuron을 사용하여 클러스터링을 한다.

클러스터링된 데이터를 가지고 $Entro(V_n^*)$ 를 계산하고 저장한다.

If $Entro(V_n^{*i}) > Entro(V_n^*) < Entro(V_n^{*i+1})$

GOTO Step3

End If

}

Step3. $N-Cluster(V) = k$

Step2에서와 같이 혼잡도가 가장 적어지는 변곡점(inflexion point)을 만드는 k 값이 $N-cluster$ 값이 된다. 위와 같은 작업을 직관적으로 클러스터링이 되지 않는 이산형 데이터를 갖는 속성에 대하여 수행한다.

2.2.3 협상패턴트리(Negotiation Pattern Tree)

의사결정트리 구축 단계는 협상데이터의 속성값과 협상패턴을 가지고 의사결정트리(decision tree: 이하 트리)를 구성하기 위하여 각 속성 값 사이의 상관관계(correlation)를 분석하여 노드의 선후관계를 정의하고 초기트리(incipient tree)를 구성하는 1단계와 전체 트리의 민감도를 이용하여 트리를 재구성하는 2단계의 구조를 가지고 있다.

본 논문에서는 결과로 생성되는 협상패턴의 분리된 트리를 협상패턴트리(Negotiation Pattern Tree; 이하 NPT)라고 한다.

1) NPT 구축 알고리즘

NPT 구축 알고리즘의 전체 과정은 그림 4와 같다.

(1) $N-cluster$

본 논문에서는 트리 확장의 기준(split criterion)으로 2.2.2 전처리 과정에서 논의한 $N-cluster$ 를 사용한다. $N-cluster$ 는 ID3의 gain ratio (Quinlan[8]), ID5R의 E-score (Utgoff[13]), AIDT의 L-value (김강호[1])와 같은 역할을 하는 것으로 한 노드의 $N-cluster > 1$ 인 경우에는 $N-cluster$ 값이 바로 결과 클래스의 개수(count)를 의미하며, 이는 바로 가지가 $N-cluster$ 값만큼 발생한다는 것을 의미한다.

(2) CR

협상데이터의 경우에는 각 속성이 이익에 미치는 상관관계(correlation)가 크다. 본 논문에서는 이러한 각 속성마다의 상관관계를 고려하기 위하여

속성과 이익과의 상관관계를 표현하는 CR을 정의하고, 이러한 상관관계를 순차적으로(sequentially) 계산하는 방법을 제시한다.

각 속성이 이익에 미치는 상관관계인 CR은 트리의 각 노드의 선후관계(부모 노드와 자식 노드)를 정하는 클래스의 속성을 정해주는 기준이 된다.

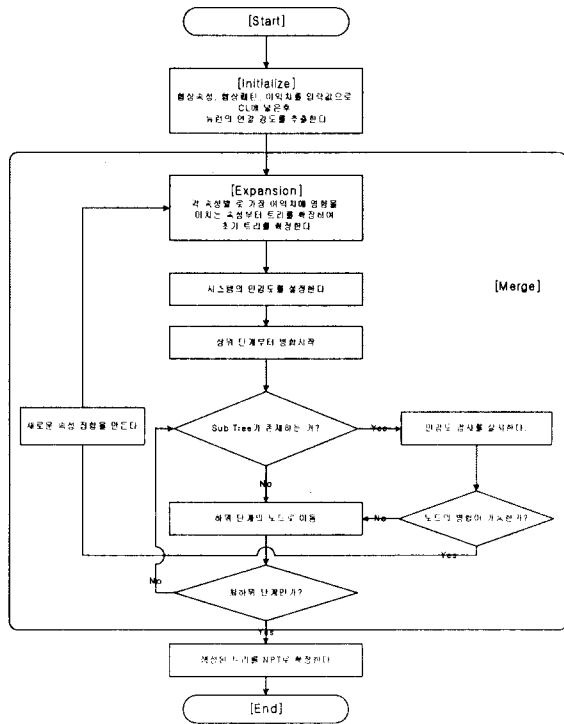


그림 4. NPT 구축 흐름도

• $CR(attribute)$: 각 속성별로 이익에 미치는 영향을 판단하는 변수로 CL의 결과로 생성되는 이익뉴런 가중치(payload neuron weight vector)와 해당 속성 클러스터링된 각 클러스터의 중심치 사이의 차이의 합이다.

$$CR(attribute) = \sum_{k=1}^m \sum_{j=1}^{m^k} \sqrt{\frac{(w_{payoff}^k - \omega_j^k)^2}{k}}$$

where

- k : 속성별 클러스터의 개수
- m^k : k 번째 클러스터에 포함된 뉴런의 개수
- w_{payoff}^k : k 번째 클러스터의 이익뉴런 가중치의 평균값
- ω_j^k : k 번째 클러스터의 j 번째 이익뉴런의 가중치

(2) NPT의 구축

• 용어 정의

NPT 구축 알고리즘의 설명을 위한 용어는 다음과 같다.

List(A) : NPT의 대상이 되는 모든 속성을 저장하는 리스트

Path : 이미 생성한 노드의 정보를 저장하고 있는 리스트

ω : 확장의 대상이 되는 속성

ω^* : 다음 확장의 대상이 되는 속성

$depth(i)$: NPT의 깊이, 예) 루트(root)는 $depth(i)$, 리프는 $depth(속성의 개수)$

$s(attribute)$: 속성에 대한 패턴을 포함한 자식

노드(sibling)의 개수

NPT의 구축절차는 다음과 같이 크게 3단계로 나누어 생각해 볼 수 있다.

Initialization

초기화(initialization)단계에서는 속성, 패턴, 협상이익을 입력값으로 CL을 거쳐 생성된 협상에 대한 클러스터들의 연결강도를 얻어낸다.

본 논문에서는 CL의 초기 뉴런의 개수를 N -neuron으로 정의한다.

• N -neuron

정의: 협상 데이터가 입력되는 CL의 뉴런의 개수이며, 뉴런의 개수는 2.2.2에서 구한 각각의 연속형 속성 N -cluster값의 곱과 이산형 속성의 클러스터의 개수의 곱이 된다.

$$N\text{-neuron} = \prod_{i=0}^m N\text{-cluster}(continuous\ attribute^i) \times \prod_{j=1}^m \Phi(discrete\ attribute^j)$$

where

n : 전체 속성의 수

m : 전체 속성의 수 중 연속형 속성의 개수

$N\text{-cluster}(continuous\ attribute^i)$: i 번째 연속형 속성에 대한 클러스터의 개수

$\Phi(discrete\ attribute^j)$: j 번째 이산형 속성에 대한 클러스터의 개수

Expansion and Merge

확장(Expansion)단계는 NPT를 위한 초기 트리를 구성하는 단계이며, 병합(Merge)단계는 초기트리를 트리의 민감도에 따라 재구성하는 단계이다.

• 트리의 민감도 검사

트리의 민감도란 트리가 분류해낸 상대방의 전략을 얼마나 자세히 분류 시킬지를 정하는 문제이다. 본 논문에서는 트리의 민감도의 척도로서 α 값을 정의하였다. 트리의 민감도를 계산해 주는 방법은 다음과 같다.

Case 1. 병합할 노드들에 패턴이 2개 이상 존재하는 경우

- 1) 민감도 α 를 정의한다. ($0 < \alpha < 1$)
- 2) 해당 트리깊이(depth)의 노드를 구성하는 속성 데이터의 최대값과 최소값의 차이를 계산하여 이 값을 β 라 한다.
- 3) If (속성 클러스터의 중심값의 차이 $\times \alpha$) $> \beta$ then 해당 클러스터는 병합 가능하다

Case 2. 병합할 노드들에 패턴이 1개씩만 존재하는 경우

- 1) 민감도 α 를 정의한다. ($0 < \alpha < 1$)
- 2) 해당 속성의 모든 데이터의 최대값과 최소값의 차이에 α 를 곱하여 이 값을 α' 라 한다.
- 3) 각 클러스터의 평균값의 차이를 계산하여 이 값을 β 라 한다.
- 4) If $\alpha' < \beta$ then 해당 클러스터는 병합 가능하다

• Subtree

동일한 깊이(depth)상의 노드의 속성은 동일하여야 하며, 같은 길이의 depth를 가지고 있는 트리를 subtree로 정의한다. 즉 구조가 가상의 루트(root)

를 기준으로 동일하여야 한다.
 • 확장단계에 대한 의사(pseudo)코드는 다음과 같다.

```

Expansion(List(A), Path)
{
    If  $\omega^*$  가 존재한다면
         $\omega = \omega^*$ 
        GOTO Step1
    Else
        List(A)에 있는 모든 속성에 대하여 CR계산
        List(A)중 최소의 CR값을 가진 속성을  $\omega$ 로
        정한다
        If  $\omega$ 에 속성이 2개 이상 존재할 경우
        임의의 속성을 선택한다.
        선택되지 않은 속성을  $\omega^*$ 으로 정한다.
        End if
        GOTO Step1
    End if
}
    
```

```

Step 1
{
     $s(\omega) = 0$ 
    For  $k = 1$  to  $N\text{-cluster}(\omega)$ 
        해당 클러스터에 패턴이 존재하는 지를 확인
        If 존재한다면
             $s(\omega) ++$ 
        End if
    Next
     $\omega$ 속성에 대한 자식노드를  $s(\omega)$ 개 생성한다
    List(A) = List(A) -  $\omega$ 
    For  $k = 1$  to  $s(\omega)$ 
        자신의 현재 위치를 Path에 기록한다
        Expansion(List(A), Path)
    Next
}
    
```

• 병합단계에 대한 의사코드는 다음과 같다.

```

Merge(i)
{
    For  $i = 0$ 
        depth(i)부터 Subtree를 찾는다
        If Subtree가 존재한다면
            민감도 검사를 한다.
            If 민감도 검사를 만족한다면
                depth(i)의 속성을 제거한 다음
                List(A)에 depth(1)부터 depth(i)까지
                의 속성을 뺀 나머지 속성을 집어넣는다
                Path 재설정을 한다
                Expansion(List(A), Path)
            Merge(i-1)
        End if
    End if
    Next
}
    
```

2.2.4 후처리 과정(Post-process)

(1) 분석

발견 1,2단계를 통해 생성한 의사결정트리에서 협상전략공간(negotiation strategy dimension)을 구성하는 단계로 각 의사결정의 리프까지의 데이터의 속성을 기준으로 협상전략공간을 구성하는 단계이다.

우선 의미 있는 속성을 부여하는 과정이 필요하다. 2.2.1 데이터 변환에서 예: 산지 {일본산, 대

만산} → 산지 {1, 2} 와 같이 이산화 시킨 데이터에 다시 {산지 1 → 일본산, 산지 2 → 대만산}과 같이 원래의 의미 있는 속성을 부여하게 된다.

예를 들어 발견 1,2 단계를 통해 얻은 트리는 그림 5와 같고, CL을 통해 얻은 협상속성과 협상진행의 패턴은 표 3과 같다.

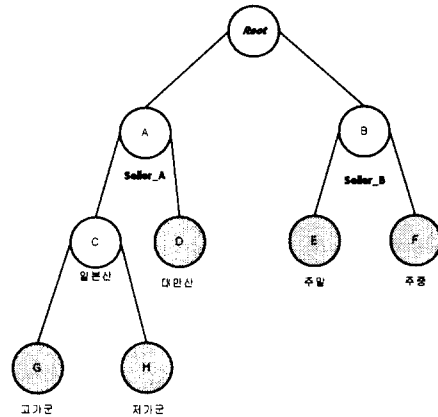


그림 5. 구축된 NFT

속성	판매자	산지	가격대	시기	ω_0^a	ω_0^b	ω_1^a	ω_1^b	ω_2^a	ω_2^b	ω_3^a	ω_3^b	payoff
$\#_1$	S_A	일본	고가	주말	17	15.62	16.89	16.14	16.16	16.16	.	.	16.16
$\#_2$	S_A	일본	저가	주중	17	-0.86	15.76	13.39	14.82	13.50	13.98	13.84	13.85
$\#_3$	S_A	대만	고가	주말	17	13.68	16.91	15.19	16.06	15.63	15.64	15.63	15.63
$\#_4$	S_B	대만	고가	주말	17	1.66	6.83	5.30	5.70	5.50	.	.	5.50
$\#_5$	S_A	일본	고가	주중	17	0.77	15.79	13.67	14.87	13.84	14.42	14.12	14.12
$\#_6$	S_B	대만	저가	주말	17	6.45	8.21	7.27	7.66	7.43	7.49	7.46	7.46
$\#_7$	S_B	일본	저가	주말	17	10.02	11.41	10.63	10.79	10.71	.	.	10.71
$\#_8$	S_A	대만	고가	주중	17	4.59	8.14	5.27	6.14	5.71	.	.	5.71

표 3. 협상속성과 협상진행의 패턴

그림 5의 트리에서 리프의 데이터 값들은 모두 상위 리프의 속성값에만 영향을 받는 데이터라는 것을 알 수 있다. 그리고 표 3에서 각 $\#_i$ 값들이 의미하는 것은 동일 행(row)내의 속성을 가졌던 협상진행의 패턴들이 클러스터링된 뉴런 연결강도의 평균 값을 의미하게 된다.

그림 5와 표 3을 연결시켜서 정리하면

- $\#_1, \#_5$ → 노드 G를 생성한 속성값에 대응되는 협상진행 패턴
- $\#_2, \#_7$ → 노드 H를 생성한 속성값에 대응되는 협상진행 패턴
- $\#_3, \#_8$ → 노드 D를 생성한 속성값에 대응되는 협상진행 패턴
- $\#_4$ → 노드 E를 생성한 속성값에 대응되는 협상진행 패턴
- $\#_6$ → 노드 F를 생성한 속성값에 대응되는 협상진행 패턴

즉 결론적으로 트리의 리프의 속성값에서 우리는 어떠한 속성이 협상에 중요한 영향을 미치는지의 유무와 그러한 속성들과 동일한 새로운 데이터가 들어왔을 때 협상진행의 패턴을 예측할 수 있다는 의미가 되며 이는 바로 협상에이전트가 전략, 즉 지식을 가지고 협상에 임할 수 있게 된다는 것을 의미한다.

(2) 협상전략공간 생성

실제 협상 속성값에 대응되는 협상진행패턴은 바로 지식으로 사용할 수 없다. 이 값을 다시 협상에

사용될 수 있는 개개의 전술(tactics)로 바꾸어 아 한다.

·패턴(negotiation pattern) : 단위 협상에서 보여진 판매자와 구매자와의 가격의 흐름을 나타낸 것으로 한 단위의 협상에서의 진행과정과 그 결과로서의 이익 (payoff)을 의미한다.

·전술(negotiation tactics) : 협상 과정 중 협상의 단위(negotiation frequency)마다의 판매자 또는 구매자의 결정(decision)을 의미하는 것으로 본 논문에서는 표 4와 같이 3가지 종류의 작전을 정의한다.

전술	Seller	Buyer
강 (Robust)	O_s^* 과 O_b^* 의 차이중 상위 1/3사이에서 결정	O_s^* 과 O_b^{*-1} 의 차이중 하위 1/3사이에서 결정
중 (Mediocre)	O_s^* 과 O_b^* 의 차이중 중간 1/3사이에서 결정	O_s^* 과 O_b^{*-1} 의 차이중 중간 1/3사이에서 결정
약 (Amicable)	O_s^* 과 O_b^* 의 차이중 상위 1/3사이에서 결정	O_s^* 과 O_b^{*-1} 의 차이중 상위 1/3사이에서 결정
초기치 및 조건	O_s^* 이 RP_s 이하일 경우에는 O_s^* 을 RP_s 값으로 대체	H_b 과 RP_b 의 차이중 하위로 10%를 늘린 후, 이를 3등분하여 결정

표 4. 전술의 종류

·전략(negotiation strategy) : 협상을 다수의 결정점(decision point)들이 모여있는 구조로 본다면 전략이란 단위 협상을 하기 위한 작전의 합이라고 할 수 있다.

·전략 셋(negotiation strategy set) : 어떠한 속성집합에 대하여 클러스터링된 모든 전략들의 집합을 의미한다.

결국 협상전략공간 생성이란 각 속성에 클러스터링된 n 값의 패턴들을 다시 작전으로 바꾼 다음 이를 전략화하고, 각 속성에 대응되는 전략들 중 동일한 속성에 포함된 전략들을 이익(payoff)이 높은 순서로 정렬하여 저장하는 것을 의미한다.

2.4 자동 협상 모델

2.4.1 협상과정

이 장에서 다룬 자동 협상 모델은 에이전트를 이용한 자동 협상 모델 중 협상 에이전트 계층을 모델화 한 것으로 2.3에서 구현한 협상전략 발견 모델에서 얻어진 협상전략을 이용하여 실제로 협상을 하는 과정을 모델링 한 것이다.

자동화된 협상과정은 다음과 같은 3가지의 과정을 따른다.

(1) Match Making

새로운 협상의 인스턴스(instance)가 생성되게 되면 에이전트는 새로운 협상의 속성을 파악하게 되고 이러한 속성을 가진 제품을 판매하는 판매자를 검색 에이전트에게 요청하게 된다. 요청을 받게 된 검색 에이전트는 검색 모델에 따라 요청한 제품과 동일한 속성을 가진 제품을 판매하는 판매자를 검색하여서 결과를 협상 에이전트에게 넘김과 동시에 협상 에이전트는 해당 판매자와 연결(connection)을 하게 된다.

해당 판매자와 연결이 된 후에는 협상 에이전트는 협상 전략 발견 모델에서 구현되어 있는 협상 전략

공간에서 새로운 협상 인스턴스의 속성과 동일한 속성을 가지고 있는 전략 셋 을 찾아내는 과정을 수행한다.

(2) 협상과정(negotiation process)

찾아낸 전략 셋 중 가장 이익(payoff)이 높은 전략을 가지고 협상에 임한 협상 에이전트는 판매자와 협상과정에 들어가게 된다.

(3) Finalize negotiation

이와 같은 방법을 반복하여 협상 가능 구간 내에서 상대방의 제안과 나의 제안이 거래 허용치 내에 들어왔을 때에는 협상을 마감(struck bargain)하고, 해당 협상의 패턴 및 이익을 데이터 베이스에 저장하게 된다. 이와 동시에 마이닝 에이전트에게 새로운 전략공간 구성을 요청(trigger)하게 된다. 이와 같은 과정으로 새로운 협상 인스턴스에 대한 자동 협상은 이루어지게 된다.

·자동 협상 모델에 관한 개략코드는 다음과 같다.

Label: Autonomous Negotiation Process

Preprocessing

sub_1 판매자와 연결한다.

sub_2 Match making

sub_3 전략 셋에 들어 있는 전략 중에서 가장 이익이 높은 전략을 뽑아낸다.

Main

Begin

초기제안(initial offer)을 시작한다.

For (10 % of $|RP_b - RP_s| > |O_s^* - O_b^{*-1}|$) {

If 판매자의 제안 \in 전략

전략에서 다음 전술을 찾아 제안한다.

Else If

후보 전략의 다음 전술을 찾아 제안한다.

End If

End

ENDSTEP

sub_1 거래를 승인한다.

sub_2 데이터를 DB에 저장한다.

3. 수행도 평가

3.1 수행 척도

본 논문에서는 자동협상모델의 수행척도로서 협상 타결확률, 협상이익(payoff)을 제시한다.

·협상타결비율(rate of struck bargain) : 자동협상모델과 상대방(opponent)과의 협상과정에서 전체 협상시도에 대한 협상이 타결된 비율이다. (struck bargain)

·협상이익(payoff) : 자동협상모델의 효율성을 계산해주기 위해서는 자동협상모델이 타결시킨 협상에서 얻은 이익을 측정하여야 한다.

3.2 실험 환경

실험은 협상 전문가의 이익구조의 형태에 따라서 행동하는 에이전트와 협상 전문가의 과거 협상데이터로부터 협상전략을 추출한 본 시스템과의 시뮬레이션 실험을 통하여 협상을 진행하였다.

(1) 협상 속성 (표 5)

실험에 사용되는 협상에 대한 제품은 컴퓨터 부품 중 그래픽 카드(graphic card)를 대상으로 하였다.

속성	속성	형식	값
판매자	협상대상자의 속성	이산형	Seller 1, Seller 2
산지	협상대상물의 속성	이산형	일본산, 대만산
가격	협상대상물의 속성	연속형	50000~290000
수량	협상대상물의 속성	연속형	1~20
판매일시	외부 변수	이산형	평일, 주말

표 5. 협상 속성

(2) 협상 전문가의 이익구조 (표 6)

Seller 분류	성향	협상 단계에서 구사하는 전략의 특징
Seller 1	제품에 대한 이익을 많이 고려함	협상의 각 단계의 진행시 강→중간으로 갈 수 있어도 강→약으로 가지 않음, 즉 2단계를 뛰어넘는 성향을 보이지는 않음
Seller 2	제품에 대한 이익을 많이 남기지 않더라도, 추후의 거래를 위해 다량 판매를 목적으로 함	강→중, 중→약 등 1단계를 뛰어넘는 경향과 강→약으로의 2단계로 뛰어넘는 성향을 동시에 보임

표 6. 협상 전문가의 특징

(3) 실험 환경

협상 전문가의 1200개의 과거 데이터로부터 NPT를 구성하기 위한 CL의 조건과 NPT의 민감도의 값은 다음과 같다.

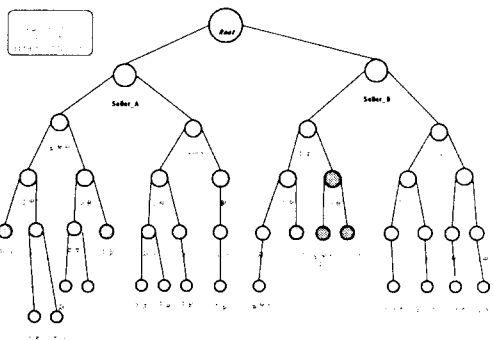
- 실험 데이터 : 1200 × 21(5 속성값, 16 협상진행값)의 벡터형태로 구성된 과거 협상 데이터
- CL의 학습횟수(epoch: 이하 me) : 5000회, 10000회, 100000회
- CL의 학습비율(learning late: 이하 lr) : 0.5, 0.7, 0.9
- NPT의 민감도 α : 0.1, 0.15, 0.2, 0.3, 0.75

3.3 결과 분석

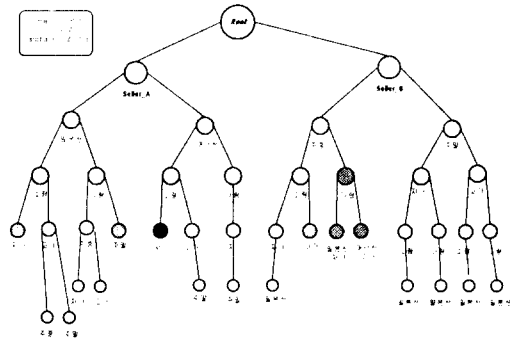
(1) NPT구성

me = 10000, lr = 0.7에서 만들어진 NPT는 다음과 같다.

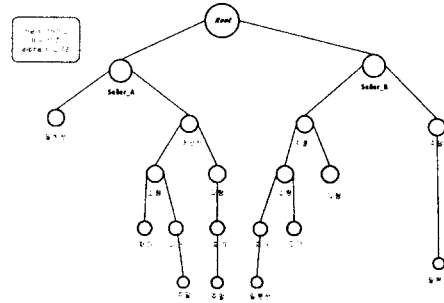
- $\alpha = 0.1, 0.15$



- $\alpha = 0.2, 0.3$



- $\alpha = 0.75$



생성된 NPT는 α 값이 증가함에 따라 패턴의 병합이 일어나는 것을 볼 수 있다. 이는 α 값이 증가함에 따라 NPT의 민감도가 떨어져서 NPT가 과거 협상 데이터의 패턴을 각 속성에 따라 민감하게 분석하지 못한다는 것을 의미한다.

(2) 협상타결비율

me \ α	0.1	0.15	0.2	0.3	0.72
5000	0.33	0.31	0.31	0.31	
10000	0.34	0.32	0.33	0.33	0.41
100000	0.40	0.40	0.39	0.39	

표 7. 협상타결비율

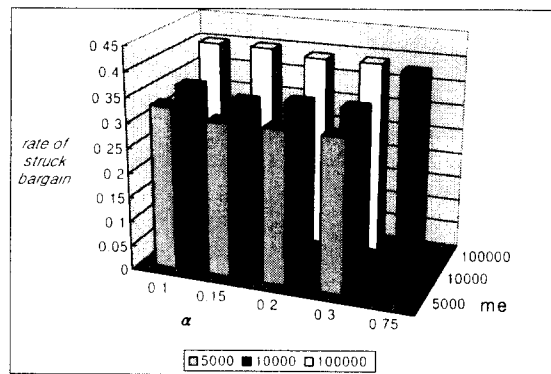


그림 6. 협상타결비율

협상전문가 에이전트와의 협상 시뮬레이션은 임의의 243가지의 협상에 대하여 각각 1000번씩의 협상 시뮬레이션을 통해 협상타결확률을 구하였다.

표 7에서 알 수 있듯이 전체적인 실험의 결과는 협상타결확률이 30%이상을 기록하는 수치로 이는 Rangaswamy[10]의 연구결과인 인간의 협상타결확률 11.7%보다 높게 나오는 것을 알 수 있다.

특히 me가 증가함에 따라 성공확률이 높아지는 것

을 알 수 있는데, 이 결과는 CL의 학습횟수가 증가함에 따라서 더욱더 패턴을 정확히 예측해 내고 이 결과는 협상에서 더욱더 효율적인 전략이 반영된 것으로 판명되었다.

그리고 예외적으로 α 를 0.75같이 높게 잡은 경우에는 시스템의 민감도가 떨어져 시스템의 민감도가 높았을 경우에는 협상을 하지 못했던 조건들에 대해서도 협상이 가능하여서 타결비율이 높아진 것으로 판명되었다.

(3) 협상이익

α \ me	0.1	0.15	0.2	0.3
5000	9.95	9.86	9.86	9.63
100000	10.69	10.69	10.49	10.49

표 8. 협상이익

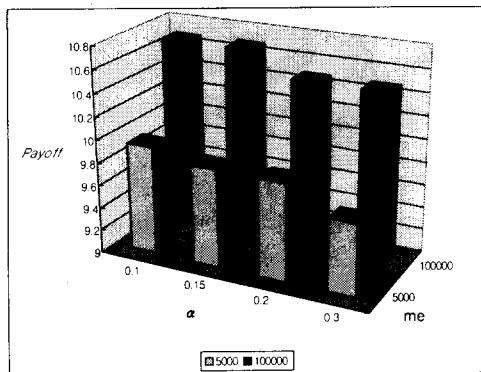


그림 7. 협상이익

협상이익의 결과는 시스템의 α 가 증가함에 따라 협상이익이 조금 증가하는 경향을 보여준다. 이 결과는 시스템의 민감도가 감소함에 따라 각 속성에 따른 패턴의 분류가 적어져, 시스템의 협상 에이전트가 민감도가 높은 경우의 협상전략보다 다양한 협상전략을 가지고 협상에 참여하기 때문으로 판명되었다.

4. 결론

본 논문에서는 과거의 협상데이터를 바탕으로 모델을 구축하고, 구축된 모델로부터 데이터 마이닝을 통하여 정보를 추출하는 모델을 제시하고, 구축된 정보(지식)를 통하여 인간의 역할을 대신하여 에이전트가 행동해야 하는 협상 프로세스를 정의하고 이를 구현할 수 있는 모델을 제시했다는데 본 논문의 의의가 있다고 하겠다.

본 모델은 이러한 새로이 대두되고 있는 비즈니스 영역에서의 정성적으로만 연구되어 오던 "협상"이라는 문제에 대한 정량적인 접근을 시도했다는 점에서 그 의의를 찾을 수 있으며, 단순히 "협상지식"에 대한 발견기법 이외에 이를 이용할 수 있는 "협상"에 대한 프로세스를 정의화 하여 새로운 비즈니스 모델(business practice)로 활용 가능하다는 점을 의의로 들 수 있다.

협상에 임할 때 가장 선행적으로 필요한 요소는 바로 판매자 또는 구매자의 이익구조에 관한 이해이다. 이러한 상대방의 이익구조를 바르게 파악할 때 일반적으로 협상에서 가장 큰 이익을 낼 수 있다. 하지만 1.1.1에서 언급한 바와 같이 인간의 협

상과정에서는 정보의 부재, 추론의 오류 등으로 판매자 또는 구매자의 이익구조의 이해가 어렵다.

본 논문에서 협상 에이전트의 지식으로서 구성된 최종의 협상전략공간이란, 즉 상대방의 과거 협상 데이터로부터 상대방이 이익 실현을 위하여 어떠한 협상 속성을 가장 중요하게 생각하는 가 하는 이익 구조에 대한 구조적인 접근결과라고 할 수 있다.

참고 문헌

- [1] 김강호, "중분 의사결정트리를 이용한 적응형 기계 진단 시스템", 고려대학교 석사학위논문, 1999
- [2] Adriaans, P., and Zantinge, D., *DATA MINING*, Addison Wesley, 1996
- [3] Chavez and Maes, P., "Kasbah: An Agent Marketplace for Buying and Selling Goods", Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96), London, UK, April 1996
- [4] Fisher, R., Ury, W. and Patton, B., "Getting to Yes : Negotiating Agreement Without Giving in", Houghton Mifflin Co, 2nd edition, April, 1992
- [5] Jim, R. O., "On Automated Negotiation and Electronic Commerce", Ph. D. Thesis
- [6] Lewicki, R., Saunders, D. and Minton, J., *Essentials of Negotiation*, Irwin, 1997.
- [7] Maes, P., Guttman R. and Moukas, A., "Agents that Buy and Sell: Transforming Commerce as we Know It", Communications of the ACM, March 1999 Issue
- [8] Quinlan, J. R., "Induction of decision trees", Machine learning, 1:81-106, 1986
- [9] Raiffa, H., *The Art and Science of Negotiation*, Belknap Pr, March, 1985
- [10] Rangaswamy, Arvind and Richard Shell, "Using Computers to Realize Joint Gains in Negotiations: Toward and Electronic Bargaining Table", Management Science, 1996
- [11] Richard P. L., John E. M. and David S. T., *Neural Information Processing systems 3*, Morgan Kaufmann Publishers.
- [12] Terpsidis, I., Pergioudakis, A. M., Doukidis, B. G. and Maes, P., "The Potential of Electronic Commerce in Re-Engineering Consumer-Retail Relationships through Intelligent Agents", J.-Y. Roger, B. Stanford-Smith, and P. Kidd. (eds.), *Advances in Information Technologies: The Business Challenge*, IOS Press, 1997
- [13] Utgoff, P. E., "Incremental Induction of Decision Trees", Machine Learning, Vol. 4, 161-186, 1989
- [14] Zeng, D. and Sycara, K., "Benefits of Learning in Negotiation", Proceedings of AAAI-97, 1997