

# 분산 환경에서의 협력적 여과를 위한 멀티 에이전트 프레임워크

지애띠

인하대학교 컴퓨터·정보공학과  
(aerry13@eslab.inha.ac.kr)

연철

인하대학교 컴퓨터·정보공학과  
(entireboy@eslab.inha.ac.kr)

이승훈

인하대학교 컴퓨터·정보공학과  
(shlee@eslab.inha.ac.kr)

김홍남

인하대학교 컴퓨터·정보공학과  
(nami@eslab.inha.ac.kr)

조근식

인하대학교 컴퓨터·정보공학과  
(gsjo@inha.ac.kr)

추천 시스템은 정보의 홍수 속에서 사용자가 하여금 자신에게 더욱 가치 있고 흥미로운 정보를 선별할 수 있도록 돕는 자동화된 정보 여과 시스템이다. 최근 분산 컴퓨팅 환경에 대한 연구가 활발히 진행되면서, 지금까지의 중앙 서버에서 모든 정보를 관리하는 중앙 집중 방식의 추천 시스템에서 P2P 환경의 접근 방식으로 선회하고 있다. 협력적 여과는 상업적인 추천 시스템에서 가장 많이 사용하는 정보 여과 기법이지만, 그 성공에도 불구하고 확장성(scalability)과 데이터의 희박성(sparsity), 악의적인 사용자의 공격(shilling attack)에 대한 방어 등에 관련된 여러 제약을 갖는다. 중앙 집중 방식에서 분산된 방식으로의 변화는 추천의 신뢰성과 개인 정보의 남용 가능성에 관련된 문제점을 일부 해결할 수 있으나, 조작된 사용자 프로파일을 사용하여 추천을 조작하려는 의도를 갖는 악의적인 사용자의 공격에는 중앙 집중 방식과 마찬가지로 취약할 수 있다.

본 논문에서는 개인 정보의 오남용과 악의적인 사용자의 공격에 관련된 문제점을 해결하고, 분산된 환경에서 효과적인 협력적 여과를 수행하여 추천의 성능과 정확성을 높이기 위한 멀티 에이전트 기반의 추천 프레임워크를 제안한다. 추천의 신뢰성을 높이기 위해 사용자 간의 신뢰 정보를 사용하며, 각 사용자의 개인 에이전트와 이동 에이전트 간의 정보교환을 통해 효과적으로 신뢰 정보를 전파하고 분산된 유사도 계산의 효율성을 높였다.

논문접수일 : 2007년 04월

게재확정일 : 2007년 08월

교신저자 : 김홍남

## 1. 서론

인터넷의 급속한 성장과 더불어 이를 통해 더욱 다양하고 많은 정보를 손쉽게 얻을 수 있게 되었다. 그러나 이와 함께 정보의 홍수라고 일컬어질 만큼 수많은 정보와 데이터들 가운데 사용자에게 가치 있고 흥미로운 것을 어떻게 찾아낼 것인가의 문제 또한 오늘날 정보 사회의 중요한 이슈

가 되었다.

추천 시스템(Recommender System)은 이러한 정보 사회의 요구로부터 발달한 자동화된 정보 여과 시스템으로, 사용자로부터 발단한 자동화된 정보 여과 시스템으로, 사용자로부터 어떤 정보가 자신에게 가치 있고 흥미로운 만한가를 결정하는 데 도움을 준다. 추천 시스템에서 가장 널리 사용되는 정보 여과 기법의 하나인 협력적 여과(Collaborative Filtering)는 크게 두 가지 접근 방법으로 발전되

어 왔다(Deshpande and Karypis, 2004). 유사한 성향을 가진 사용자는 같은 아이템을 선호할 것이라는 사실에 기초한 사용자 기반 접근 방법과(Resnick et al., 1994; Resnick and Varian, 1997; Konstan et al., 1997; Breese et al., 1998; Shardanand and Maes, 1995), 특정 아이템 혹은 아이템 집합에 대해 유사한 다른 아이템의 선호가 영향을 미치는 관계에 초점을 둔 아이템 기반 접근 방법이다(Sarwar et al., 2001; Breese et al., 1998; Shardanand and Maes, 1995; Deshpande and Karypis, 2004).

전통적인 사용자 기반 협력적 여과는 상업적인 추천 시스템에서 가장 널리 활용되고 있는 정보여과 기법임에도 불구하고, 실제 응용에서 많은 문제점을 초래할 수 있다. 사용자의 수가 증가할수록 계산의 복잡도가 크게 증가하며, 사용자-아이템 행렬의 분포가 희박한 경우 유사도의 계산이 어렵다(O'Donovan and Smyth, 2005; Deshpande and Karypis, 2004). 또한 대부분 중앙에서 사용자의 모든 데이터를 관리하고 이를 추천에 이용하는 중앙 집중적 방식으로, 추천의 신뢰성과 개인 정보의 남용 가능성에 대한 중요한 문제점이 제기될 수 있다(Kim, 2003).

이러한 문제점은 각 개인이 자신의 정보 공개 및 관리를 스스로 통제할 수 있는 개인화된 분산 추천시스템을 구현함으로써 일부 개선할 수 있다(Miller et al., 2004). 분산 추천 시스템은 최근 활발히 진행되고 있는 분산 컴퓨팅 환경에 관련된 연구에 발맞춰 여러 응용 분야로 그 범위를 확대하고 있다. 이러한 분산 환경에서 각 개인의 에이전트를 사용하여 에이전트 간의 의사소통을 통해 개인화된 추천을 제공하고 시스템의 효율성을 높이고자 하는 연구 또한 다양하게 진행되고 있다(Lange and Oshima, 1999; Foner, 1997; Good et al., 1999).

그러나 이러한 분산 추천 시스템 또한 추천에

영향을 끼칠 목적으로 조작된 사용자 프로파일을 생성하는 악의적인 사용자의 공격에 쉽게 노출될 수 있다(Miller et al., 2004). 추천에 참여하는 각 개인들 간의 신뢰망(Web of Trust)의 구축은 이와 같은 공격으로부터 시스템을 보호하기 위한 효과적인 방법의 하나이다(Massa and Avesani, 2004).

본 논문에서는 P2P 환경에서 각 개인의 에이전트가 개인의 정보를 관리하고 통제할 수 있는 멀티에이전트 기반의 프레임워크를 제안한다. 개인 에이전트와 이동 에이전트 간의 의사소통을 통해 각 개인 에이전트간의 협업이 이루어진다. 이를 바탕으로 보다 신뢰할 수 있고 개인화된 추천을 제공하고, 분산 컴퓨팅의 효율을 극대화 한다. 또한 각 개인이 다른 사용자에 대해 표현한 신뢰 정보를 바탕으로 한 협력적 여과 방법으로 보다 신뢰성 높은 추천을 제공하고자 한다. 커뮤니티 내에서 어떤 사용자를 신뢰할 수 있는가 그렇지 않은가를 결정하는 알고리즘인 Advogato 알고리즘을 사용하여, 신뢰 정보를 효과적으로 전파하면서 악의적인 사용자의 공격에 대한 저항성을 얻는다.

본 논문의 구성은 다음과 같다. 제 2장에서는 현재까지의 관련된 연구 분야를 개략적으로 살펴보고, 제 3장에서 본 논문에서 제안하고자 하는 멀티 에이전트를 이용한 분산 추천 시스템에 대해 자세히 기술한다. 제 4장에서는 다른 분산 추천 시스템과의 비교 실험과 분석을 통해 제안하는 시스템의 효율성을 보이고, 마지막으로 제 5장에서 결론을 맺고 향후 연구에 대하여 언급한다.

## 2. 관련연구

### 2.1 협력적 여과

현재 상업적인 추천 시스템에서 가장 널리 사용

되고 있는 기반 기술인 협력적 여과(Collaborative Filtering)는 “Word of Mouth”, 즉 구전되는 정보가 구매 의사결정에 지대한 영향을 미친다는 사실에서 출발한다(Shardanand and Maes, 1995). 예를 들어, 극장에서 사전 정보가 많지 않은 최신 개봉 영화를 관람하려고 하는 관객이 있다고 하자. 영화를 선택하는 데 있어 해당 영화를 먼저 관람한 주변 사람의 의견이나 관련 사이트나 커뮤니티 등을 통해 접할 수 있는 영화평 등은 다른 요인에 비해 비교적 큰 영향을 미친다. 만약 자신과 취향이 비슷해 유사한 장르를 선호하는 사람이 정보를 제공한다면, 그것이 미치는 영향은 더욱 클 수 있다.

### 2.1.1 메모리 기반의 협력적 여과

Resnick et al.에 의해 제안된 전통적인 사용자 기반의 협력적 여과는 위와 같은 관점에서 어떤 아이템에 대한 목적사용자(Target user)의 선호도 점수를 예측하는 추천 방법이다(Resnick et al., 1994; Resnick and Varian, 1997; Konstan et al., 1997; Breese et al., 1998; Shardanand and Maes, 1995).

<표 1> 사용자 아이템 행렬의 예

item user	AI	Star Wars	Matrix	X-men	Titanic
Alice	?	3	5	4	2
Bob	1	2	?	1	4
Cathy	5	4	5	3	?
David	2	3	2	1	5

사용자는 아이템에 대한 자신의 선호도를 점수로 표현하고, 이를 <표 1>과 같은 사용자 아이템 행렬로 표현할 수 있다.

우선, 이웃(neighborhood)으로 지칭되는 유사한

사용자의 집단을 구성한다. 이를 위해 각 사용자 간의 유사성 정도를 측정하는데, 코사인 유사도 식 (1)와 피어슨 상관관계(Pearson Correlation) (Devore, 1995)기반의 유사도 식 (2) 등의 측정 방법이 가장 대표적으로 사용되며, 다음과 같다.

$$sim(u, w) = \cos(\vec{u}, \vec{w}) = \frac{\sum_{i \in I} R_{u,i} \cdot R_{w,i}}{\sqrt{\sum_{i \in I} R_{u,i}^2} \sqrt{\sum_{i \in I} R_{w,i}^2}} \quad (1)$$

$$sim(u, w) = \frac{\sum_{i \in I} (R_{u,i} - \bar{R}_u)(R_{w,i} - \bar{R}_w)}{\sqrt{\sum_{i \in I} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{i \in I} (R_{w,i} - \bar{R}_w)^2}} \quad (2)$$

두 명의 사용자  $u$ 와  $w$ 의 선호도의 유사성 정도를 측정하려 할 때,  $R_{u,i}$ 와  $R_{w,i}$ 는 각각 사용자  $u, w$ 의 아이템  $i$ 에 대한 선호도를 수치로 나타낸다. 식 (1)의 경우, 전체 아이템 집합에서 두 명의 사용자가 동시에 선호도 점수를 표현한 모든 아이템  $i$ 에 대한 코사인 유사도를 합산하여 두 사용자 간의 유사도를 측정한다. 결과 값은 0과 1사이의 실수로, 1에 가까울수록 두 사용자의 선호도가 유사하다고 판단할 수 있다.

피어슨 상관관계는 두 변수가 서로 얼마나 관계가 있는가를 판단하는 측정방법으로, 이를 기반으로 한 식 (2)와 같은 유사도 측정방법을 사용할 수 있다. 코사인 유사도의 경우 사용자마다 점수를 매기는 기준이 다를 수 있다는 점, 즉 어떤 아이템에 대해 선호도를 표현할 때 같은 점수를 매기더라도 사용자마다 선호하는 정도가 다를 수 있다는 점을 간과하게 된다. 이를 극복하기 위해 아이템  $i$ 에 대한 각 사용자의 선호도 점수  $R_{u,i}$ ,  $R_{w,i}$ 에서 각 사용자가 다른 아이템에 매긴 점수의 평균점수인  $\bar{R}_u$ 와  $\bar{R}_w$ 를 각각 차감한다(Devore, 1995).

위와 같은 방법으로 목적사용자와 유사한 사용자들이 이웃 집단으로 구성되면, 이들의 유사도

값과 점수는 아직 점수를 매기지 않은 특정 아이  
템에 대한 목적사용자의 선호도가 어느 정도인지  
를 예측하는 데 사용된다. 이웃 집단을 구성하는  
이웃의 수가 너무 적으면 올바른 예측이 어려우  
며, 이웃의 수가 늘어날수록 예측의 정확도가 높  
아지지만 그만큼 계산량은 늘어나므로 가장 효율  
적인 이웃 집단의 크기를 결정해야 한다(Sarwar  
et al., 2001, Herlocker et al., 1999).

예측에 사용되는 식은 다음과 같다(Resnick et  
al., 1994).

$$P_{u,i} = \overline{R_u} + \frac{\sum_{w \in N} (R_{w,i} - \overline{R_w}) \cdot sim(u, w)}{\sum_{w \in N} sim(u, w)} \quad (3)$$

$P_{u,i}$ 는 식을 통해 예측할 아이템  $i$ 에 대한 목적  
사용자  $u$ 의 예측 선호도 점수를 의미한다. 이웃  
집단  $N$ 에 포함된 모든 사용자  $W$ 에 대해, 아이템  
 $i$ 에 대한 선호도 점수  $R_{w,i}$ 에서 해당 사용자가 점  
수를 매긴 모든 아이템에 대한 선호도 점수의 평  
균  $\overline{R_w}$ 를 뺀 값을 합산하며, 이 때 목적사용자와  
각 사용자의 유사도  $sim(u, w)$ 를 가중치로 한  
다. 이에 다른 아이템에 대한 목적사용자  $u$ 의 선  
호도 점수의 평균  $\overline{R_u}$ 를 합산하면  $P_{u,i}$ 의 값을 얻  
을 수 있다. 얻어진 예측 점수를 바탕으로 목적사  
용자  $u$ 에게 예측 선호도 점수가 높은 아이템을 추  
천한다.

위와 같이 가장 유사한 이웃 집단을 구성하여  
이를 바탕으로 추천하는 사용자 기반의 협력적 여  
과를 메모리 기반의 협력적 여과라고도 하는데,  
비교적 높은 예측 성능에도 불구하고 확장성과 실  
시간 성능, 사용자 아이템 행렬의 데이터 희박성  
에 관련된 심각한 제약은 갖는다(O'Donovan and  
Smyth, 2005; Deshpande and Karypis, 2004).

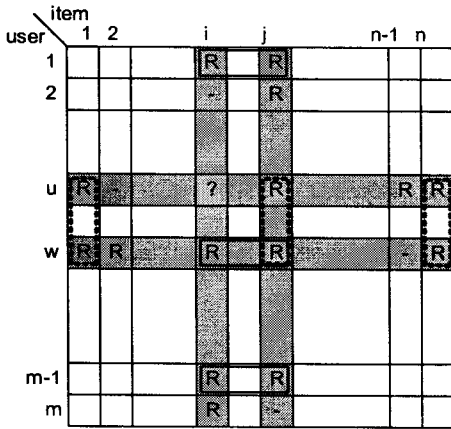
사용자와 아이템의 수가 증가할수록 이웃 집단

을 찾기 위한 유사도 계산의 복잡도는 크게 증가  
한다. 사용자의 수가 매우 많은 일반적인 상업적  
웹 사이트의 경우, 추천을 위해 각 사용자에 대한  
이웃 집단을 구성하는 것은 엄청난 계산을 요구하  
고 지속적인 사용자의 변화가 발생하므로 미리 계  
산된 사용자 간의 유사도나 이웃 집단의 이득을  
기대할 수 없다(Deshpande and Karypis, 2004).

또한 사용자 간의 유사도는 서로 다른 사용자  
가 같은 아이템에 매긴 점수를 바탕으로 계산되는데,  
아이템의 수가 많으면 많을수록 동시에 점수가  
매겨진 아이템(co-rated items)의 비율을 적어  
진다. 즉, 다른 사용자와 동시에 점수를 매긴 아이  
템을 전혀 가지고 있지 않은 사용자가 존재하여  
상관관계를 계산할 수 없거나, 그러한 아이템을  
가지고 있더라도 그 수가 매우 적어 추천의 정확  
성을 떨어뜨릴 수 있다(Sarwar et al., 2001; O'Donovan  
and Smyth, 2005; Deshpande and Karypis, 2004).

## 2.1.2 모델 기반의 협력적 여과

전통적인 협력적 여과의 문제점을 해결하기 위  
해 여러 모델 기반의 접근 방법이 제안되었다. 단  
순한 내용 분석을 통해 데이터 집합 내의 모든 아  
이템에 대한 자동적으로 임의의 선호도 점수를 생  
성하여 삽입하거나(Good et al., 1999; Sawar et  
al., 1998), 유사 이웃 집단의 클러스터를 미리 생  
성하여 추천에 사용하는 방법 등이 제안되었으며  
(Mobasher, 2002; Unger and Foster, 1998), 전체  
사용자-아이템의 행렬을 축약하기 위한 SVD 행  
렬을 사용하기도 한다. 이와 같은 접근 방법은 미  
리 사용자와 아이템 간의 관계를 나타내는 모델을  
구축하여 추천에 사용한다는 점에서 공통점을 갖  
는다. 대표적인 모델 기반의 협력적 여과 방법 중  
하나로 아이템과 아이템 간의 유사도를 모델 구축



[그림 1] 사용자 기반 협력적 여과와  
아이템 기반 협력적 여과

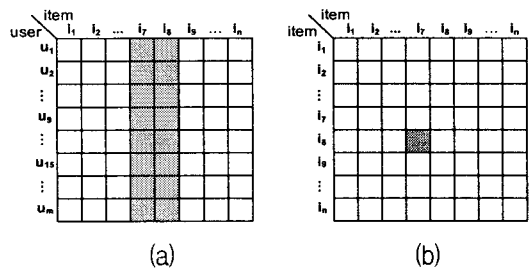
에 사용하는 아이템 기반의 협력적 여과에 대해 알아보자(Sarwar et al., 2001).

[그림 1]은 사용자 간의 유사도를 기반으로 하는 전통적인 협력적 여과와 아이템 간의 유사도를 사용하는 아이템 기반의 협력적 여과의 유사도 계산 방법을 비교하여 보여준다. 사용자  $u$ 의 아이템  $i$ 에 대한 점수를 예측하는 과정에서 점선으로 표시된 점수 쌍은 사용자  $u$ 와  $w$ 의 유사도를, 실선으로 표시된 점수 쌍은 아이템  $i$ 와  $j$ 의 유사도를 구하기 위해 사용되는 예를 보여준다.

아이템 기반의 협력적 여과의 주요 동기는 사용자가 과거에 이미 선호한다고 표현한 아이템 집합과 유사한 아이템을 선호할 가능성이 높다는 사실에 있다. 사용자 간의 유사도가 아닌 아이템 간의 유사도를 계산하여 얻어진 유사 이웃 아이템 집합을 바탕으로 미리 특정 아이템에 대한 유사도 모델을 구축한다는 점에서 전통적인 협력적 여과와 구별된다(Sarwar et al., 2001; Breese et al., 1998; Shardanand and Maes, 1995; Deshpande and Karypis, 2004).

$$sim(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\sum_{u \in U} R_{u,i} \cdot R_{u,j}}{\sqrt{\sum_{u \in U} R_{u,i}^2} \sqrt{\sum_{u \in U} R_{u,j}^2}} \quad (4)$$

아이템  $i$ 와  $j$ 의 유사도는 식 (4)와 같이 계산되며, 사용자 전체 집합  $U$ 에 포함된 모든 사용자  $u$ 가 두 아이템  $i$ 와  $j$ 에 매긴 선호도 점수에 대한 코사인 유사도를 구한다.



[그림 2] 사용자 아이템 행렬을 기반으로 구축된  
아이템-아이템 행렬

[그림 2]의 (a)와 같은 사용자-아이템 행렬을 기반으로 (b)와 같은 아이템-아이템 간의 유사도를 나타내는 행렬 형태의 유사도 모델을 구축할 수 있다. 아이템-아이템 행렬의 각 값은 해당되는 두 아이템 간의 유사도로 채워지며, 한 사용자가 두 개 이상의 아이템에 대한 점수를 매긴 경우에만 이 유사도 값을 계산할 수 있다. 이 아이템-아이템 행렬을 바탕으로 하여, Sarwar et al.은 예측 점수의 계산 방법으로 가중치 합(Weighted Sum)을 제안했다(Sarwar et al., 2001).

$$P_{u,i} = \frac{\sum_{j \in N} R_{u,j} sim(i, j)}{\sum_{j \in N} |sim(i, j)|} \quad (5)$$

사용자  $u$ 의 아이템  $i$ 에 대한 예측 점수  $P_{u,i}$ 를 구하기 위해, 유사 아이템 집합  $N$ 에 포함된 모든 아이템  $j$ 에 대한  $u$ 의 점수에 아이템  $i$ 와  $j$ 의 유사도를 가중치로 하여 평균한다.

아이템 기반의 협력적 여과는 미리 유사도 모델을 구축하여 사용함으로써 전통적인 방법에 비해 매우 빠른 추천이 가능하다. 또한 아이템 집합의 변화는 사용자 집합의 변화에 비해 매우 적으므로, 이미 구축되어 있는 아이템 유사도 모델을 재사용하더라도 초기의 추천 성능에 비해 정확도가 떨어지지 않는다. 때문에 모델의 구축은 추천과 함께 실시간으로 이루어질 필요가 없다. 이는 하나의 중앙 서버에서 유사도 계산과 추천의 전 과정을 처리하는 부담을 줄일 수 있을 뿐 아니라, 분산된 환경에서의 추천 시스템의 가능성을 보여준다(Miller et al., 2004; Sarwar et al., 2001). 반면 모델을 구축하는 데 많은 시간이 소요되며, 일반적으로 사용자 기반의 협력적 여과 방법에 비해 추천의 질이 떨어지는 경향이 있다(Deshpande and Karypis, 2004).

## 2.2 분산 추천 시스템

지금까지의 추천 시스템에 관련된 연구는 시스템의 중앙에서 모든 정보를 관리하는 중앙 집중적 방식의 시스템이 주류를 이루었다. 그러나 이와 같은 방식의 시스템에서는 추천에 대한 신뢰성과 개인 정보의 남용 가능성에 대한 심각한 문제가 야기될 수 있다(Miller et al. 2004; Sarwar et al., 2001; Kim, 2003).

최근 활발하게 진행되고 있는 분산 컴퓨팅 환경에 관련된 연구 경향에 발맞추어 점차 P2P 방식의 분산 추천 시스템은 이와 같은 문제점을 부분적으로 개선할 수 있는 해결방안 중 하나이다. 중앙 시스템이 아닌 개인이 자신의 정보를 관리하기 때문에 원하지 않는 정보는 공개하지 않을 수 있고, 상업적인 목적의 정보 이용을 방지할 수 있다. 또한 실제 응용에서 사용자 수의 증가에 따라

크게 증가하는 계산량을 분산함으로써 기존 시스템에 비해 큰 이점을 얻을 수 있다(Miller et al., 2004; Kim, 2003).

아이템 기반의 접근 방법은 사용자 기반 협력적 여과 방식에서 발생하는 데이터 집합의 희박성과 계산의 복잡도 문제를 개선하기 위한 방법의 하나로 제안되었으나, 사전에 미리 구축된 아이템 간 유사도 모델을 사용하여 개인화된 분산 추천 시스템의 가능성을 보였다(Sarwar et al., 2001). 또한 본 논문에서 벤치 마크 한 PocketLens는 다양한 분산 환경의 구조를 제시하고 이에 적용할 수 있는 계속적으로 증가하는 아이템 기반의 유사도 계산방법을 제안하였다(Miller et al., 2004).

## 2.3 에이전트를 이용한 접근 방법

분산된 환경에서 에이전트의 사용은 네트워크의 부담을 줄이고, 비동기적이고 자율적인 시스템의 운영이 가능하다는 점에서 매우 유용하다(Lange and Oshima, 1999).

*Yenta*는 Foner가 제안한 최초의 멀티 에이전트 기반의 협력적 여과 시스템으로, 각 에이전트는 사용자의 취향과 선호도를 묘사하는 키워드를 찾아 이를 학습한다. 유사한 취향의 사용자를 찾기 위해 학습된 키워드를 비교하는데, 이 때 에이전트는 사용자 개인의 신원을 요구하지 않는다(Foner, 1997). Good et al.은 개인의 정보 여과 에이전트를 이용하여 여러 에이전트간의 협업에 의한 추천 결과와 커뮤니티 사용자들의 의견을 종합하여 추천을 제공함으로써 추천의 질을 향상시켰다(Good et al., 1999).

## 2.4 신뢰기반 추천 시스템

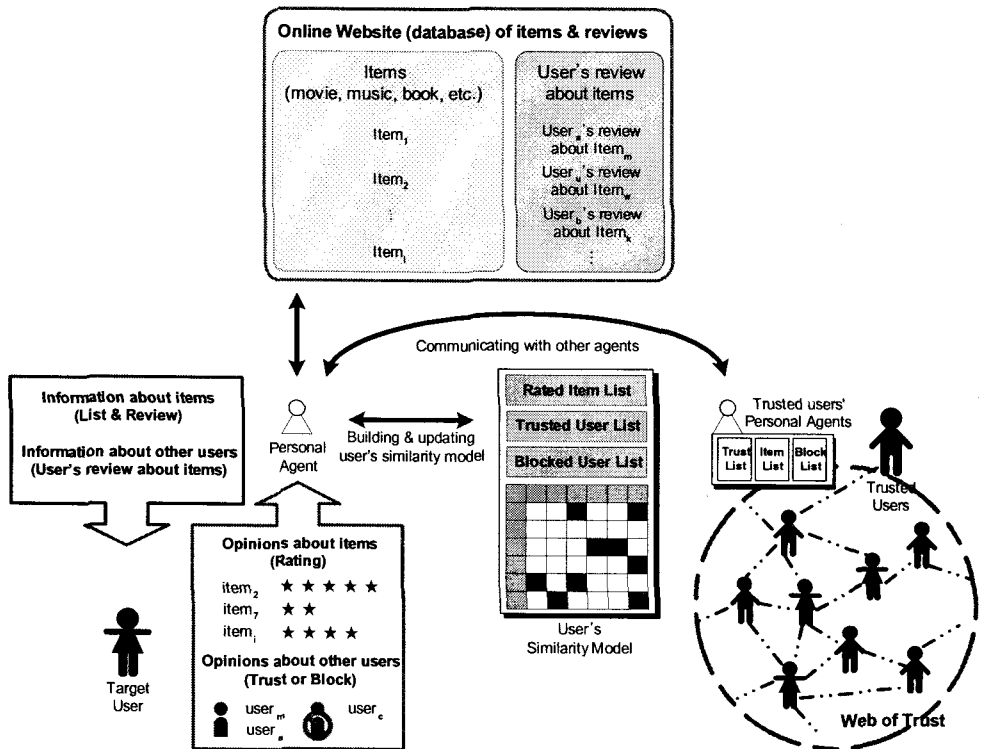
이미 언급한 바와 같이, 온라인 추천 시스템

로부터 얻어지는 추천을 신뢰할 수 있는가에 관련된 문제는 개인화된 분산 추천 시스템으로 일부 해결할 수 있다(Miller et al., 2004). 그러나 이 역시 중앙 집중적 추천시스템과 마찬가지로 shilling attack, 즉 다수의 조작된 프로파일을 생성하여 추천 결과를 조작하려 시도하는 악의적인 사용자의 공격에 노출될 수 있다(Miller et al. 2004). 추천에 참여하는 개인 간 신뢰망(Web of Trust)의 구축은 이와 같은 공격을 효과적으로 방어할 수 있는 중요한 수단 중 하나로 제시되고 있다(Massa and Avesani, 2004; Jung, 2005).

Sinha et al.은 사람들이 일반적인 온라인 추천 시스템에 비해, 친구나 아는 사람의 추천을 더욱 선호하는 경향이 있다는 사실을 보였다(Sinha and

Swearingen, 2001). 또한, Ziegler et al.은 일반적으로 사람들이 임의의 개인 집단 보다는 신뢰하는 개인의 집단과 더욱 유사한 선호도를 보일 가능성이 높다는 것을 언급한 바 있다(Ziegler and Lausen, 2004). 이는 신뢰 정보가 추천에 매우 효과적으로 사용될 가능성이 크다는 것을 의미한다.

그러나 개인이 직접적으로 표현한 신뢰 정보는 매우 희박한 경향이 있다(Guha et al., 2004). 따라서 이 신뢰 정보를 효과적으로 전파하여 좀 더 많은 신뢰할만한 사용자를 추천에 사용하기 위한 많은 연구가 진행되고 있다(Guha et al., 2004; Ziegler et al., 2005). Guha et al.은 몇 가지 기본 전파 원리(Atomic Propagation)를 정의하고 이를 반복적으로 적용하여 최종적으로 신뢰(Trust)와 불신



[그림 3] 멀티 에이전트를 이용한 신뢰기반 협력적 여과 추천 시스템의 개요

(Distrust)정보를 갖는 행렬을 얻는 접근방법을 제안하였다(Guha et al. 2004). Ziegler et al.은 *Advogato trust metric*에 기반한 *Appleseed* 알고리즘을 제안하고, 사회망(Social Network)상에서 효과적으로 신뢰 정보를 전파하면서 악의적 사용자의 집단 공격을 방어할 수 있는 수단으로 사용하였다(Ziegler et al., 2005; Levien and Aiken, 2000).

### 3. 멀티 에이전트 기반의 협력적 여과 추천 시스템

[그림 3]은 본 논문에서 제안하고자 하는 시스템의 대략적인 구조를 보여준다.

우선, 추천의 대상이 되는 아이템의 목록과 해당 아이템에 대한 간단한 기술, 그리고 어떤 특정 아이템을 이미 추천 받았거나 사용한 경험이 있는 사용자들의 해당 아이템에 대한 리뷰를 제공하는 온라인 사이트를 가정한다. 일반적인 온라인 상점에서 제공하는 상품 목록 및 상품에 대한 사용자의 평가 등을 이러한 온라인 사이트의 한 예시로 볼 수 있으며, *Epinion.com*과 같이 상품의 판매가 아닌 평가 및 추천만이 목적인 온라인 사이트 역시 하나의 좋은 예가 될 수 있다.

추천을 받고자 하는 사용자는 각각 자신의 개인 에이전트를 갖는다. 아이템과 다른 사용자에 대한 개인의 의사 표현 정보는 이 개인 에이전트가 관리하며, 사용자는 개인 에이전트를 통해 자신이 정보를 공유하기를 원하는 사용자, 즉 신뢰할 만한 사용자나 신뢰할 수 없는 사용자를 선택하여 공개할 수 있다.

시스템은 크게 네 부분으로 나누어 설명할 수 있다. 우선 첫 번째로 각 사용자의 개인 에이전트가 온라인 사이트를 통해 얻은 정보를 사용자에게

제공하여, 이에 대한 사용자의 의견 정보를 얻어 저장한다. 다음으로 추천을 받을 사용자(목적사용자)의 개인 에이전트가 앞서 얻은 신뢰사용자(Trusted Users)의 목록을 바탕으로 이동 에이전트가 이동할 유일한 경로를 찾는다. 이동 에이전트는 이 경로를 따라 이동하면서 개인 에이전트가 모델을 구축하는데 필요한 정보를 수집하고, 개인 에이전트는 이 정보를 이용하여 사용자를 위한 유사도 모델을 구축한다. 마지막으로 이 유사도 모델을 이용하여 사용자에게 추천을 제공하며, 추천에 대한 사용자의 피드백을 참조하여 유사도 모델을 갱신한다.

#### 3.1 사용자 정보의 획득 및 신뢰망 구성

자세한 기술에 앞서 시스템에서 사용되는 몇 가지 용어에 대해 정의한다.

$P_X$  : 사용자(User)

$P_O$  : 유사도 모델 소유자(Model Owner) = 목적사용자(Target User)

$P_C$  : 현재 이동에이전트가 접근한 사용자(Current User)

$\{AGENT_{P_X}\}$  : 사용자  $P_X$ 가 신뢰하는 사용자의 목록

$\{BLOCK_{P_X}\}$  : 사용자  $P_X$ 가 불신하는 사용자의 목록

$\{ITEMS_{P_X}\}$  : 사용자  $P_X$ 의 <아이템, 선호도 점수> 목록

$PATH_{P_X}$  : 사용자  $P_X$ 의 이동에이전트가 이동할 경로

$AGENT_{P_X}$  : 사용자  $P_X$ 의 개인 에이전트

$AGENT_{P_X}^M$  : 사용자  $P_X$ 의 이동 에이전트



첫 번째 단계로 사용자의 개인 에이전트  $AGENT_{P_x}$  는 온라인 사이트를 통해 아이템의 리스트 및 각 아이템에 대한 간단한 설명, 해당 아이템에 대한 다른 사용자의 리뷰 등을 얻어 사용자에게 제공한다. 사용자  $P_x$  는 개인 에이전트에게서 얻은 아이템 정보를 검토하여 특정 아이템에 대한 선호도를 표현하거나, 다른 사용자의 리뷰로부터 이 사용자가 아이템에 대해 얼마나 유용한 정보를 제공하는가를 판단하여 특정 사용자에 대한 신뢰 정보를 표현할 수 있다. 본 시스템에서는 아이템에 대한 선호도는 1~5사이의 숫자 점수로 표현되며, 사용자에게 대한 신뢰도는 신뢰하는 사용자(Trusted User)와 불신하는 사용자(Distrusted or Blocked User)로 표현될 수 있다.

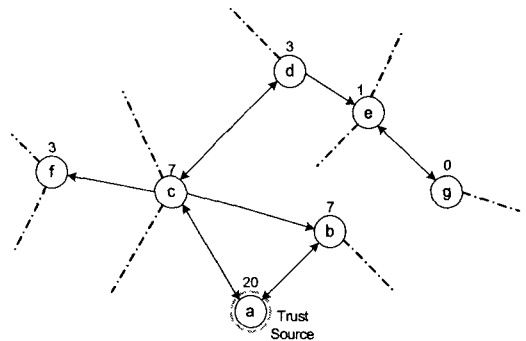
이 신뢰정보를 바탕으로 개인 에이전트 간의 P2P망(Peer-to-Peer Network)을 가정하자. 사용자  $P_x$  가 신뢰하는 사용자들의 집합은 단순한 목록 형태로 저장되며, 이를  $\{TRUST_{P_x}\}$  라 한다. P2P망 내에 존재하는 모든 사용자  $P_x$  의 신뢰하는 사용자 목록  $\{TRUST_{P_x}\}$  를 기반으로 하여 양방향 그래프(Bidirectional Graph) 형태의 신뢰망(Web of Trust)이 구성되며, 서로 간의 아무런 정보 표현이 존재하지 않는 사용자 간에는 간선(edge)이 존재하지 않는다. 사용자  $P_x$  가 선호도를 표현한 아이템과 해당 선호도 점수를 의미하는  $\langle item, rating \rangle$  쌍은  $\{ITEMS_{P_x}\}$  목록에 저장되고, 이 정보는 다른 시스템과는 달리 오직 각 사용자의 개인 에이전트인  $AGENT_{P_x}$  에만 저장한다.

### 3.2 신뢰 정보 기반의 이웃 집단 구성 알고리즘

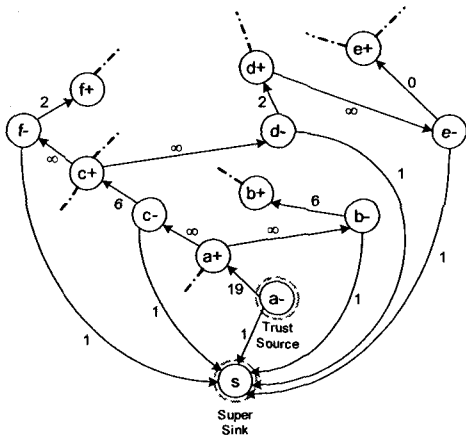
이제  $AGENT_{P_x}$  는 사용자  $P_x$  에게 추천을 제

공하기 위해 유사도 모델에 필요한 정보를 수집할 이동 에이전트  $AGENT_{P_x}$  의 이동경로  $\{PATH_{P_x}\}$  를 찾는다. 이 이동경로를 찾기 위해 *Advogato* 최대 흐름 알고리즘(Advogato Maximum Flow Algorithm)을 사용한다(Levien and Aiken, 2000). 이 알고리즘은 *Ford-Fulkerson* 최대 흐름 알고리즘(Ford Fulkerson Maximum Flow Algorithm)을 기반으로 제안된 것으로, 어떤 사용자가 온라인 커뮤니티 내에서 이미 믿을만하다고 판단된 다른 사용자들에게 신뢰받는가 그렇지 않은가를 판단하기 위한 알고리즘이다(Guha et al. 2004; Levien and Aiken, 2000). 사용자 간의 간선(edge)를 통과하는 최대 흐름을 찾는 과정에서 양방향 그래프 형태의 신뢰망은 목적사용자  $P_x$  를 루트로 하는 트리(tree) 형태로 재구성되어, 이동 에이전트가 중복된 노드를 방문하는 것을 방지하고 검색 범위를 적절한 수준으로 줄인다.

*Ford-Fulkerson* 최대 흐름 알고리즘을 적용하기 위해서는 그래프의 변환이 필요하다. [그림 4]는 한 개의 소스 노드와 여러 개의 목적 노드, 즉 여러 개의 말단 노드를 갖고 노드에 캐퍼시티가 할당된 트리 형태가 된다. 이 그래프를 [그림 5]처럼 한 개의 소스 노드와 한 개의 목적 노드를 갖고 간선에 캐퍼시티가 할당된 형태로 변환한다.



[그림 4] 신뢰망에서의 캐퍼시티 할당



[그림 5] Advogato 알고리즘을 사용하여 변형된 그래프

변환 과정은 [알고리즘 1]과 같다.

$C(P_X)$ 의 캐퍼시티를 갖는 각 노드  $P_X$ 는  $P_{X+}$ 와  $P_{X-}$ 로 분할되고, 두 노드 간의 간선에  $C(P_X) - 1$ 의 캐퍼시티가 할당된다. 한 개의 목적 노드를 갖도록 하기 위해 가상의 목적 노드(Super Sink)를 생성한 후, 각  $P_{X-}$ 노드로부터 목적 노드로의 캐퍼시티 1을 갖는 간선을 생성한다. 변환 이전 그래프에서  $P_X$ 와  $P_Y$  간의 간선은  $P_{X+}$ 와  $P_{X-}$  간의 간선으로 변환되고, 무한대의 캐퍼시티를 갖는다.

$AGENT_{P_0}$ 는 이 그래프에 *Ford-Fulkerson* 최대흐름 알고리즘을 적용하여 소스 노드로부터 목적 노드로의 최단 거리를 찾으면서, 해당되는 노드를  $\{PATH_{P_0}\}$ 에 추가한다([알고리즘 2]).

*Advogato* 알고리즘은 온라인 커뮤니티 외부로부터 악의적인 사용자의 집단 공격을 막기 위해 고안된 알고리즘으로, 추천 결과를 조작하거나 시스템을 붕괴시킬 의도를 갖는 악의적인 사용자의 공격을 차단하는데 매우 유용하다. 조작된 선호도

[알고리즘 1] Advogato 그래프 변형 알고리즘

```

Advogato Graph Transform
function transform ( $G = (V, E, C_V)$ ) {
  set  $E' \leftarrow \emptyset, V' \leftarrow \emptyset$ 
  for all  $x \in V$  do
    add node  $x+$  to  $V'$ 
    add node  $x-$  to  $V'$ 
    if  $C_V(x) \geq 1$  then
      add edge ( $x-, x+$ ) to  $E'$ 
      set  $C_{E'}(x-, x+) \leftarrow C_V(x) - 1$ 
      for all edge ( $x, y$ )  $\in E$  do
        add edge ( $x+, y-$ ) to  $E'$ 
        set  $C_{E'}(x+, y-) \leftarrow \infty$ 
      end do
      add edge ( $x-, supersink$ ) to  $E'$ 
      set  $C_{E'}(x-, supersink) \leftarrow 1$ 
    end if
  end do
  return  $G' = (V', E', C_{E'})$ 
}
    
```

[알고리즘 2] Ford Fulkerson 최대흐름 알고리즘

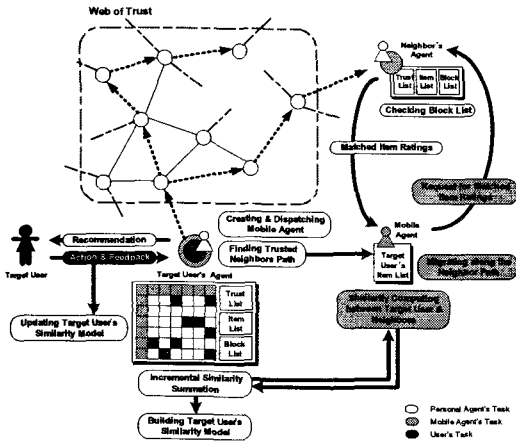
```

Ford-Fulkerson Maxflow Algorithm
function maxflow ( $G, source, supersink$ ) {
  for each edge ( $x, y$ )  $\in E'$  in  $G'$  do
     $F(x, y) \leftarrow 0$ 
     $F(y, x) \leftarrow 0$ 
  end do
  while there exists a path  $P$  from  $source$  to  $supersink$ 
  in the residual network  $G_F$  do
     $C_F(P) \leftarrow \min \{C_F(x, y) : (x, y) \text{ in } P\}$ 
    for each edge ( $x, y$ ) in  $P$  do
       $F(x, y) \leftarrow F(x, y) + C_F(P)$ 
       $F(y, x) \leftarrow -F(x, y)$ 
    end do
  end while
}
    
```

점수를 갖는 다수의 악의적인 사용자가 존재한다 하더라도, 이동 에이전트가 방문할 경로에 포함될 가능성이 있는 악의적인 노드의 개수는 악의적인 사용자 노드의 전체 개수가 아닌 악의적인 노드로

의 간선을 가지고 있는 노드의 개수에 의해 결정된다. 때문에 소스 노드의 사용자로부터 악의적인 사용자로의 직접적인 신뢰 표현이 존재하지 않는다고 가정할 때, 조작된 선호도 점수는 추천에 거의 영향을 미치지 못한다(Guha et al., 2004; Levien and Aiken, 2000).

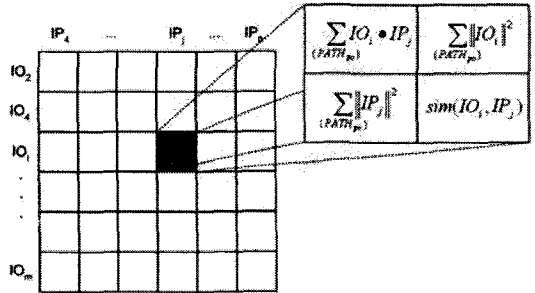
### 3.3 유사도 모델의 점진적인 구축



[그림 6] 유사도 구축의 개략적인 과정

목적사용자  $P_O$ 의 유사도 모델 구축을 위해  $AGENT_{P_b}$ 는 이동 에이전트  $AGENT_{P_b}^M$ 를 생성한다.  $AGENT_{P_b}^M$ 는 이동 경로에 포함된 사용자 노드의 목록인  $\{PATH_{P_b}\}$ 와  $P_O$ 의 선호 아이템과 해당 아이템의 선호도 점수의 목록인  $\{ITEMS_{P_b}\}$ 를 포함한다. 변형된 트리의 탐색은 깊이 우선 탐색 알고리즘(Depth First Search Algorithm)을 사용한다.

$P_C$ 를  $AGENT_{P_b}$ 가 현재 방문한 사용자 노드라 하면,  $AGENT_{P_b}$ 는  $\{PATH_{P_b}\}$ 와  $\{TRUST_{P_b}\}$ 에 공통적으로 포함된 노드가 더 이상 존재하지 않을 때까지 경로를 탐색한다.  $\{PATH_{P_b}\}$ 에 더 이상 노드



[그림 7] 분산환경에서 계속적으로 증가하는 유사도의 계산

가 존재하지 않으면  $AGENT_{P_b}^M$ 는 제거된다.

분산된 환경의 추천을 받을 각 사용자를 위한 유사도 모델 구축을 위해 유사도 계산에 필요한 변수를 [그림 7]과 같이 저장하여 사용하며(Miller et al., 2004), 각 에이전트는 다음과 같은 과정을 수행한다.

- (1)  $AGENT_{P_b}^M$ 는 현재 방문한 사용자 노드  $P_C$ 의 개인 에이전트  $AGENT_{P_C}$ 와의 의사소통을 통해,  $P_O$ 와  $P_C$ 가 모두 점수를 매긴 아이템의 집합을  $IO$ , 현재 방문 노드의 사용자만 점수를 매긴 아이템 집합을  $IP$ 로 정의한다.

$$IO = \{ITEMS_{P_C}\} \cap \{ITEMS_{P_O}\}$$

$$IP = \{ITEMS_{P_C}\} - \{ITEMS_{P_O}\}$$

- (2)  $IO_i \in IO, IP_j \in IP$  인 모든 아이템 쌍( $IO_i, IP_j$ )에 대해,  $AGENT_{P_b}^M$ 는 식 (6)의 값을 계산한 후 결과값을  $AGENT_{P_b}$ 에게 전달한다.  $Rating_{P_C, IO_i}$ 는 사용자  $P_C$ 가 아이템  $IO_i$ 에 매긴 선호도 점수를 의미한다.

$$W_{IO_i, IP_j} = (Rating_{P_C, IO_i} \times Rating_{P_C, IP_j})$$

$$W_{IO_i, IO_i} = (Rating_{P_C, IO_i})^2$$

$$W_{IP_j, IP_j} = (Rating_{P_C, IP_j})^2 \quad (6)$$

(3)  $AGENT_{P_0}^M$ 가  $\{PATH_{P_0}\}$  내에 존재하는 모든 사용자 노드에서 식 (6)의 값을 계산하여  $AGENT_{P_0}$ 에게 전달하면,  $AGENT_{P_0}$ 는 이 값을 식 (7)과 같이 계속적으로 합산한다.

$$\begin{aligned} W_{Nmer} &= W_{Nmer} + W_{IO_i, IP_j} & (7) \\ W_{Denom1} &= W_{Denom1} + W_{IO_i, IO_i} \\ W_{Denom2} &= W_{Denom2} + W_{IP_j, IP_j} \end{aligned}$$

(4)  $AGENT_{P_0}$ 는 합산된 값을 이용하여 식 (8)과 같이  $(IO_i, IP_j)$ 의 유사도를 계산한다.

$$Sim(IO_i, IP_j) = \frac{W_{Nmer}}{\sqrt{W_{Denom1}} \sqrt{W_{Denom2}}} \quad (8)$$

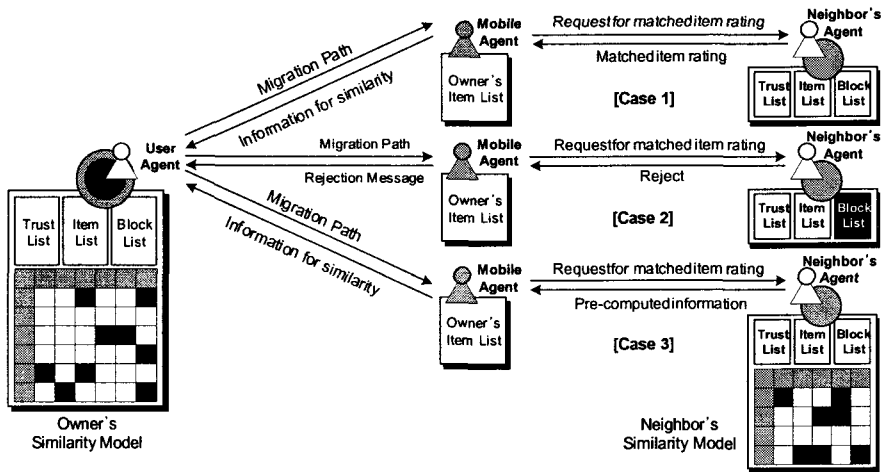
위의 과정은 코사인 유사도를 계산하는 방법으로, 과정 2를 간단히 수정하여 다른 여러 가지 유사도 계산 방법으로 변경할 수 있다. 예를 들어, 아이템 간의 피어슨 상관관계를 계산하기 위해 제안된 방법인 수정된 코사인 유사도(Adjusted cosine similarity)를 계산하기 위해(Sarwar et al.,

2001), 식 (6)은 식 (9)와 같이 변경될 수 있다.

$$\begin{aligned} W_{IO_i, IP_j}' &= (Rating_{P_0, IO_i} - AvgRating_{P_0}) \\ &\quad \times (Rating_{P_0, IP_j} - AvgRating_{P_0}) \\ W_{IO_i, IO_i}' &= (Rating_{P_0, IO_i} - AvgRating_{P_0})^2 \\ W_{IP_j, IP_j}' &= (Rating_{P_0, IP_j} - AvgRating_{P_0})^2 \quad (9) \end{aligned}$$

일반적으로  $AGENT_{P_0}$ 는 위 과정을 통해 유사도 모델을 구축하고 갱신할 수 있다. 그러나  $\{PATH_{P_0}\}$  내의 노드를 탐색하는 과정에서 방문한 각 노드의 현재 상태에 따라 유사도 모델 구축의 과정은 변경될 수 있다.

[그림 8]은 탐색 시 발생할 수 있는 세 가지 경우를 도식화 한 것으로 [Case 1]은 일반적인 경우를 나타낸다. [Case 2]에서와 같이, 현재 방문 노드의 사용자  $P_C$ 가 불신한다고 표현한 사용자의 목록을  $\{BLOCK_{P_C}\}$ 라 하고,  $P_0$ 가  $\{BLOCK_{P_C}\}$ 에 포함되어 있다고 하자.  $P_C$ 는  $P_0$ 를 불신하기 때문에 자신의 정보가  $P_0$ 에게 공개되는 것을 원



[그림 8] 방문한 사용자의 상태에 따른 모델 구축

하지 않을 것이다. 때문에  $P_C$ 의 개인 에이전트인  $AGENT_{P_C}$ 는  $AGENT_{P_O}^M$ 의 정보 요청을 거절한다. [Case 3]은 현재 방문 노드인  $P_C$ 가 이미 자신의 유사도 모델을 소유하고 있는 경우이다.  $AGENT_{P_O}^M$ 는 이미 구축되어 있는  $P_C$ 의 유사도 모델로부터  $P_C$ 의 하위 노드로부터 얻을 수 있는 각 아이템 쌍의 유사도 정보를 얻을 수 있기 때문에,  $P_C$ 의 하위 노드를 더 이상 탐색할 필요가 없다.  $P_C$ 의 유사도 모델은  $P_C$ 를 소스 노드로 하는  $\{PATH_{P_C}\}$ 를 기반으로 구축된 모델로서,  $\{PATH_{P_C}\}$ 에 포함된 노드는  $\{PATH_{P_O}\}$ 에 포함된 노드와 거의 유사하며,  $P_C$ 가  $P_O$ 에서 가까운 거리에 위치한 노드일수록 더욱 유사하다. 이 경우,  $AGENT_{P_O}^M$ 는 초기의 의도보다 더 많은 수의 사용자 노드를 방문하는 효과를 얻게 되며, 때문에  $P_O$ 의 모델은 더 많은 사용자의 정보를 포함한다. 게다가,  $P_C$ 의 모델에 포함된 사용자는  $AGENT_{P_O}$ 가 이미 방문했거나, 앞으로 방문할 예정인 사용자와 중복될 수 있다. 그러나, 이 “중복된 사용자”는 상위 노드의 사용자로부터 더 많이 신뢰받고 있는 사용자로 볼 수 있으며, 이들의 정보를 더 많이 반영함으로써 오히려 추천에 긍정적인 영향을 줄 수 있다. 또한, 앞으로 방문해야 할 노드의 수를 크게 감소시킬 수 있기 때문에 탐색 비용이 크게 절감된다.

### 3.4 사용자 피드백의 전파와 모델 갱신

위와 같은 과정을 통해 구축된 유사도 모델을 기반으로 목적사용자  $P_O$ 의 개인 에이전트  $AGENT_{P_O}$ 는  $P_O$ 에게 적절한 추천을 제공할 수 있다. 행렬로 표현된 유사도 모델에서, 단순히

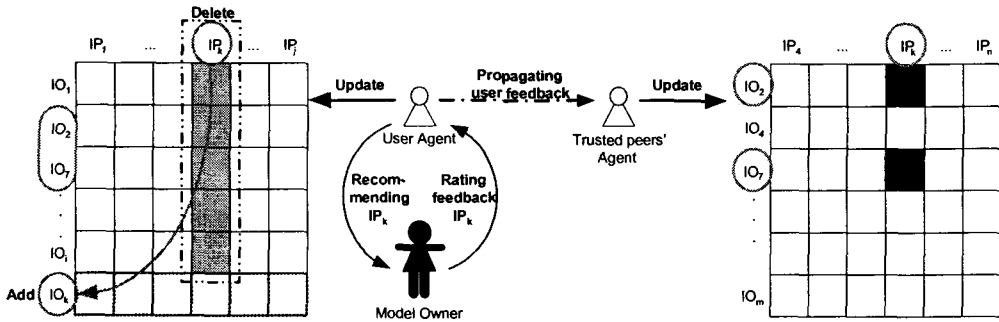
$IP_j \in IP$ 인 특정 아이템  $IP_j$ 에 대한 각 열의 값의 평균을 구하여 높은 평균 값을 갖는 아이템을 추천하거나, 식 (10)을 이용하여  $IP_j$ 에 대한 목적사용자  $P_O$ 의 명시적인 예측 선호도 점수를 구할 수 있다.

$$p\_rating_{P_O, IP_j} = \frac{\sum_{IO_i \in IO} \{sim(IO_i, IP_j) \times Rating_{P_O, IO_i}\}}{\sum_{IO_i \in IO} |sim(IO_i, IP_j)|} \quad (10)$$

모델 기반의 접근 방법에서 주요하게 다루어져야 할 문제점 중 하나는 한 번 모델이 구축되고 사용된 후, 다시 구축되기 전까지의 기간 동안에 일어나는 데이터의 변경을 모델에 어떻게 반영하여 갱신할 것인가 하는 것이다.

아이템  $IP_k$ 가 목적사용자  $P_O$ 에게 추천 되었을 때,  $P_O$ 는 해당 아이템에 대한 선호도 점수를 표현할 수 있다. 이러한  $P_O$ 의 피드백이 발생하면  $AGENT_{P_O}$ 는  $IP_k$ 를  $\{ITEMS_{P_O}\}$ 에 추가한다. 갱신된 정보, 즉  $\langle IP_k, IP_k \text{에 대한 } P_O \text{의 선호도 점수} \rangle$ 를 포함하는  $\{ITEMS_{P_O}\}$ 를  $P_O$ 가 신뢰하는 사용자의 목록  $\{ITEMS_{P_O}\}$ 에 포함된 사용자들의 개인 에이전트에게 전달한다.

$P_C$ 를  $\{TRUST_{P_O}\}$ 에 포함된 사용자라 하자. 그리고  $P_C$ 의 관점에서 볼 때, 아이템  $IP_k$ 와  $\{ITEMS_{P_C}\} \cap \{ITEMS_{P_O}\}$ 에 포함된 아이템 즉,  $P_C$ 와  $P_O$ 가 동시에 선호도 점수를 표현한 아이템들과의 유사도는  $P_C$ 의 유사도 모델에 포함되어 있지 않다고 가정하자. 이 경우  $AGENT_{P_C}$ 는 전달받은  $\langle IP_k, IP_k \text{에 대한 } P_O \text{의 선호도 점수} \rangle$ 를 이용하여 이들 간의 유사도를 계산한 후, 모델을 부분적으로 갱신하게 된다. 그리고  $AGENT_{P_C}$ 는  $P_C$ 가 신뢰하는 사용자의 목록인  $\{TRUST_{P_C}\}$ 에 포함된 사용자들에게 해당 정보를



[그림 9] 아이템의 추천과 사용자 피드백의 전파

전파하고, 이를 이용하여 지속적인 갱신이 이루어질 수 있다. [그림 9]는 이와 같은 과정을 나타낸다.

#### 4. 실험 및 결과

이번 장에서는 본문에서 제안한 신뢰도 기반의 협력적 여과 기법을 P2P 환경에서의 추천에 적용한 결과를 보인다. 우선, *PocketLens*에서 제안한 시스템 구조 중 하나로 분산된 사용자들 임의적으로 탐색하여 유사도 모델 구축에 사용하는 가장 일반적인 방식의 P2P 분산 추천 시스템과 본 논문에서 제안하는 시스템의 전반적인 성능을 평균 절대 오차(Mean Absolute Error)를 이용하여 비교한다.

그리고, 멀티 에이전트와 신뢰 정보를 사용함으로써 얻을 수 있는 유사도 모델 구축의 효율성과 약의적인 의도를 가진 사용자의 공격에 대한 시스템의 안정성에 대한 실험을 통해 제안하는 시스템의 우수성을 보인다.

##### 4.1 실험 데이터와 성능 측정법

*Epinion.com*은 사용자가 여러 아이템을 평가하고 아이템에 대한 선호도를 1부터 5사이의 점수로

표현할 수 있도록 한 온라인 커뮤니티이다. 또한, 사용자는 다른 사용자의 상품평을 읽고, 그 정보가 얼마나 자신에게 효용가치가 있는가를 평가하여 그 사용자에 대한 신뢰도를 표현할 수 있다.

실험을 위해 2006년 5월까지의 *Epinion.com*에서 제공하는 데이터를 수집하였으며, 수집된 데이터 집합은 48,058명의 사용자 간의 479,167개의 신뢰 정보와 이들이 172,414개의 아이템에 대해 선호도를 표현한 선호도 점수 662,363개를 포함한다. 이 데이터 집합에서 최소 5개 이상의 아이템에 점수를 표현하고 최소 25명의 다른 사용자에게 신뢰 정보를 표현한 사용자와 최소 10명 이상의 사용자에게 점수를 받은 아이템을 선별하여, 데이터의 희박성을 극복하고 좀 더 효과적인 실험을 수행할 수 있도록 했다. 사용자-아이템 행렬 내에 얼마만큼의 선호도 점수가 포함되어 있는가를 판단할 수 있는 희박성 수준(Sparsity Level)은  $1 / (\text{ratings} / \text{users} * \text{items})$ 의 식으로 나타낼 수 있으며, 정제된 데이터 집합의 희박성 수준은 0.9913이다. 정제된 데이터 집합은 다음과 같다.

[표 2] 실험에 사용된 데이터 집합

users	trusts	items	rating
4,751	216,490	2,955	121,862

데이터 집합은 두 부분으로 나뉘어지는데, 학습 데이터는 모든 각각의 사용자에게 대해 한 개의  $\langle item, rating \rangle$  쌍을 제외한 모든 데이터이며, 각 사용자마다 제외된 한 개의  $\langle item, rating \rangle$  쌍은 평가 데이터로 사용되어 해당 아이템에 대한 사용자의 선호도 점수를 예측하도록 한다.

성능 평가 방법으로는 다른 여러 알고리즘에서 얼마나 정확한 예측을 했는가에 대한 평가 방법으로 가장 많이 사용되는 평균 절대 오차를 사용했다(Miller et al. 2004; Sarwar et al., 2001; Massa and Avesani, 2004; Herlocker et al., 1999; O'Donovan and Smyth, 2005; Deshpande and Karypis, 2004). 평가 데이터 집합에 포함된 모든 사용자에게 대한 평균 절대 오차는 다음과 같이 정의할 수 있다.

$$MAE = \frac{\sum_{i \in M} |p\_rating_i - a\_rating_i|}{|M|} \quad (11)$$

$M$ 은 평가 데이터에 포함된 전체 아이템 집합,  $\langle a\_rating_i, p\_rating_i \rangle$ 는 평가 데이터 집합에 포함된 아이템  $i$ 에 대한 사용자의 실제 표현한 점수(Actual Rating)와 예측된 점수(Predicted Rating)를 의미한다.

악의적인 데이터가 시스템에 미치는 영향의 정도를 측정하기 위해 IOS(Influence on the Systems)를 사용한다. 정상적인 경우 예측되어야 하는 선호도 점수와 악의적인 데이터가 포함되었을 때 예측된 점수의 차를 평균하여 얻어지는 값으로 다음과 같이 정의된다.

$$IOS = \frac{\sum_{i \in M} |p\_rating_i - p\_rating'_i|}{|M|} \quad (12)$$

$p\_rating$ 는 원래의 학습 데이터 집합을 이용하여 모델을 구축하였을 경우 아이템  $i$ 에 대한 예

측점수,  $p\_rating'$ 는 학습 데이터 집합에 악의적인 데이터가 포함되었을 경우 아이템  $i$ 에 대한 예측 선호도 점수를 의미한다.

## 4.2 실험과 분석

### 4.2.1 이웃크기의 증가에 따른 성능 평가

우선, 전반적인 성능 면에서 제안하는 시스템의 우수성을 보이기 위해 유사도 모델 구축에 필요한 이웃의 크기를 증가시키면서 모델 구축과 선호도 점수 예측의 과정을 수행한다. 실험에 앞서, 각 사용자의 선호도 점수의 벡터  $\vec{r}$ 을  $\|\vec{r}\| = 1$ 이 되도록 정규화한다. 이는 많은 수의 아이템에 선호도 점수를 표현한 사용자가 비교적 적은 수의 아이템에 선호도 점수를 표현한 사용자에게 비해 유사도 모델에 더 많은 영향을 미치는 것을 방지하기 위해서이다(Miller et al., 2004). 예외적으로 수정된 코사인 유사도(Adjusted-cosine Similarity)의 식을 사용할 경우에는 정규화하지 않은 값을 사용한다.

유사도 모델 구축의 과정에서, 이동 에이전트는 본문에서 제안하는 방법에 의해 트리 형태로 변형된 신뢰망 그래프를 따라 이동하면서 이웃 사용자 노드를 탐색한다. 이웃 사용자의 선호도 아이템과 선호도 점수 정보를 바탕으로 목적사용자와의 의사소통을 통해 아이템 간의 유사도를 계산하고 유사도 모델을 구축한다. 유사도의 계산에는 코사인 유사도(TCFMA+cosine, 식 (6))와 수정된 코사인 유사도(TCFMA+adjusted, 식 (9))를 사용한다.

이를 신뢰망에 관계없이 임의적으로 이웃 사용자를 탐색하고 유사도의 계산에는 코사인 유사도를 사용하는 임의 접근 방법(Random)과 비교하였다. 이 때, 이웃 크기는 한 아이템 쌍의 유사도를 구하기 위해 필요한 최소 이웃의 수를 의미한다.

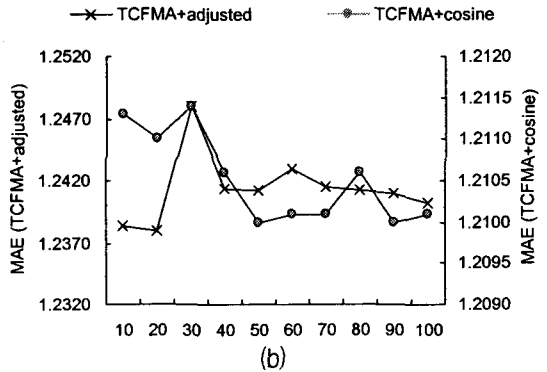
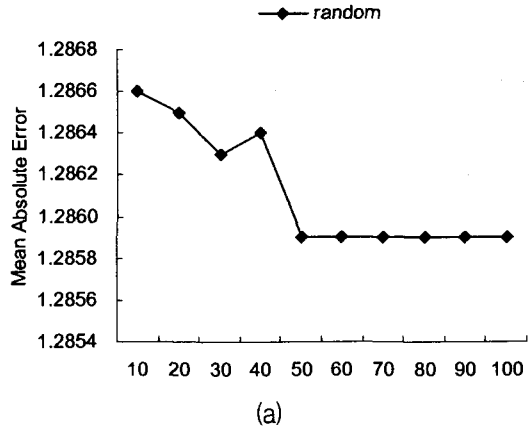
즉, 이웃 크기가 10일 때 아이템  $i$ 와  $j$ 의 유사도를 구하는 경우,  $i$ 와  $j$ 두 아이템에 모두 선호도 점수를 표현한 사용자가 10명 이상이 될 때까지 정보를 수집하여 유사도를 구한다는 의미이다.

<표 3> 이웃 크기 변화에 따른 시스템의 성능 비교

Neighbor peer size	10	30	50	70	100
Random	1.2866	1.2863	1.2859	1.2859	1.2859
TCFMA + cosine	1.2113	1.2114	1.2100	1.2101	1.2101
TCFMA + adjusted	1.2384	1.2480	1.2412	1.2415	1.2402

학습 데이터에 포함된 각 사용자를 목적사용자로 하는 유사도 모델을 구축한 후, 각 목적사용자에 대해 평가 데이터 집합에 속한 아이템의 예측 점수를 구하고, 이를 실제 점수와 비교하여 평균 절대 오차를 구한다. 결과는 <표 3>과 같다. <표 3>에서 볼 수 있듯이, 임의적인 이웃 탐색(Random)을 통해 구축된 유사도 모델은 제안하는 방법으로 구축된 모델에 비해 평균 절대 오차의 값이 크다. 즉, 실험에 사용한 모든 이웃 집단에 대해 제안된 시스템의 추천의 정확도가 비교적 높다는 사실을 확인할 수 있다.

[그림 10]의 (a)와 (b)에서, 세 가지 경우 모두 이웃의 크기가 증가함에 따라 평균 절대 오차의 값이 점점 감소하다가, 이웃 크기가 50일 때를 전후해서 그 감소의 정도가 완만해 지는 것을 확인할 수 있다. 그러나 (a)의 임의의 이웃 탐색의 경우 이웃의 크기가 작은 경우 추천의 질이 현저하게 떨어지지만, (b)의 제안하는 방법을 사용했을 때 적은 수의 이웃 정보를 이용하더라도 비교적 안정적인 유사도 모델을 구축할 수 있다는 사실을 알 수 있다.



[그림 10] 이웃 크기의 변화에 따른 시스템의 전반적인 성능 비교

#### 4.2.2 유사도 모델 구축의 효율성

유사도 모델 구축의 과정에서 제안된 방법의 경우 이동 에이전트가 미리 설정된 경로를 따라 이동하면서 경로에 포함된 사용자에 대한 계산 결과를 모델 소유자의 개인 에이전트에게 보내는 반면, 임의 탐색의 경우에는 구축이 완료될 때까지 모델 소유자와 임의의 모든 사용자 간의 요청과 응답 메시지가 계속적으로 교환되어야 한다. 또한, 미리 발견된 경로에 포함된 사용자는 목적사용자와 유사한 취향을 갖게 될 가능성이 더 높기 때문



<표 4> 모델 구축에 요구되는 평균 시간과 평균 접근 사용자 수

Peer_id	Required Time (msec)		# Accessed Peers	
	Random	TCFMA + cosine	Random	TCFMA + cosine
ad***jm	17088.60	7093.33	4106.30	893.26
al***isit	15793.10	6657.47	3751.23	563.33
al***e2	17843.13	14953.13	4390.33	2844.47
am***ar	18696.46	4008.48	4393.80	299.99
ar***m	17898.93	4511.93	4413.47	477.37
ar***rg	15295.40	10650.40	3682.47	1495.53
bp***y	16747.40	15285.80	4073.93	2841.23
de***y	16244.77	16034.33	4044.33	3114.13
ge***p	14003.61	7409.92	3418.29	977.41
in***off	18024.47	14060.37	4519.17	2593.57
lm***ia	10766.77	5980.57	2720.17	854.93
lo***e	18679.48	5557.42	4503.70	637.11
mj***in_la	13978.20	12399.80	3425.63	1982.80
ob***n69	9164.53	8807.43	2149.57	1076.67
sh***el575	12822.38	5512.68	3078.62	631.01
sh***e_	17980.70	15036.53	4516.03	2910.13
ter***ain	17948.90	16840.53	4449.97	3479.60
th***13	16368.80	14829.63	4050.60	2689.73
th***erkat	16358.73	11331.23	3999.13	1705.30
tj***ms	9852.07	6964.60	2398.97	1005.43
Average	15577.82	10196.28	3804.29	1653.65

에 시스템으로 하여금 불필요한 탐색을 줄이는 데 도움을 줄 수 있다.

제안된 방법이 유사도 모델 구축 과정에서 얼마나 더 효율적인가에 대해 알아보기 위해, 모델이 구축되는데 걸리는 시간과 필요한 사용자 수의 평균을 비교하였다. 비교를 위해 평가 데이터 집합에서 임의의 20명의 사용자를 선별하여, 이웃 집단의 크기가 각 아이টে에 대해 모두 30 이상이고, 이 아이টে에 중 하나라도 이웃 집단의 크기가 50이 될 때까지의 시간과 접근 사용자 수를 측정하였다. 각 사용자에 대한 각각 유사도 모델의 구축

을 30번 반복하여 평균을 얻었으며, 실험 결과는 <표 4>와 같다.

<표 4>에서 20명의 사용자의 평균 모델 구축 시간을 비교하면, 임의 탐색의 경우 15577.82 msec, 제안된 방법의 경우 10196.28 msec로 제안된 방법의 성능이 뛰어난 것을 볼 수 있다. 사용자 *am\*\*\*ar*의 경우, 모델 구축 시간이 각각 18696.46 msec, 4008.48 msec로 현저한 차이를 보인다. 그러나, 사용자 *de\*\*\*y*와 같이 모델 구축 시간이 각각 16244.77 msec, 16034.33 msec로 모델 구축 시간의 차가 크게 나타나지 않은 경우도 관측되었는

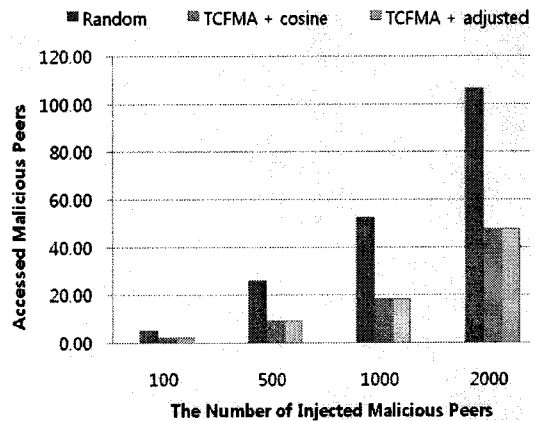
데, 이는 사용자  $de^{***y}$ 의 신뢰 경로에 포함된 사용자, 즉 이동 에이전트에 의해 미리 발견된 경로에 포함된 사용자들이  $de^{***y}$ 와 공통으로 선호하는 아이템을 많이 가지고 있지 않은 경우로 볼 수 있다. 따라서 모델 구축을 위해 보다 많은 사용자에 접근하기 때문에 random한 방법과 비슷한 모델 구축 시간이 소요되었다. 그러나 대부분의 사용자에 대해서 큰 차이가 발생하므로, 미리 발견된 경로에 포함된 사용자가 이웃이 될 가능성이 높다는 것 또한 알 수 있다.

모델 구축 시 접근하는 사용자의 수를 비교하면, 평균 3804.29명과 1653.65명으로 두 배 이상의 큰 차이를 보인다. 이는 모델 구축 시 소요 시간과 같은 맥락에서 분석할 수 있는데, 미리 발견된 신뢰 경로에 포함된 사용자는 이웃이 될 가능성이 높기 때문에 불필요한 이웃의 탐색이 요구되지 않는다. 따라서 임의 탐색의 경우보다 적은 사용자에 접근하여도 모델 구축에 필요한 충분한 정보를 모두 얻을 수 있다.

### 4.2.3 악의적인 공격에 대한 안정성

악의적인 공격이 있을 경우 시스템이 받는 영향을 알아보기 위해, 임의의 조작된 사용자를 학습 데이터 집합에 삽입하여 성능을 알아보았다. 조작된 사용자는 각각 임의의 50개 아이템에 대한 선호도 점수를 갖고, 각 사용자가 다른 사용자를 신뢰한다는 정보를 가진다. 반면 원래의 학습 데이터 집합에 포함된 사용자는 일부만이 이 조작된 사용자들에게 신뢰 정보를 표현한 것으로 한다. 앞선 실험과 마찬가지로 이웃 집단의 크기는 성능 변화의 폭이 완만해지기 시작하는 50으로 고정하고, 조작된 사용자의 수를 100명부터 2000명까지 증가시키면서 실험하였다.

제안된 두 가지 방법인  $TCFMA+cosine$ ,  $TCFMA+adjusted$ 의 모델 구축을 임의 탐색을 통한 모델 구축과 비교하는데, 각 모델이 구축될 때 목적 사용자의 이동 에이전트가 접근하는 조작된 사용자의 평균 수를 알아보고, 조작된 사용자가 삽입되었을 때의 시스템의 왜곡 정도를  $IOS$  측정법 식 (12)을 통해 알아보았다.

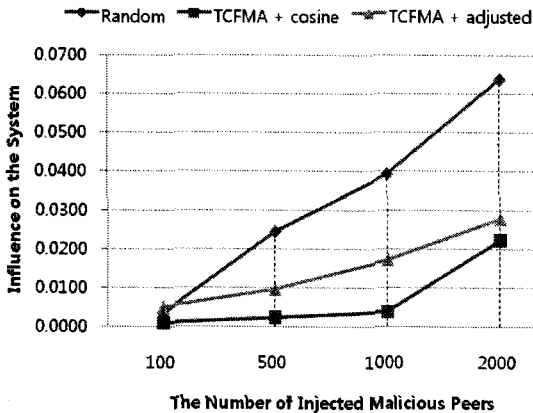


[그림 11] 악의적인 사용자의 증가에 따른 평균 접근 수

[그림 11]에서 볼 수 있듯이, 임의 탐색의 경우 제안된 방법에 비해 두 배 이상 많은 조작된 사용자에 접근하게 된다. 제안된 두 가지 방법( $TCFMA+cosine$ ,  $TCFMA+adjusted$ )은 이웃 집단을 탐색하는 방법은 같고 아이템 간의 유사도를 계산하는 방법만 다르기 때문에 조작된 사용자에 접근하는 횟수가 정확히 같다. 특히, 조작된 사용자가 500명, 1000명일 때 이에 접근하는 수는 임의 탐색의 경우 각각 26.05, 52.37명, 제안된 두 가지 방법의 경우 9.42, 18.31명으로 세 배에 가까운 차이를 보인다.

수집을 통해 얻은 실험 데이터만을 사용하는 일반적인 경우와 이 데이터 집합에 조작된 사용자

가 삽입된 경우의 예측 점수의 차이를 측정하여, 시스템에 의해 제공되는 추천이 변경되는 정도를 측정할 수 있는데, 식 (12)에서 정의한 바와 같이 이를 *IOS* 수치라 한다. [그림 12]와 같이, 임의의 탐색의 경우 삽입된 조작된 사용자의 수가 증가함에 따라 *IOS* 수치가 크게 증가한다.



[그림 12] 악의적인 사용자의 공격이 시스템에 미치는 영향

[그림 11]의 실험 결과를 함께 고려하면, 조작된 사용자에 더 많이 접근하여 이들의 정보를 얻어오는 횟수가 많아지기 때문에 시스템에 미치는 영향이 더 크다고도 볼 수 있다. 제안된 두 가지 방법의 경우 *TCFMA+adjusted*에 비해, *TCFMA+cosine*의 방법을 사용한 경우가 시스템에 미치는 영향이 더 적은 것을 볼 수 있다. 이미 언급한 바와 같이, 이 두 가지 방법은 같은 방법으로 사용자를 탐색하기 때문에 같은 수의 조작된 사용자에 접근하지만 아이템 간의 유사도를 계산하는 방법이 다르기 때문에 이러한 차이를 나타내게 된다. 2000명의 조작된 사용자를 삽입한 경우를 살펴 보면, 제안된 방법으로 유사도 모델을 구축할 때 *TCFMA+cosine*와 *TCFMA+adjusted*의 방법

이 각각 0.0224와 0.0278의 *IOS* 수치를 보이는 반면, 임의의 탐색으로 모델을 구축할 경우 0.0639의 *IOS* 수치를 보인다. 두 가지 실험 결과를 모두 종합하면, 조작된 사용자, 즉 시스템에 영향을 끼치려는 목적으로 삽입되거나 조작된 데이터가 존재할 경우 제안된 시스템이 약 두 배 이상의 안정성을 보인다는 것을 알 수 있다.

## 5. 결론 및 향후 연구

P2P 환경에서의 협력적 여과 추천 시스템은 현재 다양한 응용에 대한 연구가 활발히 진행되고 있는 분야이다. 본 논문에서는 협력적 여과 추천 시스템에서 발생할 수 있는 개인 정보의 오남용과 악의적인 사용자의 공격에 관련된 문제점을 해결하기 위한 새로운 방법을 제안하였다. 온라인 추천 시스템으로부터 생성된 추천의 신뢰성과 정확도를 높이기 위해 사용자 간의 신뢰 정보를 사용하였으며, 멀티 에이전트를 사용하여 신뢰망 내에서 효과적으로 신뢰 정보를 전파하고 분산된 유사도 계산의 효율성을 높이려고 하였다.

우선, P2P 환경에서의 멀티 에이전트 이용의 이점으로, 모델 구축의 효율성 면에서 제안된 방법을 사용했을 경우 유사도 모델 구축 시에 필요한 이웃 집단의 수와 모델 구축 시간을 크게 단축하였다.

또한, 추천의 정확성 면에서 제안된 방법은 일반적인 P2P 방식에 비해 낮은 오차율을 보였으며, 비교적 적은 사용자의 정보를 추천에 이용했을 경우에도 보다 정확한 추천을 얻을 수 있었다.

마지막으로 본 시스템은 악의적인 사용자의 공격에 대한 시스템의 안정성 면에서, 일반적인 P2P 방식의 시스템에 비해 보다 안정적이었다. 임의의

악의적인 사용자가 삽입된 경우, 유사도 모델 구축 시 이 악의적인 사용자에게 접근하는 횟수와 시스템에 미치는 부정적 영향의 정도를 의미하는 IOS 수치가 일반적인 방법에 비해 두 배 이상 낮게 나타났다.

그러나, 실험에 사용된 *Epinion.com*의 데이터 집합은 사용자-아이템 행렬의 밀도가 지나치게 낮아 이에 따른 어려움이 많았다. 따라서 이와 같이 희박성 수준이 높은 실 세계 데이터에 좀 더 실질적으로 적용할 수 있도록 시스템의 보완이 요구된다는 결론을 얻을 수 있었다.

향후 연구로 신뢰 정보의 전파 과정에서 해결되어야 할 중요한 문제점이 제시될 수 있다. 본 논문에서 제안된 시스템의 경우 이동 에이전트가 이동할 경로로서 발견된 모든 사용자에게 대해, 그 레벨이 다른 경우에도 그 신뢰도의 정도가 같은 것으로 취급하였기 때문에 유사도 모델에 동등한 영향을 미친다. 그러나 실 세계에서 신뢰 정보는 여러 사용자의 단계를 따라 전파되어감에 따라 점점 그 신뢰의 정도가 감소하게 된다. 이러한 신뢰도의 감소(Trust Decay)를 시스템에 반영하기 위한 적절한 신뢰 정보 전파 메커니즘에 대한 연구가 요구된다.

## 참고문헌

- [1] Scott A. Golder and Bernardo A. Huberman (2005). "The Structure of Collaborative Tagging Systems", <http://arxiv.org/abs/cs/0508082>.
- [2] Miller, B., J. Konstan, L. Terveen, and J. Riedl, "PocketLens: Towards a Personal Recommender System", *ACM Transactions on Information Systems*, Vol.22(2004), 437~476.
- [3] Guha, R., R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of Trust and Distrust", *Proceedings of the 13th International Conference on World Wide Web*, (2004), 403~412.
- [4] Ziegler, C-N. and G. Lausen, "Propagation Models for Trust and Distrust in Social Networks", *Information Systems Frontiers*, Vol.7(2005), 337~358.
- [5] Evien, R. and A. Aiken, "Attack Resistant, Scalable Name Service", Draft submission to *the 4th International Conference on Financial Cryptography*, (2000).
- [6] Sarwar, B., G. Karypis, J. Konstan, and J. Reidl, "Item-based Collaborative Filtering Recommendation Algorithms", *Proceedings of the 10th International Conference on World Wide Web*, (2001), 285~295.
- [7] Sinha, R. and K. Swearingen, "Comparing Recommendations Made by Online Systems and Friends", *Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries* (2001).
- [8] Ziegler, C-N. and G. Lausen, "Analyzing Correlation between Trust and User Similarity in Online Communities", *Proceedings of the 2nd International Conference on Trust Management, Lecture Notes in Computer Science*, Vol.2995(2004), 251~265.
- [9] Massa, P. and P. Avesani, "Trust-aware Collaborative Filtering for Recommender Systems", *Proceedings of International Conference on Cooperative Information Systems, Lecture Notes in Computer Science*, Vol. 3290(2004), 492~508.
- [10] Lange, D. and M. Oshima, "Seven Good Reasons for Mobile Agents", *Communications of the ACM*, Vol.42, No.3(1999), 88~89.
- [11] Foner, L. N, "YENTA: A Multi-agent Re-

- ferral System for Matchmaking”, *Proceedings of 1st International Conference on Autonomous Agents*, (1997), 301~307.
- [12] Good, N., J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, “Combining Collaborative Filtering with Personal Agents for Better Recommendations”, *Proceedings of the 16th National Conference on Artificial Intelligence*, (1999), 439~446.
- [13] Herlocker, J. L., J. Konstan, A. Borchers, and J. Riedl, “An Algorithmic Framework for Performing Collaborative Filtering”, *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, (1999), 230~237.
- [14] Sawar, B., J. Konstan, A. Bochers, J. Herlocker, B. Miller, and J. Riedl, “Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System”, *Proceedings of the Conference on Computer Supported Cooperative Work*, (1998), 345~354.
- [15] Jung, J. J., “Visualizing Recommendation Flow on Social Networks”, *Journal of Universal Computer Science*, Vol.11, No. 11(2005), 1780~1791.
- [16] Kim, H. J., J. J. Jung, and G. S. Jo, “Conceptual Framework for Recommendation System based on Distributed User Ratings”, *Lecture Notes in Computer Science*, Vol.3032(2003), 115~122.
- [17] Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An Open Architecture for Collaborative Filtering of Netnews”, *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, (1994), 175~186.
- [18] Resnick, P. and H. R. Varian, “Recommender Systems”, *Communications of the ACM*, Vol.40, No.3(1997), 56~58.
- [19] Konstan, J. A., B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, “GroupLens: Applying Collaborative Filtering to Usenet News”, *Communications of the ACM*, Vol.40 No.3(1997), 77~87.
- [20] Breese, J., D. Heckerman, and C. Kadie, “Empirical Analysis of Predictive Algorithms for Collaborative Filtering”, *Microsoft Research*, 1998.
- [21] O'Donovan, J. and B. Smyth, “Trust in Recommender Systems”, *Proceedings of the 10th International Conference on Intelligent User Interfaces*, (2005), 167~174.
- [22] Shardanand, U. and P. Maes, “Social Information Filtering: Algorithms for Automating “Word of Mouth.””, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (1995), 210~217.
- [23] Massa, P. and B. Bhattacharjee, “Using Trust in Recommender Systems: An Experimental Analysis”, *Proceedings of 2nd International Conference on Trust Management, Lecture Notes in Computer Science*, Vol.2995(2004), 221~235.
- [24] Deshpande, M. and G. Karypis, “Item-based Top-N Recommendation Algorithms”, *ACM Transactions on Information Systems*, Vol.22 No.1(2004), 143~177.
- [25] Devore, J. L., Probability, and Statistics for Engineering and the Sciences, 4th Edition, Duxbury Press, 1995.
- [26] Mobasher, B., H. Dai, and M. Toa, “Discovery and Evaluation of Aggregation Usage Profiles for Web Personalization”, *Data Mining and Knowledge Discovery*, 6(2002), 61~82.
- [27] Unger, L. and D. Foster, “Clustering Methods for Collaborative Filtering”, *Proceedings of the Workshop on Recommendation Systems* (1998), 112~125.

Abstract

## A Multi-Agent Framework for Distributed Collaborative Filtering

Ae-Ttie, Ji\* · Cheol, Yeon\* · Seung-Hun, Lee\* · Heung-Nam, Kim\* · Geun-Sik Jo\*

Recommender systems enable a user to decide which information is interesting and valuable in our world of information overload. As the recent studies of distributed computing environment have been progressing actively, recommender systems, most of which were centralized, have changed toward a peer-to-peer approach. Collaborative Filtering (CF), one of the most successful technologies in recommender systems, presents several limitations, namely sparsity, scalability, cold start, and the shilling problem, in spite of its popularity. The move from centralized systems to distributed approaches can partially improve the issues; distrust of recommendation and abuses of personal information. However, distributed systems can be vulnerable to attackers, who may inject biased profiles to force systems to adapt their objectives. In this paper, we consider both effective CF in P2P environment in order to improve overall performance of system and efficient solution of the problems related to abuses of personal data and attacks of malicious users. To deal with these issues, we propose a multi-agent framework for a distributed CF focusing on the trust relationships between individuals, i.e. web of trust. We employ an agent-based approach to improve the efficiency of distributed computing and propagate trust information among users with effect. The experimental evaluation shows that the proposed method brings significant improvement in terms of the distributed computing of similarity model building and the robustness of system against malicious attacks. Finally, we are planning to study trust propagation mechanisms by taking trust decay problem into consideration.

**Key Words** : Multi-Agent System, Distributed Recommender System, Collaborative Filtering, Shilling Attack, Web of Trust, Social Network

---

\* Department of Computer & Information Engineering, Inha University