평가 스트림 추세 분석을 이용한 추천 시스템의 공격 탐지☆

Attack Detection in Recommender Systems Using a Rating Stream Trend Analysis

김 용 욱* 김 준 태**
Yonguk Kim Juntae Kim

요 약

추천 시스템은 사용자의 선호도를 분석하고, 아이템들에 대한 사용자의 선호도를 예측하여 책, 영화, 음악 등과 같은 아이템을 사용자에게 추천하는 시스템이다. 추천 시스템에서 가장 널리 활용하는 기법은 협동적 여과 기법이며, 협동적 여과 기법은 추천 대상 사용자에게 아이템을 추천할 때 유사 사용자의 평가 정보를 이용한다. 협동적 여과 기반 추천은 시스템 공격자가 악의적 목적을 가지고 아이템에 대한 평가를 조작하였을 경우 추천 성능이 저하되며, 이와 같은 추천 시스템에 대한 악의적 행위를 추천 공격이라 한다. 지속적으로 변화하는 평가 데이터를 데이터 스트림 관점에서 분석하면 추천 시스템의 공격을 예측할 수 있다. 본 논문에서는 협동적 여과 기반 추천 시스템에서 아이템 평가의 스트림 추세를 이용하여 추천 시스템에 대한 공격을 탐지하는 방법을 제안한다. 평가 데이터를 구성하는 아이템 평가 정보는 시간에 따라 수시로 변화되는 특성을 나타내기 때문에 일정 주기에 따라 아이템의 평가 변화를 측정하면 추천 시스템의 공격을 탐지할 수 있다. 본 논문에서 제안하는 기법은 연속적으로 입력되는 평가 스트림을 공격 탐지 검사 주기를 기반으로 정상적인 스트림 추세와 비교하여 비정상적인 스트림 추세를 탐지한다. 본 논문에 제안한 기법을 추천 공격에 적용하면 추천 시스템의 운용성과 평가 데이터의 재사용성을 향상시킬 수 있다. 본 논문에서 제안한 기법을 다양한 실험을 통해 효과를 확인하였다.

ABSTRACT

The recommender system analyzes users' preference and predicts the users' preference to items in order to recommend various items such as book, movie and music for the users. The collaborative filtering method is used most widely in the recommender system. The method uses rating information of similar users when recommending items for the target users. Performance of the collaborative filtering-based recommendation is lowered when attacker maliciously manipulates the rating information on items. This kind of malicious act on a recommender system is called 'Recommendation Attack'. When the evaluation data that are in continuous change are analyzed in the perspective of data stream, it is possible to predict attack on the recommender system. In this paper, we will suggest the method to detect attack on the recommender system by using the stream trend of the item evaluation in the collaborative filtering-based recommender system. Since the information on item evaluation included in the evaluation data tends to change frequently according to passage of time, the measurement of changes in item evaluation in a fixed period of time can enable detection of attack on the recommender system. The method suggested in this paper is to compare the evaluation stream that is entered continuously with the normal stream trend in the test cycle for attack detection with a view to detecting the abnormal stream trend. The proposed method can enhance operability of the recommender system and re-usability of the evaluation data. The effectiveness of the method was verified in various experiments.

☞ keyword : Recommendation System(추천 시스템), Recommendation Attack(추천 공격), Data Stream(데이터 스트림)

* 정 회 원 : 동국대학교 컴퓨터공학과 박사수료
yukim@dongguk.edu
** 정 회 원 : 동국대학교 컴퓨터공학과 교수
jkim@dongguk.edu
[2011/02/10 투고 - 2011/02/18 심사 - 2011/03/16 심사완료]

☆ 이 논문은 2007년도 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2007-313-D00706)

1. 서 론

추천 시스템은 사용자가 원하는 제품을 찾을 수 있도록 도와주므로 사용자에게 유용한 시스템이며, 추천 시스템에 저장되는 선호도 정보는 제품을 제공하는 기업에게도 귀중한 자산이다[1,2]. 추천 시스템에서 제품의 평가 정보를 활용할 때는 사용자의 여러 가지 행위를 이용한다. 사용자들의 제품 평가에 대한 행위로 대표적인 것은 제품에 대한 평점이나 리뷰이다. 이러한 평가 행위는 고객들의 제품 선택에 중요한 요소로 작용되고, 제조업자에게도 이러한 평가 정보들은 제품 판매량과 직결되므로 매우 중요한 정보로 활용되며, 추천 시스템에서는 사용자 성향 분석에 활용 가능하다.

사용자에게 필요한 정보를 제공하는 추천 시스템은 크게 세 가지로 나뉜다. 개인화 서비스를 구현하기 위해 사용하는 정보나 알고리즘에 따라 추천 시스템은 내용기반 추천 시스템, 인구통계학적 추천 시스템, 협동적 여과 추천 시스템 등으로 나뉜다[3]. 내용기반 추천 시스템은 각 아이템에 관한 서술 등의 아이템 내용 정보를 분석하고 이용하는 것으로써, 뉴스 기사나 웹 페이지와 같이 내용 정보가 풍부한 경우에 적합하다[4]. 그러나 영화나 음악과 같은 멀티미디어 정보처럼 내용 분석이 어려운 경우 유사도 측정 및 아이템 추천이 어렵다는 단점이 있다. 인구통계학적 추천 시스템은 사용자들의 나이, 성별, 직업과 같은 개인 정보를 이용한다[4]. 사용자들의 인구 통계학적 정보를 활용하여 사용자들 사이의 거리를 측정하고, 이웃한 사용자들을 찾아 특정 아이템에 대한 이웃 사용자들의 평균 선호도를 계산하여 사용자의 선호도를 예측한다. 아이템 범주에 대한 개인적인 성향은 인구통계학적 속성에 의해 쉽게 구별될 수 있기 때문에 인구통계학적 정보를 기반으로 하는 추천 시스템은 백화점과 같은 다양한 종류의 아이템이나, 인구통계학적 집단 성향이 잘 나타나는 영역에 적합하다. 협동적 여과 추천 시스템은 아이템들에 대한 각 사용자들의 평가 정보를 이용하여 유사 사용자의 선호도를 기반으로 특정 아이템에 대한 사용자의 선호도를 예측한다[5,6]. 협동적 여과 추천 시스템은 아이템의 내용 정보를 필요로 하지 않기 때문에 내용을 분석하기 어려운 음악이나 영화 같은 아이템 등을 추천하는 데 효과적이다. 다양한 추천 방법들 중 특히 협동적 여과 추천 방법은 다수 사용자의 평가를 기반으로한 추천 개념을 기반으로 Minnesota 대학의 GroupLens 프로젝트로부터 시작되어[7], Amazon을 비롯한 다양한 웹 사이트에 성공적으로 적용되어 왔다[8]. 최근의 추천 시스템에 대한 연구는 다양한 협동적 추천 방식에 대한 연구가 주류를 이루고 있다[9-14].

협동적 여과를 이용한 추천 시스템은 시스템 공격자가 악의적 목적을 가지고 아이템에 대한 평가를 조작하였을 때 추천 성능이 저하되며, 이와 같은 추천 시스템에 대한 비정상적인 악의적 행위를 '추천 공격'이라 한다. 추천 시스템에 대한 공격은 아이템에 대한 평가 정보를 악의적으로 왜곡하여 추천 시스템의 성능에 악영향을 끼친다[15,16]. 이러한 추천 공격은 특정한 아이템의 추천 비율을 높이거나 낮추기도 하며, 불특정 다수의 아이템을 대상으로 공격을 수행하기 때문에 전체 추천 시스템의 성능을 저하시키기도 한다. 그러므로 추천 시스템에 대한 공격을 탐지하여 미리 차단함으로써 추천 결과의 왜곡을 방지하는 견고한 추천 시스템을 개발해야 한다[17,18]. 본 논문은 추천 시스템의 악의적 공격에 대해 견고한 추천 시스템 설계를 목적으로 하며, 본 논문에서 제안하는 방법은 아이템의 평가 스트림 추세를 분석하여 추천 공격을 탐지한다.

아이템이 개발되어 시장에 출시되었을 때 아이템은 일시적으로 시장의 주목을 받으나, 일정 시간이 지나면 아이템에 대한 사용자들의 관심도는 낮아진다. 이러한 사용자의 관심 변화는 추천 시스템에서도 중요한 정보로 활용될 수 있다. 아이템의 인기도가 높을 경우에는 많은 사용자들로부

터 선호되기 때문에 추천 비율이 높아지며, 인기도가 낮은 아이템은 사용자의 관심으로부터 멀어진 것이기 때문에 추천 비율이 낮아진다. 아이템 평점과 같은 사용자의 행위는 추천 시스템에서 쉽게 취득할 수 있는 정보이며, 이러한 사용자의 행위는 지속적인 스트림 데이터(stream data) 형식으로 추천 시스템에 입력된다. 이러한 스트림 데이터를 분석하면 추천 시스템 공격을 탐지할 수 있다.

본 논문에서는 아이템 평가에 대한 스트림 데이터를 활용하여 추천 시스템 공격을 탐지하는 기법을 제안한다. 본 논문에서 제안하는 스트림 데이터 분석을 이용한 공격 탐지 기법을 평가 스트림 추세 분석(RSTA, Rating Stream Trend Analysis)이라 한다. 평가 스트림 추세 분석은 일정한 주기 간격으로 아이템의 선호도 평가 등을 분석하여 추천 시스템 공격을 탐지한다. 본 논문에서 제안하는 기법은 추천 시스템의 공격을 탐지할 수 있고, 악의적 공격자가 오염시킨 아이템을 추출할 수 있는 장점이 있으며, 공격 아이템을 제거한 후 평가 데이터를 추천 시스템에 재사용할 수 있는 장점이 있다.

본 논문의 2장에서는 관련 연구에 대해 설명하고, 3장에서는 본 논문에서 제안하는 평가 스트림 추세 분석에 대해 설명한다. 4장에서는 제안하는 기법의 실험 결과를 분석하고, 5장에서는 결론과 향후 연구에 대해 기술한다.

2. 관련연구

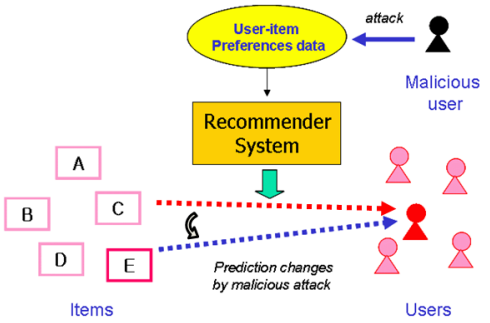
협동적 여과 방법은 유사 사용자들의 선호 아이템 정보를 바탕으로 추천 대상 사용자에게 취향에 맞는 아이템을 추천하는 기법이며, 유사 사용자를 찾을 경우에는 추천 대상 사용자의 아이템들에 대한 선호도를 다른 사용자들의 아이템들에 대한 선호도와 비교한 후, 유사도값에 따라 유사한 사용자들을 추출한다. 유사 사용자들의 아이템 선호도를 기반으로 추천 대상 사용자에게 아이템을 추천하며, 사용자 사이의 유사도 측정에는 일반적으로 코사인 유사도나 피어슨 상관계수를 이용하고, 피어슨 상관계수는 식 1과 같다. 추천 대상 사용자 u의 아이템 i에 대한 선호도 예측값은 유사 사용자의 선호도 가중 평균으로 측정하며, 이와 같은 방식을 취하면 유사도가 높은 사용자가 선호한 아이템의 가중치를 높이는 효과가 있다. 피어슨 상관계수를 이용한 아이템 선호도 예측 공식은 식 2와 같다.

$$W_{u,w} = \frac{\sum_i (r_{u,i} - \overline{r_u})(r_{w,i} - \overline{r_w})}{\sqrt{\sum_i (r_{u,i} - \overline{r_u})^2 (r_{w,i} - \overline{r_w})^2}} \quad (1)$$

$$P_{u,i} = \overline{r_u} + \frac{\sum_w (r_{w,i} - \overline{r_w}) \cdot W_{u,w}}{\sum_w |W_{u,w}|} \quad (2)$$

식 1, 2에서 $W_{u,w}$는 사용자 u와 사용자 w의 유사도이고, $P_{u,i}$는 사용자 u의 아이템 i에 대한 선호도 예측값이다. $r_{u,i}$는 사용자 u의 아이템 i에 대한 선호도를 나타내고, $\overline{r_u}$는 사용자 u의 선호도 평균을 나타낸다.

추천 시스템에 대한 '공격'은 위조된 사용자 평가를 추천 시스템에 삽입함으로써 추천 시스템이 올바르게 추천을 수행하지 못하도록 추천 시스템에 악영향을 미치는 작업을 말한다[19]. 예를 들어 제조업자들은 시장에서 자신들의 제품이 다른 제품들보다 많이 판매되기를 원하므로, 추천 시스템에서도 자신의 경쟁 제품이 자주 추천되는 것을 바라지 않는다. 그러므로 사용자들에게 우수한 평가를 받은 제품이 추천되지 않도록 우수한 제품의 평가를 위·변조하는 악의적 행위를 수행할 수 있다. 추천 시스템에 악의적인 영향을 주는 방법 중 하나는 소프트웨어 에이전트를 추천 시스템에 접속시켜 무작위로 아이템들을 선정하여 평가값을 조작하는 것으로, 이와 같은 악의적 행위는 추천 시스템의 추천 성능을 급격히 저하시킨다. 이러한 악의적 행위를 추천 시스템에 대한 공격이라 하며, (그림 1)에 이러한 추천 시스템에 대



(그림 1) 협동적 여과 추천 시스템에 대한 공격

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
U1	1	2	2	0	0	0	0	0	0	0
U2	0	0	0	1	2	3	0	0	0	0
U3	0	0	0	0	0	0	3	2	3	0
U4	1	0	3	1	0	0	0	0	0	0
U5	0	0	0	2	3	0	1	0	0	0
U6	0	0	0	0	3	3	3	2	0	0
U7	3	1	1	1	0	0	0	0	0	0
U8	0	0	1	3	3	0	0	0	0	0
U9	3	3	3	0	0	0	0	0	3	3
U10	3	3	3	0	0	3	0	3	0	0

U1과유사도	
U1	-
U2	0.00
U3	0.00
U4	0.70
U5	0.00
U6	0.00
U7	0.67
U8	0.15
U9	0.75
U10	0.75

(그림 2) 추천 시스템 공격 예제

한 공격 개념을 나타낸다. '공격자'는 공격 행위를 수행하는 에이전트 또는 사람을 의미한다. 일반적으로 공격은 조작된 사용자 평가를 추천 시스템에 삽입함으로써 이루어진다. 추천 시스템에 삽입하는 조작된 사용자 평가를 공격 프로파일(attack profile)이라고 한다. 공격자는 추천 시스템의 성능을 저하시키는 공격 프로파일을 작성하여 공격을 수행한다. 특정 아이템에 관계없이 추천 시스템의 추천 성능을 저하시키는 공격을 임의 공격(random attack)이라 한다[19,20].

임의 공격은 특정 사용자나 제품에 포커스를 두지 않고 임의의 사용자와 아이템을 대상으로 한다. 임의 공격과 같이 추천 시스템의 추천 성능을 저하시키는 공격은 특정한 아이템을 목표로 하지 않고, 시스템 내의 임의의 아이템들을 목표로 공격이 수행된다. 임의 공격은 무작위적인 공격 방법이므로 사용자나 아이템의 성향이나 특징에 관계없이 공격이 수행됨으로, 추천 시스템에 미치는 공격 영향력이 특정 아이템을 목표로 한 공격보다 크다. 임의 공격은 광범위하게 아이템과 사용자들에게 영향을 주기 때문에 위험성이 높은 추천 시스템 공격 방법이다.

(그림 2)는 임의 공격을 보인 예제이며, 표에 나타난 수치는 사용자의 아이템 평가값을 의미한다. U1부터 U8은 정상적인 사용자를 나타내며, I1부터 I10은 아이템을 나타낸다. U9와 U10은 임의 공격을 수행하는 공격자로, U9와 U10은 사용자의 성향에 관계없이 무작위로 아이템 평가를 악의적으로 조작한 공격자이며, 원시 평가 데이터에서는 U9와 U10이 사용자로 존재하지 않는다. 원시 데이터에서 U1 사용자와 유사한 사용자를 추출하면 유사도 측정에 의해 U4가 U1과 가장 유사한 사용자로 추출되고, U7은 두 번째로 유사성이 높은 사용자로 추출된다. U1과 U4의 유사도를 측정하면 유사도는 0.70이 되고, U1과 U7의 유사도는 0.67이 된다. U1은 아이템 I1, I2, I3을 평가하였고 U4는 아이템 I1, I3, I4를 평가하였기 때문에, 추천 시스템에서는 U1 사용자에게 U4 사용자가 평가한 I4를 추천하게 될 것이다. 원시 평가 데이터에서 U9와 U10 공격자의 공격이 발생한 경우, U1과 유사한 사용자를 추출하면 U1과 유사도가 가장 높은 사용자로 U9와 U10이 추출된다. U9와 U10은 U1과 0.75의 유사도를 나타내어 U4보다 유사도 값이 높다. 그러므로 추천 공격이 수행된 후에 추천 시스템은 U1 사용자에게 U9와 U10이 선호한 아이템 I6, I8, I9, I10을 추천하게 된다. 추천 시스템에 공격이 수행되면 사용자의 정상적인 성향이 무시된 상태에서 아이템 추천을 수행하기 때문에 추천 시스템의 성능은 급격히 저하된다.

Burke 등[20]은 협동적 추천 시스템에 대한 공격의 유형에 대하여 연구하고, 공격자의 공격을 탐지 및 분류하기 위한 연구를 수행하였다. 이 연구에서는 공격자의 프로파일 유형에 따라 공격의 유형을 분류하였다. 그리고 공격을 탐지하기 위해서 프로파일에 포함된 평가 값의 평균 편차, 프로파일에 포함된 평균 평가 횟수, 아이템에 대한 평

균 평가 값 등을 분류 속성으로 이용하였다. 이 연구에서 활용한 탐지 방법은 한 사용자의 프로파일 자체에서 속성을 추출하여 유사 사용자를 찾은 후에 공격자를 탐지한다. 그러나 이 방법은 임의 공격과 같이 무작위적인 규칙 없는 공격일 경우, 유사 사용자를 찾기 어려워 낮은 정확도를 보이게 된다.

Dellarocas[21]는 eBay 같은 온라인 상업 커뮤니티에서 사용되는 유명한 시스템에 대한 몇 가지 공격들에 대하여 정리하였다. 그리고 현재 존재하는 협동적 필터링 알고리즘과 유사하며, 공격의 효과를 최소화하는 예측 알고리즘을 제안하였다.

Lam과 Riedl[1]은 조작된 사용자의 평가에 대하여 분석하고, 추천 공격을 유형별로 분류하였다. 추천 시스템에 영향을 주는 방법은 사용자들이 시스템에 접속하여 아이템들을 평가하는 방법을 취한다. 그러므로 공격자들의 조작된 평가는 다른 사용자들이 잘못된 판단을 내리도록 유도한다. Lam과 Riedl은 조작된 평가와 같이 추천에 영향을 주는 요소들을 분석하였다.

추천 시스템 공격에 대해 다양한 연구가 진행되고는 있으나, 아직까지 추천 시스템의 임의 공격을 적절하게 예측할 수 있는 연구는 미비한 상태이다. 본 논문에서 제안하는 기법은 평가 스트림 추세를 분석하여 추천 시스템의 임의 공격을 예측한다.

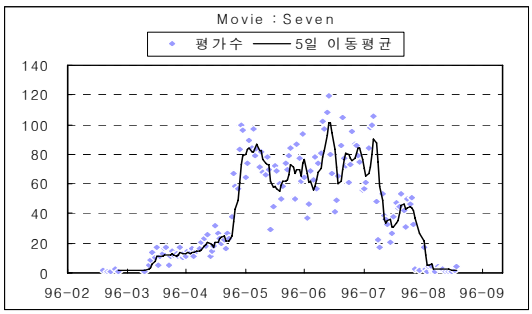
3. 평가 스트림 추세 분석을 이용한 공격 탐지

추천 시스템에 대한 임의 공격 목적은 추천 시스템의 추천 성능을 저하시키는 것이다. 그러므로 임의 공격은 특정 아이템을 대상으로 하는 것이 아니라, 추천 시스템에 포함된 임의의 아이템을 대상으로 공격을 수행한다. 추천 시스템에 대한 공격은 아이템을 대상으로 수행되므로 아이템에 대한 공격을 탐지하면 추천 시스템에 대한 공격을 효과적으로 탐지할 수 있다. 아이템에 대한 공격은 아이템 평가를 변화시키는 방법을 취하므로 아이템의 평가 변화를 분석하면 추천 시스템 공격을 탐지할 수 있다.

아이템이 개발되어 시장에 출시되었을 때, 아이템은 일시적으로 시장의 주목(예 : 평가 등)을 받아 인기가 상승 곡선을 그린다. 그러나 일정 시간이 지나면 다른 아이템의 출현이나 시간의 경과에 따라 아이템에 대한 사용자들의 관심도는 낮아진다. 이와 같이 아이템의 관심 변화는 스트림 형태로 지속적으로 변화한다. 이와 같은 아이템 관심에 대한 스트림 데이터 변화를 분석하면 추천 시스템 공격을 쉽게 탐지할 수 있다. 아이템 관심에 대한 정보는 스트림 형태로 시스템에 입력되기 때문에, 이와 같은 스트림 정보 분석을 통해 추천 시스템 공격을 탐지하는 방법을 본 논문에서는 평가 스트림 추세 분석이라 정의한다. 추천 시스템 공격자는 시스템의 성능을 신속하게 저하시키려는 목적을 갖기 때문에 아이템의 평가 정보를 빠르게 변화시킨다. 이 경우에 본 논문에서 제안하는 평가 스트림 추세 함수는 급격히 변화하게 되며, 평가 스트림 추세 함수에 대한 변화율을 분석하면 아이템에 대한 공격을 예측할 수 있다.

스트림 데이터에 대한 추세 변화를 EachMovie 데이터의 예에서 보면 다음과 같다. EachMovie 데이터는 18개월 동안 72.916명의 사용자가 영화에 대해 선호도 평가를 수행한 데이터 집합이다. 평가에 사용된 영화의 개수는 1,628개이고, 사용자 평가는 2,811,983건이다. 사용자의 평가는 0.0부터 1.0까지 6단계의 평가값을 활용한다. EachMovie 데이터에는 사용자 정보로 ID, 나이, 성별 등이 나타나고, 영화 정보로 영화 이름, 극장 상영일, 영화 장르 등이 나타나며, 영화 장르는 10개가 존재한다.

(그림 3)은 EachMovie 데이터에 나타난 영화 'Seven'의 평가 변화이다. X축은 날짜를 나타내며, Y축은 1일 주기로 측정한 사용자들의 평가 횟수이다. 일반적으로 아이템들에 대한 평가 수의 변화는 (그림 3)과 같은 형태의 분포를 보인다. (그



(그림 3) 영화 "Seven"의 평가 횟수

림 3)에 나타난 추세 곡선은 5일 이동 평균을 이용하여 평가 변화를 표현한 것이다. 그림에 나타난 바와 같이 시간 흐름에 따라 아이템에 대한 사용자의 평가는 점차 증가하고 있으며, 영화에 대한 평가는 증가된 상태에서 몇 달간 지속되다가 점차 감소하여 더 이상 평가되지 않는 것으로 나타난다. 아이템 평가와 같은 인기도는 이와 같이 일정한 유형의 추세 곡선을 나타내는 스트림 형태로 시스템에 입력된다.

본 논문에서는 견고한 추천 시스템을 개발하기 위해 평가 스트림 추세 함수(Rating Stream Trend Function, RSTF)를 정의하여 추천 시스템에 대한 악의적 공격을 탐지하는 기법을 제안한다. 지속적인 평가 데이터의 변화를 데이터 스트림 관점에서 검사하면 추천 시스템의 공격을 예측할 수 있으며, 아이템들의 평가 정보는 시간에 따라 수시로 변화되는 특성을 나타내기 때문에 일정한 시간에 따라 아이템의 평가 변화를 측정하면 추천 시스템의 공격을 탐지할 수 있다. 그러므로 각 아이템들에 대한 공격 탐지를 시간 변화에 따라 주기적으로 검사하면 각 아이템들의 평가 변화에 따라 아이템 공격을 예측할 수 있다.

본 논문에서 제안하는 추천 시스템에 대한 공격 탐지 기법인 평가 스트림 추세 함수를 식 4에 나타내었다. $RSTF_{\triangle t_i}(I_j)$ 는 아이템 I_j 의 Δt_i 시간 동안의 평가 변화율을 의미하며, 식 3으로 정의한다. 식에서 t는 시간을 의미하며, $\triangle t_i$는 초기 시간 t_0에서 t_i시간까지의 시간 변화를 나타낸다. t_i는 공격 탐지가 이루어지는 i 번째 시점을 의미하며, t_i의 시간 간격은 임의의 등구간으로 설정하여 공격 탐지에 이용한다. I_j는 j번째 아이템을 나타내고, $V_{\triangle t_i}(I_j)$는 $\triangle t_i$ 시간 구간에 발생한 아이템 I_j에 대한 평가 수이다.

$T_{\triangle t_i}$는 $\triangle t_i$에 대한 공격 탐지 시간으로 고정된 $\triangle t_i$의 크기에 따라 $T_{\triangle t_i}$는 유동적인 값으로 아이템의 특성에 따라 변화될 수 있는 값이다. 이와 같이 $\triangle t_i$에 따라 $T_{\triangle t_i}$를 변화시킬 수 있게 한 이유는 아이템에 따라 평가 패턴이 다양하게 나타날 수 있기 때문이다. $T_{\triangle t_i}$는 $\triangle t_i$ 에 따른 조정 시간으로 $\triangle t_i$는 같다고 하더라도 $T_{\triangle t_i}$는 서로 다른 값을 취할 수 있다. 즉, $T_{\triangle t_i}$는 $\triangle t_i$에 κ값을 곱한 것과 같은 형식을 취하여 $\kappa\triangle t_i$ 형태로 재 표현 할 수 있다. $T_{\triangle t_i}$가 이와 같은 형식을 취하는 이유는 탐지 시간에 대한 평가 행태가 비슷한 경우의 아이템일 경우의 아이템을 경우에는 κ값을 1로 결정하여 아이템들의 공격 스트림을 분석해도 무방하지만, 일반적인 평가 패턴이 다른 아이템의 경우에는 다른 아이템들과 공격 스트림을 분석할 때 해당 아이템의 특성을 고려하여야 함으로 κ값을 이용하여 탐지 시간 조정이 필요하게 된다. 예를 들면 사용 기간이 짧은 아이템(예 : 볼펜, 식료품 등)일 경우는 짧은 시간에 많은 평가가 발생되겠지만, 사용 기간이 긴 아이템(예 : 자동차, 가구 등)의 경우에는 짧은 시간에 많은 평가가 발생되지 않기 때문에 사용 기간이 짧은 아이템에 비해 평가 시간을 길게 하는 것이 공격 탐지나 시스템 운영에 효율적이다. 아이템 특성이나 공격 탐지 시간에 따라 공격량의 변화는 매우 다양하게 변화될 수 있으므로, 공격을 정확히 탐지하기 위해서는 탐지 구간에 따라 탐지 시간을 적절하게 변화를 주는 것이 공격 탐지에 효율적일 수 있다. 예를 들면 공격 탐지 구간에 비해 공격량이 매우 적은 공격일 경우에는 정상적인 행위로 판단될 수 있는 오류가 있다. 이와 같은 공격 특성을 나타낼 경우에는 탐지 구간을 변화시켜

공격 탐지에 활용하는 것이 효과적이다. 이와 같은 아이템별 공격 특성에 따라 평가 스트림 추세 함수는 $T_{\triangle t_i}$를 아이템 평가 특성에 따라 조절하여 사용할 수 있다.

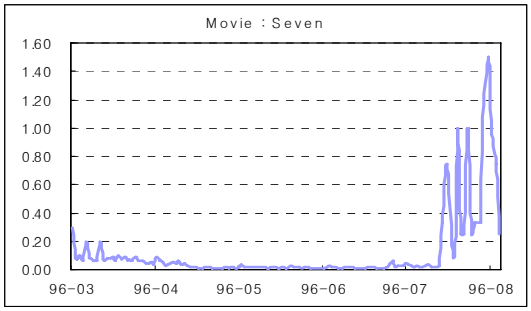
α_i는 아이템 I_j의 스트림 패턴 횟수이고, β_i는 아이템 I_j를 평가했던 사용자 총수이다. 그러므로 α_i는 I_j의 스트림 패턴의 변화량을 나타내고, β_i는 I_j의 인기도를 나타낸다. 초기의 α_i는 1값을 갖지만, 아이템의 특성상 스트림 탐지 구간의 변경이 필요하여 탐지 구간이 변경되면 변경 횟수가 α_i에 누적된다. 그러므로 α_i값을 확인하면 아이템의 스트림 패턴 변화 특성을 확인할 수 있다. 평가 스트림 추세 함수에 α_i와 β_i를 적용함으로 아이템의 인기도와 스트림 패턴 변화를 나타낼 수 있는 장점이 있다.

구간별 평가 스트림 추세 함수는 각 $\triangle t_i$의 아이템에 대한 평가 변화율을 의미한다. 전체 평가 스트림 추세 함수는 구간별 평가 스트림 추세 함수를 결합한 형태로 표현되며, 식 4는 아이템에 대한 전체 평가 스트림 추세 함수로 아이템의 구간별 평가 스트림 추세 함수에 대한 평균을 의미한다.

$$RSTF_{\Delta t_i}(I_j) = \frac{T_{\triangle t_i}(I_j)+\alpha_i}{V_{\triangle t_i}(I_j)+\beta_i} \quad (\triangle t_i = |t_i - t_0|) \tag{3}$$

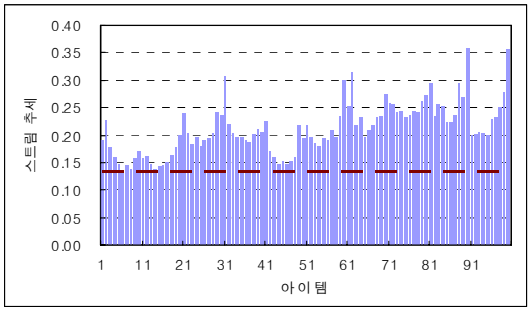
$$\begin{aligned} RSTF_t(I_j) &= (RSTF_{\triangle t_1}(I_j) + RSTF_{\triangle t_2}(I_j) + \cdots + \\ &\quad RSTF_{\triangle t_i}(I_j) + \cdots + RSTF_{\triangle t_n}(I_j)) \\ &\quad \cdot \frac{1}{n} \quad (\triangle t_i = |t_i - t_0|) \\ &= \left(\frac{T_{\triangle t_1}(I_j)+\alpha_1}{V_{\triangle t_1}(I_j)+\beta_1} + \frac{T_{\triangle t_2}(I_j)+\alpha_2}{V_{\triangle t_2}(I_j)+\beta_2} + \cdots \right. \\ &\quad \left. + \frac{T_{\triangle t_i}(I_j)+\alpha_i}{V_{\triangle t_i}(I_j)+\beta_i} + \cdots + \frac{T_{\triangle t_n}(I_j)+\alpha_n}{V_{\triangle t_n}(I_j)+\beta_n} \right) \\ &\quad \cdot \frac{1}{n} \quad (\triangle t_i = |t_i - t_0|) \\ &= \frac{1}{n} \cdot \sum_{k=1}^{n} \left(\frac{T_{\triangle t_k}(I_j)+\alpha_k}{V_{\triangle t_k}(I_j)+\beta_k} \right) \quad (\triangle t_k = |t_k - t_0|) \end{aligned} \tag{4}$$

(그림 4)는 영화 'Seven'의 평가 데이터를 본 논문에서 제안하는 평가 스트림 추세 함수로 표현한 것이다. 평가 스트림과 평가 스트림 추세 함수는 역함수 관계로 표현되기 때문에 아이템이 활성화된 상태에서는 빠른 진폭 변화로 함수값이 낮게 표현된다. 이와 같은 결과로 아이템이 활성화된 상태에서는 비활성화 상태보다 진폭 변화 횟수가 급격히 많아지고 낮은 함수값을 갖는다. 그림에 나타난 바와 같이 평가 스트림 추세 함수를 사용하여 아이템의 활성화 상태를 쉽게 표현할 수 있다. 일반적으로 아이템이 처음 출시된 경우에는 사용자들에게 인기를 얻지 못하여 평가 스트림의 변화가 적지만, 시간이 흐름에 따라 사용자의 관심이 증대되어 평가 스트림은 빠르게 변화하는 양상을 나타낸다.



(그림 4) 영화 'Seven'의 평가 스트림 추세

본 논문에서 제안하는 기법은 이러한 아이템의 평가 스트림을 이용하여 아이템의 활성화 상태를 측정한다. 만약 추천 공격이 발생하여 아이템이 공격을 받게 되면 평가 스트림은 빠르게 변화될 것이고, 공격받지 않은 정상적인 아이템의 스트림 곡선과는 다른 형태의 스트림 곡선을 생성한다. 이와 같은 결과가 발생하는 이유는 아이템 공격의 경우 효과적인 공격을 위해서는 짧은 시간 내에 공격이 이루어져야 하기 때문이며, 스트림 곡선의 추세는 정상적인 아이템의 스트림 곡선과는 많은 차이가 나타난다. 즉, 공격이 수행되면 아이템의 평가 스트림 변화는 매우 빠르게 증가한다. 평가 스트림 추세 함수는 아이템의 활성화 상태를 측정할 때 적용되며, 활성화가 비정상적으로



(그림 5) 아이템별 평가 스트림 추세

높게 나타날 경우에는 스트림 변화량은 매우 크게 증가되고, 정상적인 활성화 상태가 되면 스트림 변화량은 낮아지게 된다.

추천 공격은 짧은 시간에 공격이 집중된다는 특징이 있기 때문에 공격이 발생할 경우에 평가 스트림 추세 함수의 변화율은 급격하게 증가한다. 이러한 정보 형태는 추천 공격을 탐지하는 데 유용하게 사용할 수 있다. 본 논문에서 제안하는 기법은 아이템의 평가 스트림 추세 함수를 기반으로 공격 아이템을 탐지하며, 평가 스트림 추세 변화를 주기적으로 측정하면 아이템에 대한 공격을 쉽게 탐지할 수 있다.

평가 스트림 추세 함수의 적용은 다음과 같다. 평가 스트림 추세를 측정할 주기가 되면 각 아이템별로 평가 스트림 추세 함수를 이용하여 평가 스트림 추세를 측정하고, 각각의 아이템 평가 스트림 추세들을 비교하여 가장 작은 평가 스트림 추세를 선택한다. 이 때 선택 되어진 평가 스트림 추세를 최저 평가 스트림 추세라 정의한다. 다음 평가 스트림 추세를 측정할 시간 주기가 되면 다시 각각의 아이템 평가 스트림 추세를 측정하여 최저 평가 스트림 추세를 갱신한다.

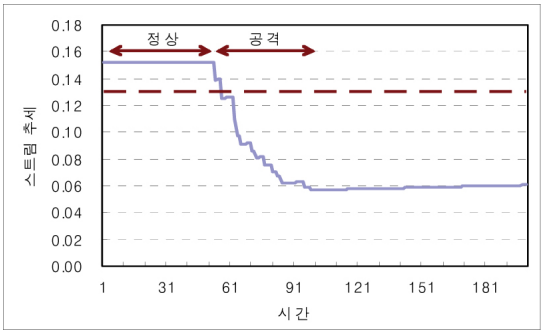
(그림 5)는 100개 아이템에 대한 평가 스트림 추세를 나타낸 예이다.

아이템의 평가 스트림 추세는 그림과 같이 아이템들이 받은 평가와 시간에 따라 다양한 형태로 표현되며, 그림에서의 점선은 이전 단계에서 얻은 최저 평가 스트림 추세이다.

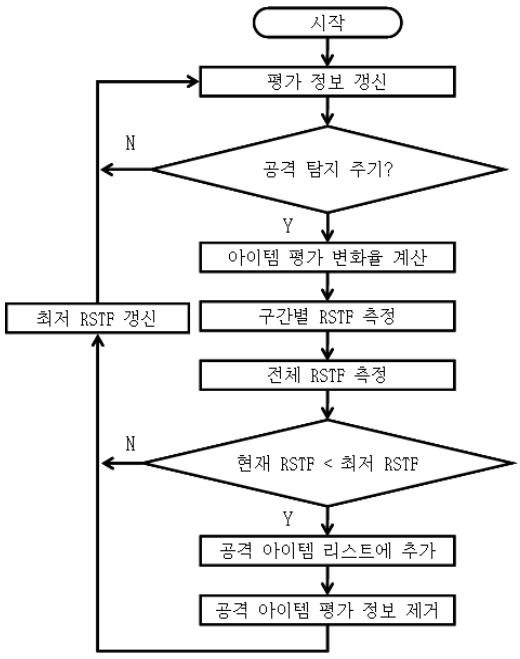
공격에 대한 탐지는 현재 단계에서 측정된 아이템의 새로운 평가 스트림 추세를 이전 탐지 단계에서 측정한 평가 스트림 추세와 비교하여, 현재 단계의 새로운 평가 스트림 추세가 이전 단계의 최저 평가 스트림 추세보다 작은 경우를 공격받은 것으로 결정한다. 최저 평가 스트림 추세는 추천 시스템에 등록된 모든 아이템들의 평가 스트림 추세와 비교하여 추출한다. 만약 공격을 탐지할 때 모든 아이템들을 고려하지 않고 한 아이템에 대한 평가 스트림 추세만을 고려한다면 시스템이 활성화되어 아이템들에 대한 거래가 전체적으로 증가한 상황에서 공격 탐지를 수행할 경우, 정상적인 아이템 활성화를 공격으로 탐지하는 잘못된 결과를 초래할 수 있다. 그러나 본 논문에서 제안하는 평가 스트림 추세는 시스템에 등록된 모든 아이템들의 평가 스트림 추세를 고려하여 추출한다. 이와 같은 방법은 임의의 아이템에 대한 평가량이 급증하였다고 해도 다른 아이템들의 평가량이 얼마나 급증하였는지 고려하기 때문에, 사용자 증가로 인한 탐지 오류도 방지할 수 있다. 그러므로 이와 같은 상황에서도 평가 스트림 추세 함수는 정상적인 작업을 수행할 수 있으며, 시스템 사용량이 급증하는 최악의 경우에도 모든 아이템의 평가 스트림 추세를 비교하여 최저 평가 스트림 추세를 결정하기 때문에 추천 공격을 쉽게 탐지 할 수 있다.

(그림 6)은 아이템의 평가 스트림 추세 변화와 공격탐지를 함께 나타낸 것이다. 그림에서 X축은 시간의 흐름을 나타내며, Y축은 평가 스트림 추세 변화를 나타낸다.

그림에서 점선은 최저 평가 스트림 추세이다. 추천 공격이 수행되면 공격이 진행되는 동안 아이템의 평가 스트림 추세 함수는 빠르게 변화하며 그림에서와 같이 급격히 떨어지게 된다. 이와 같은 상황에서 평가 스트림 추세 함수가 급격히 떨어지는 양상이 최저 평가 스트림 추세보다 작아지면 공격으로 탐지한다.



(그림 6) 평가 스트림 추세 변화와 공격 탐지



(그림 7) 추천 공격 탐지에 대한 흐름도

(그림 7)은 평가 스트림 추세를 이용하여 추천 공격을 탐지하는 순서도이다.

추천 시스템 가동이 시작되면 사용자의 아이템에 대한 평가는 지속적으로 갱신되며, 갱신되는 상황을 평가 스트림 추세 함수는 주기적으로 검사를 수행한다. 아이템의 평가 스트림 추세를 검사할 때는 아이템에 대한 평가 변화율을 분석하여 아이템의 구간별 평가 스트림 추세를 생성하다. 구간별 평가 스트림 추세를 이용하여 전체 평가 스트림 추세를 생성하고, 모든 아이템에 대한 평가 스트림 추세와 비교하여 최저 평가 스트림 추세를 생성한다. 현재 검사 단계의 최저 평가 스트림 추세와 이전 검사 단계의 최저 평가 스트림 추세를 비교하여 최저 평가 스트림 추세를 갱신한다. 아이템의 평가 스트림 추세가 최저 평가 스트림 추세보다 작을 경우에 공격으로 결정한다. 공격이 탐지되었을 경우 공격받은 아이템을 공격 리스트에 추가하고, 공격 리스트에 추가된 아이템의 평가 정보는 평가 데이터에서 삭제한다. 다음 공격이 발생할 때까지 추천 시스템은 재가동 되어 평가 정보를 누적한다. 본 논문에서 제안하는 평가 스트림 추세 분석은 추천 시스템의 공격을 탐지할 수 있으며, 노이즈가 발생한 아이템을 추출할 수 있는 장점이 있다. 또한 평가 스트림 추세 분석은 주기적으로 추천 공격을 탐지하기 때문에 실시간 추천 공격에 대응할 수 있다.

4. 실험 및 결과

4.1 실험 환경 및 가상 데이터

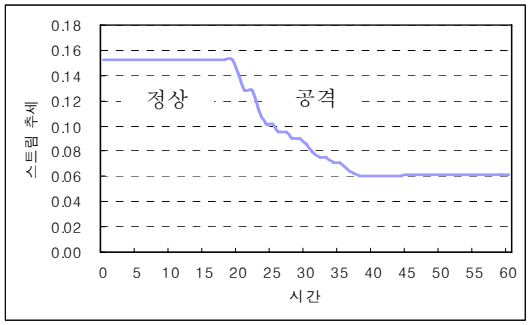
실험에서는 추천 시스템에 대한 공격 탐지를 위해 가상 데이터를 생성하였으며, 가상 데이터는 크기에 따라 세 가지로 나뉘고, 각 크기에 따른 가상 데이터는 평가 유형에 따라 다시 세 가지로 나뉜다. 크기에 따른 첫 번째 가상 데이터는 500명의 사용자와 500개의 아이템으로 구성된 데이터이고, 두 번째 가상 데이터는 700명의 사용자와 700개의 아이템으로 구성된 데이터이며, 세 번째 가상 데이터는 1,000명의 사용자와 1,000개의 아이템으로 구성된 데이터이다. 실험은 CPU 2.4GHz, 주기억장치 2GB, 윈도우즈XP 환경에서 수행하였으며, 10-fold cross validation을 적용하였다. 추천 성능 비교에는 적중률을 사용하였으며, 아이템 추천에는 TopN을 적용하였다.

첫 번째 가상 데이터는 평가에 대한 원시 데이터로 시간 흐름에 따라 각 사용자들이 아이템에 대해 평가를 수행한 데이터이다. 평가 데이터에

포함된 아이템들은 20개의 그룹 중 하나에 포함되도록 하였으며, 아이템 그룹은 아이템의 범주를 의미한다. 일반적으로 아이템들은 특정 범주에 속하기 때문에, 20개의 범주를 생성하여 하나의 아이템은 하나의 범주에 속할 수 있도록 랜덤하게 아이템 범주를 생성하였다. 상용화 추천 시스템의 평가 데이터에는 아이템 평가가 매우 희소하게 나타나기 때문에, 가상 데이터를 생성할 때는 희소성을 고려하여 평가가 나타난 아이템은 전체 아이템의 10%를 넘지 않도록 설정하였다. 일반적으로 사용자가 많은 아이템들에 대해 평가를 수행한 경우는 극히 드물며, 이와 같은 현실을 고려하여 사용자가 아이템에 평가를 수행할 때는 4개의 그룹을 랜덤하게 선택하여 아이템 평가를 생성하였다. 생성된 원시 평가 데이터는 블리언 형태로 존재한다. 원시 평가 데이터에서 아이템 평가값이 '1'인 경우는 해당 아이템을 양성으로 평가한 것을 의미하고, '0'인 경우는 해당 아이템에 대해 평가가 음성으로 평가된 것을 의미한다. 사용자 평가는 시간의 흐름에 따라 일정한 시간 주기 내에서 랜덤하게 아이템을 평가하도록 하여 원시 평가 데이터를 생성하였다.

두 번째 가상 데이터는 원시 데이터에 공격을 수행한 공격 데이터이다. 원시 데이터에 공격을 수행할 경우에는 랜덤하게 범주를 선택하여 선호도 공격을 수행하였으며, 원시 데이터를 공격하여 생성된 데이터를 공격 데이터로 사용하였다. 공격은 짧은 시간 내에 아이템에 대한 평가가 많이 발생해야 아이템에 대한 추천값이 빠르게 변화되어 공격 효과를 높일 수 있기 때문에, 공격 데이터는 공격 구간을 원시 데이터의 평가 구간에 비해 짧게 설정하였다. 그리고 공격량은 전체 사용자의 10%를 랜덤하게 선정하여 공격자로 지정하였으며, 공격의 대상이 되는 공격 아이템은 전체 아이템 중에서 10%를 랜덤하게 선정하여 공격을 수행하였다.

세 번째 가상 데이터는 재평가 데이터이며, 재평가 데이터는 공격이 수행된 데이터에서 공격받은 아이템을 제거한 후에 다시 정상적인 평가가



(그림 8) 시간 변화에 따른 평가 스트림 추세

이루어진 데이터이다. 공격받은 평가 데이터를 사용하지 않으면 새롭게 평가 데이터를 구축하여야 한다. 시스템이 사용자의 반응을 이용하여 평가 데이터를 구축하는 데는 오랜 시간이 소요되며, 협동적 여과를 사용한 추천 시스템에서는 유사 사용자들의 평가가 추천 결과에 적용되기 때문에 평가 데이터를 생성하는 데 많은 시간이 필요하다. 또한 완전한 평가 데이터 구축이 이루어질 때까지 추천 시스템은 추천을 수행할 수 없다는 문제가 발생한다. 추천 공격에 의해 발생하는 여러 문제들을 해결하기 위해서는 공격 데이터에서 공격받은 아이템을 제거한 후에 평가 데이터를 추천 시스템에서 재사용하여야 하며, 공격받은 평가 데이터의 재사용성을 측정하기 위해 재평가 데이터를 생성하였다.

(그림 8)은 500×500 크기의 공격 데이터에서 시간 변화에 따른 아이템의 평가 변화를 나타낸 그림이다.

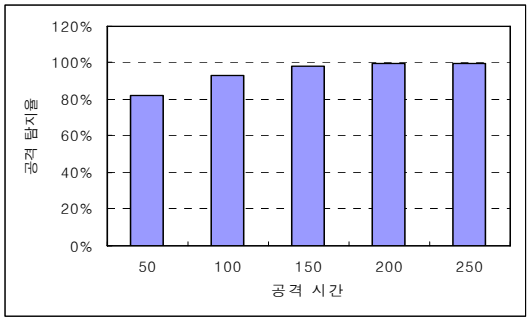
그림에서 X축은 시간 흐름에 따른 평가 스트림 추세 검사 횟수를 표시한 것이고, Y축은 평가 스트림 추세를 검사하는 주기에 따른 아이템의 평가 스트림 추세 함수 값이다. X축을 기준으로 20 미만 구간은 정상적으로 아이템 평가가 이루어진 구간이고, 20에서 40까지의 구간은 추천 공격이 수행된 영역이다. 정상적인 평가 구간에서는 평가 스트림 추세에 따른 아이템의 평가 변화가 완만한 형태를 나타나지만, 공격 구간에서는 아이템 평가에 대한 공격이 발생하여 평가 스트림 추세

에 따른 아이템 평가 변화가 큰 폭으로 변화한다는 것을 알 수 있다. 추천 시스템에 대한 공격은 빠른 시간 내에 추천 성능을 저하시켜야 하는 것을 목적으로 하기 때문에, 공격 평가 스트림은 정상적인 아이템의 평가 스트림 추세보다 급격한 변화가 발생한다. 본 논문에서 제안하는 평가 스트림 추세 분석을 이용하여 아이템의 평가 변화를 측정하면 추천 공격을 탐지할 수는 장점이 있다.

추천 공격 실험에서는 협동적 여과 기법을 이용하여 추천 성능 측정하였으며, 성능 측정을 위해 적중률을 사용하였고, 10-fold cross validation을 적용하여 실험을 수행하였다. 실험 절차는 다음과 같다. 먼저 실험 데이터를 10개의 동일한 크기의 집단으로 나눈다. 1개 집단을 테스트 집단으로 활용하고, 나머지 9개 집단을 훈련 집단으로 활용한다. 10개의 집단이 각각 1번은 테스트 집단이 되도록 교차하여 실험을 수행한다. 테스트 집단이 결정되면 테스트 집단에서 테스트 사용자를 선출한다. 테스트 사용자의 선호 아이템을 테스트 아이템으로 결정하고, 테스트 아이템의 평가 값을 삭제한다. 다음으로 사용자들 간의 유사도를 측정하고, 유사도를 이용하여 가장 유사한 K명의 사용자를 추출한다. 그리고 성향이 유사한 K명의 아이템 선호도를 계산한다. 선호도에 의해 추출된 상위 N개의 아이템 리스트를 테스트 사용자에게 추천한다. 테스트 아이템이 추천 리스트에 포함되어 있으면 적중한 것으로 결정한다. 테스트 집단에 포함된 모든 테스트 사용자의 테스트가 종료될 때까지 반복 수행한다. 테스트가 완료되면 추천 정확도를 적중률로 나타낸다.

4.2 실험 결과

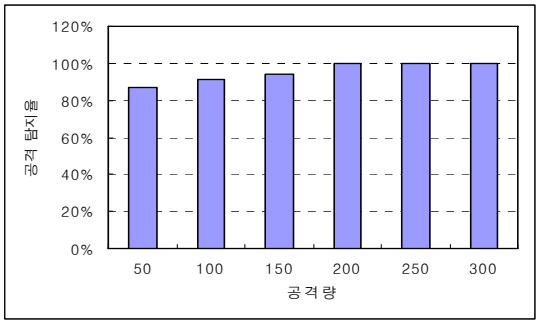
실험은 크게 두 가지로 나뉘며, 하나는 공격 탐지 실험이고, 다른 하나는 성능 측정 실험이다. 첫 번째 실험은 평가 스트림 추세를 이용하여 추천 시스템에 대한 공격을 탐지하는 실험이며, 실험에서는 공격 시간과 공격량의 변화에 따른 공격 탐지를 측정하였다. 평가 스트림 추세는 시간의 흐름에 따른 아이템에 대한 평가 정보를 활용하며, 평가 정보에 대한 스트림 데이터를 주기적으로 분석하여 평가 스트림 추세를 측정하였다. 성능 측정 실험에서는 공격이 추천 시스템의 성능에 미치는 영향을 확인하기 위해 원시 데이터와 공격 데이터를 활용하여 추천 성능을 비교하였으며, 추천 공격에 의해 노이즈가 발생한 아이템을 제거한 후 추천 시스템이 정상적인 추천을 수행할 수 있는지 실험하기 위해 공격 제거 데이터와 재평가 데이터를 이용하여 추천 성능을 측정하였다.



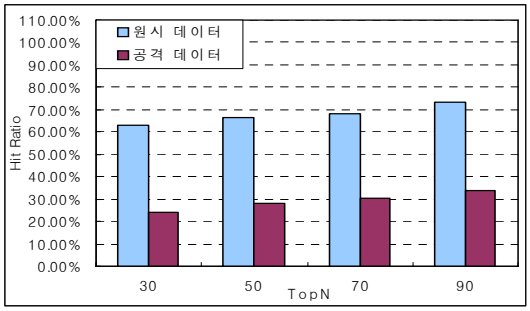
(그림 9) 시간에 따른 공격 탐지

(그림 9)는 공격 시간에 따른 공격 탐지 실험 결과이며, 실험에서는 1,000×1,000 데이터를 활용하였다.

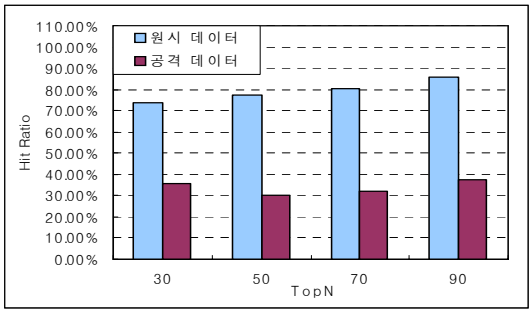
공격 시간 증가에 따른 공격 탐지 실험에서는 공격 시간을 50, 100, 150, 200, 250까지 증가시키며, 평가 스트림 추세를 이용하여 추천 공격 탐지를 수행하였다. 공격 탐지율은 실제 공격받은 아이템 중 공격이라고 탐지된 아이템의 비율이며, 평가 스트림 추세를 측정하여 공격 받은 아이템을 탐지하였다. 공격 시간이 50일 때는 공격받은 아이템에 대한 탐지율이 82%였으며, 공격 시간이 100일 때는 93%, 공격 시간이 150일 대는 98%, 공격 시간이 200이상에서는 공격 받은 아이템 탐지율이 100%였다. 공격 시간이 적은 경우에는 공격 시간이 많은 경우보다 낮은 탐지 성능을 나타낸다. 공격 시간이 적으면 원시 데이터의 평가값에 작용한 공격 평가값의 정보량이 한계가 있으므로 낮은 탐지 성능을 나타내는 것이다. 그러나 공격



(그림 10) 공격량에 따른 공격 탐지



(그림 11) 원시 데이터와 공격 데이터의 적중률(500×500)



(그림 12) 원시 데이터와 공격 데이터의 적중률(700×700)

시간이 길어지면 공격 평가값은 원시 데이터의 평가값에 많은 영향을 미칠 수 있을 정도의 정보량이 쌓이게 되어, 원시 평가 데이터는 공격 평가값에 종속적이게 되므로 추천 공격 탐지가 우수하게 나타난다. 공격 시간이 길어질수록 공격을 수행하는 시간 동안 측정된 평가 스트림 추세에 급격한 변화가 발생한다. 그러므로 실험에서와 같이 공격 시간이 증가할수록 평가 스트림 추세 변화량이 커지므로 공격 탐지율이 높아진다. 그리고 공격 시간이 증가함에 따라 원시 평가값은 본래의 특성을 잃게 되고, 전체 평가값에서 차지하는 원시 평가값의 비율은 공격 시간이 증가함에 따라 매우 적어진다. 즉, 원시 평가 데이터의 평가값은 공격 수행 시간이 길어질 경우 공격 평가값에 많은 영향을 받게 된다. 본 논문에서 제안한 기법이 공격 시간에 따른 공격 탐지를 우수하게 수행할 수 있다는 것을 본 실험으로 확인하였다.

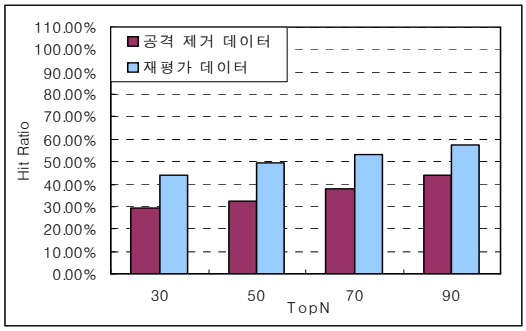
(그림 10)은 공격량의 변화에 따른 공격 탐지 실험 결과이며, 실험에는 1,000×1,000 데이터를 활용하였다. 공격량은 공격 시간 동안 공격에 참여한 사용자의 수를 의미한다.

공격량의 변화에 따른 공격 탐지 실험에서는 공격량을 50, 100, 150, 200, 250, 300으로 증가시켜 공격 탐지율을 측정한다. 실험 결과에서 공격량이 50일 때는 공격 탐지율이 82%이며, 공격량이 100일 때는 91%, 공격량이 150일 때는 94%, 공격량이 200이상일 때는 공격 아이템을 100% 탐지하였다. 평가 스트림 추세는 아이템별로 사용자들의 평가를 반영하기 때문에 공격량이 증가하면 아이템에 대한 평가 스트림 추세 변화량이 커지게 되므로 공격 시간이 짧더라도 공격 탐지율이 증가한다. 공격량이 증가함에 따라 공격 탐지율이 높아지는 것을 여러 실험 구간에서 확인할 수 있다.

(그림 11)은 500×500 데이터에 대한 원시 데이터와 공격 데이터에 대한 실험 결과이고, (그림 12)는 700×700 데이터에 대한 원시 데이터와 공격 데이터에 대한 실험 결과이다. 아이템 추천에는 협동적 여과 기법을 활용하였고, 테스트 사용자에게 아이템을 추천하기 위한 선호도 벡터를 만들기 위해서 테스트 사용자와 성향이 유사한 20명의 선호도 정보를 활용하였다. N은 사용자에게 추천한 아이템의 수로써 선호도 벡터에서 선호도 값이 가장 높은 아이템을 순서대로 N개를 선정한 것이다. 500×500 데이터에 대한 실험에서 원시 데이터에 대한 적중률은 N=30일 때 62.94%, N=50일 때 66.38%, N=70일 때 68.20%, N=90일 때

73.21%로 나타났다. 500×500 데이터에 대한 실험에서 평균 적중률은 67.68%로 나타났다. 700×700 데이터에 대한 실험에서 원시 데이터와 공격 데이터에 대한 실험 결과는 다음과 같다. 원시 데이터에 대한 적중률은 N=30일 때 74.01%, N=50일 때 77.29%, N=70일 때 80.28%, N=90일 때 85.89%로 나타났다. 700×700 데이터에 대한 실험에서 평균 적중률은 79.37%로 나타났으며, 추천 성능은 N값이 증가함에 따라 높은 적중률을 보였다. 그리고 협동적 여과 기법은 유사 사용자의 성향을 이용하여 아이템을 추천하는 방식을 취하기 때문에 유사 사용자의 수가 증가하면 사용자의 성향을 쉽게 분석할 수 있고, 이러한 유사 사용자들의 성향에 의해 추출된 아이템들은 선호도 순에 의해 리스트 되기 때문에 추천 아이템의 수량 증가에 따라 추천 정확도가 증가하는 것이다.

500×500 데이터에 공격이 수행된 후의 적중률은 N=30일 때 24.27%, N=50일 때 28.33%, N=70일 때 30.29%, N=90일 때 33.53%로 나타났다. 추천 공격이 수행된 후의 평균 적중률은 29.10%로 나타나, 원시 데이터의 평균 적중률과 38.58%의 높은 성능 차이를 나타냈다. 700×700 데이터에 공격이 수행된 후의 적중률은 N=30일 때 35.73%, N=50일 때 30.25%, N=70일 때 32.02%, N=90일 때 37.41%로 공격 후에 적중률이 큰 폭으로 낮아졌다. 본 실험 결과에서 확인할 수 있듯이 공격이 수행된 후에는 추천 시스템의 성능이 급격히 저하된다. 공격이 수행된 후에 이와 같이 낮은 적중률이 나타나는 이유는 추천 시스템에 대한 공격이 사용자의 성향과 관계없이 아이템에 대해 조작된 오류 평가들을 삽입하기 때문이다. 협동적 추천 방식은 사용자에 대해 아이템 추천을 수행할 때, 추천 대상 사용자와 유사 사용자를 추출하고, 유사 사용자들의 아이템들에 대한 선호도를 계산한다. 그리고 추천 대상 사용자가 평가하지 않은 아이템들 중에서 유사 사용자들의 선호도가 높은 아이템들을 추천하는 방식을 취한다. 그러므로 공격자가 추천 대상이 되는 사용자의 유사 사용자로 선정된다면, 공격자의 아이템에 대한 평가



(그림 13) 공격 제거 데이터와 재평가 데이터의 적중률 (500x500)

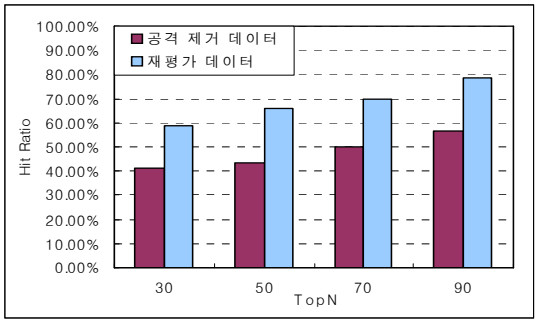
가 추천 대상 사용자의 아이템에 대한 평가를 왜곡시키는 문제를 발생시킨다. 즉, 추천 공격은 선호도가 유사한 사용자 그룹에 노이즈를 삽입하는 결과를 발생시키며, 추천 시스템은 삽입된 노이즈로 인해 유사 사용자 추출에 문제가 발생하여 추천 성능이 낮아진다.

700×700 데이터의 적중률이 500×500 데이터의 적중률보다 높게 나타난 이유는 700×700 데이터는 사용자 수가 500×500 데이터보다 많고 활용할 평가 정보도 많기 때문이며, 평가 정보가 풍부해지면 연관성이 높은 유사 사용자 추출이 수월해지기 때문에 높은 적중률을 나타낸다. 그리고 본 결과에서 알 수 있듯이 사용자의 수가 증가하여 평가 정보가 증가되어도 추천 공격은 원시 평가 정보보다 정보량이 많은 조작 평가값을 이용하여 공격을 수행하기 때문에 원시 평가의 성질을 쉽게 변경시킬 수 있으며, 원시 평가의 성질 변경은 추천 시스템의 성능과 직결되는 요인으로 작용하여 급격한 추천 성능 저하라는 문제를 발생시킨다.

(그림 13)은 500×500의 공격 제거 데이터와 재평가 데이터에 대한 실험 결과이다. 공격 아이템 제거한 후 공격 데이터에 대한 적중률은 평가 스트림 추세를 이용하여 공격받은 것으로 탐지된 아이템을 제거한 후의 추천 적중률이다. 공격 아이템 제거 후의 적중률은 N=30일 때 29.23%, N=50일 때 32.56%, N=70일 때 37.60%, N=90일 때

44.10%로 나타났다. 평균 적중률은 35.87%로 나타났으며, 공격 받은 상태에서의 적중률과 비교하면 평균 6.77%의 성능 향상이 나타난 것이다. 공격받은 아이템을 삭제하면 추천 공격으로 삽입된 노이즈가 제거되어 잘못된 아이템 추천이 발생하지 않기 때문에, 공격 아이템을 제거한 후의 적중률이 추천 공격이 발생했을 때의 적중률보다 높게 나타난 것이다. 이와 같은 방법을 취하면 아이템을 추천할 때 공격받은 아이템의 평가 정보는 선호도 계산에서 제외되므로 잘못된 추천을 방지할 수 있다. 그러나 공격받은 데이터를 삭제하여도 원시 데이터에 비해 매우 낮은 추천 성능을 나타내는 이유는 다음과 같다. 공격받은 아이템이 추천 대상에서 삭제되기 때문에 해당 아이템이 실제로 선호되는 경우에도 추천 대상 아이템에서 제외되는 문제가 발생한다. 그리고 공격받은 아이템이지만 아직 공격으로 탐지되지 않은 아이템들이 존재할 경우도 사용자의 선호도를 왜곡시킬 수 있기 때문에 공격 삭제 데이터의 적중률이 원시 데이터의 적중률보다 매우 낮게 나타난다. 또한 올바른 유사 사용자들이 평가한 고급 평가 정보가 공격에 의해 노이즈가 발생하여 고급 평가 정보를 제거하였기 때문에 유사 사용자들의 평가 정보를 활용하는 데 제약이 따르기 때문이다. 이와 같이 임의 공격은 추천 시스템의 성능 저하에 큰 영향을 미치는 최악의 공격 방법이다.

공격 아이템 제거 후에 정상적인 아이템 평가가 다시 수행된 재평가 데이터에서의 적중률은 N=30일 때 43.78%, N=50일 때 49.25%, N=70일 때 53.29%, N=90일 때 57.37%로 나타났다. 500×500 재평가 데이터에서의 평균 적중률은 50.92%로 나타났으며, 이와 같은 결과는 공격 아이템 제거 상태에서의 평균 적중률보다 15.05%의 성능 향상을 나타낸 것이며, 공격이 수행되었을 때의 평균 적중률과 비교하면 38.58%의 성능 향상을 나타낸 것이다. 이와 같이 결과를 보더라도 공격 아이템을 제거한 후에 평가 데이터를 재사용하여도 정상적인 사용이 가능하다는 것을 본 실험으로 알 수 있었다. 그러나 원시 데이터의 평균 적중률과



(그림 14) 공격 제거 데이터와 재평가 데이터의 적중률 (700×700)

비교하면 16.76%의 성능 차이가 발생하였다. 재평가 데이터는 공격받은 후 정상적인 평가가 다시 수행된 데이터이기는 하지만, 원시 데이터와 성능 차이가 발생하는 이유는 평가에 대한 공격이 정상 평가로 판단될 정도의 미세한 공격은 공격으로 탐지되지 않아, 미세한 공격 노이즈가 평가 데이터에 존재하기 때문이다. 이와 같은 미세한 공격 노이즈가 유사도 측정에 영향을 미치기 때문에, 공격 아이템을 제거한 후에 재평가가 수행된 재평가 데이터에서의 적중률이 원시 데이터보다 낮게 나타나는 것이다. 그러나 이와 같은 노이즈는 협동적 여과 기법의 특성으로 재평가 기간이 길어지면 아이템 평가의 정보 이용도는 증가되고, 시간에 따른 정상 평가량이 증가하게 되면 노이즈는 정상 평가에 희석되어 정상적인 원시 데이터의 순도에 근접한 평가 정보가 될 것이다.

(그림 14)는 700×700의 공격 제거 데이터와 재평가 데이터에 대한 실험 결과이다. 공격 제거 데이터의 적중률은 N=30일 때 41.18%, N=50일 때 43.59%, N=70일 때 49.81%, N=90일 때 56.74%로 나타났다. 공격 데이터의 적중률과 비교하면 평균 13.98%의 성능 향상이 나타난 것이며, 공격 아이템을 삭제하면 공격으로 인해 아이템에 삽입된 노이즈가 제거되기 때문에 공격받은 상태보다는 추천 성능이 향상된다. 500×500 데이터에서 적중률과 비교할 때 700×700의 공격 제거 데이터가 더 높은 적중률을 나타냈다. 이와 같은 결과가 발

생하는 이유는 정보량과 관계된 것이며, 500×500 데이터에 존재하는 평가 정보보다 700×700 데이터에 존재하는 평가 정보가 더 풍부하기 때문에 공격 아이템을 삭제하여도 추천 성능이 높게 나타나는 것이다.

재평가 데이터의 적중률은 N=30일 때 58.78%, N=50일 때 65.96%, N=70일 때 69.52%, N=90일 때 78.34%로 나타났다. 700×700 재평가 데이터의 평균 적중률은 68.15%로 나타났으며, 공격 데이터와 비교하면 45.52%의 성능 향상이 나타난 우수한 결과이고, 원시 데이터의 적중률과 비교하면 11.22%의 성능 차이가 나타난다.

평가 스트림 추세를 이용하여 공격 아이템을 제거한 후의 공격 데이터를 추천 시스템의 평가 데이터로 재사용하여도 원시 데이터에 근접한 추천 성능을 나타낼 수 있다는 것을 본 실험으로 확인하였다. 추천 공격의 의해 손상을 입은 평가 데이터를 사용하지 않는다면 추천 시스템은 평가 데이터를 확보할 때까지 추천을 수행할 수 없다. 또한 평가 데이터는 사용자의 반응을 사용하여 구축하기 때문에 활용성이 높은 평가 데이터를 구축하는 데 오랜 시간이 걸린다는 문제가 있다. 이와 같은 문제점들을 고려하면 재평가 데이터의 사용은 추천 시스템의 운용성을 극대화시키는 방법이며, 평가 데이터의 재사용성을 높이는 방법이기도 하다.

5. 결론 및 향후 연구

협동적 여과 방식의 추천은 아이템들에 대한 사용자들의 선호도 정보를 활용하여 아이템 추천을 수행하는 기법으로 다른 추천 기법에 비해 추천 성능이 뛰어나 가장 널리 활용되는 방법이다. 그러나 협동적 여과를 이용한 추천 시스템은 시스템 공격자가 악의적 목적을 가지고 아이템에 대한 선호도를 조작하였을 때 추천 성능이 저하되는 문제가 발생한다.

본 논문에서는 평가 스트림 추세 분석을 이용하여 추천 공격을 탐지하는 방법을 제안한다. 평가 스트림 추세 분석은 시간의 흐름에 따른 사용자 선호도 변화를 측정하여 아이템의 인기도나 평가 추세를 표현할 수 있으며, 일정한 시간 주기로 아이템의 평가 스트림 추세 변화를 측정하여 추천 공격을 탐지할 수 있다. 다양한 실험을 통해 추천 공격이 추천 시스템의 추천 성능을 저하시킨다는 것을 확인할 수 있었으며, 평가 스트림 추세 분석을 추천 공격에 적용하면 쉽게 추천 공격을 탐지할 수 있다는 것을 실험으로 확인하였다. 평가 스트림 추세 분석은 공격 시간과 공격자 수가 증가함에 따라 공격 탐지율이 높게 나타났으며, 이와 같은 결과가 발생하는 이유는 공격 시간과 공격자의 수가 증가하면 평가 스트림 추세 변화량이 커지게 되어 공격 탐지율이 높아지기 때문이다. 추천 공격을 탐지하였을 경우 공격받는 아이템을 평가 데이터에서 제거함으로써 추천 적중률을 높일 수 있다는 것과 공격 아이템을 제거한 후에 평가 데이터를 재사용함으로써 원시 데이터를 이용한 추천 성능에 근접할 수 있다는 것을 여러 실험을 통해 확인하였다. 추천 공격으로 오염된 평가 데이터를 추천 시스템에서 활용하지 않으면 추천 시스템은 평가 데이터를 구축할 때까지 추천을 수행할 수 없다는 문제가 발생하며, 일반적인 평가 데이터는 사용자의 반응을 이용하기 때문에 구축하기까지 오랜 시간이 걸린다는 문제가 있다. 이러한 현실적 상황을 고려할 때 공격이 발생한 평가 데이터를 재사용한다는 것은 추천 시스템의 운용성 측면이나 평가 데이터의 재사용성 측면에서 매우 중요한 방법이다. 본 논문에서 제안한 평가 스트림 추세 분석은 추천 공격을 탐지하여 공격 받은 아이템을 추출할 수 있는 장점이 있으며, 오염된 평가 데이터를 재사용할 수 있는 방법을 제공한다. 그러므로 본 논문에서 제안한 기법은 추천 시스템이 적용된 일반적인 상거래 시스템에서 문제가 되고 있는 공격에 대한 해결 방안과 견고한 추천 시스템 구축에 적용될 수 있다.

향후 연구로는 추천 공격이 평가 데이터에 미치는 영향을 평가 정보량에 따른 분석을 통해 추

천 공격과 평가 정보량의 연관관계로 측정하여 공격으로 오염된 평가 데이터의 재사용성에 관한 추가적인 연구가 필요하다.

참 고 문 헌

[1] S. Lam and J. Riedl, “Shilling Recommender Systems for Fun and Profit,” Proceedings of the 13th International WWW Conference, 2004.

[2] J. O’Donovan and B. Smyth, “Is Trust Robust?: An Analysis of Trust-based Recommendation,” Proceedings of the 5th ACM Conference on Electronic Commerce, 2006

[3] J. Schafer, J. Konstan, and J. Riedl, “Recommender System in E-Commerce,” Proceedings of the ACM Conference on Electronic Commerce, 1999.

[4] M. Pazzani, “A Framework for Collaborative, Content-based and Demographic Filtering,” Artificial Intelligence Review, pp.393-408, 1999.

[5] P. Melville, R. Mooney, and R. Nagarajan, “Content-Boosted Collaborative Filtering,” Proceedings of the SIGIR-2001 Workshop on Recommender Systems, 2001.

[6] U. Shardanand and P. Maes, “Social Information Filtering: Algorithms for Automating ’Word of Mouth’,” Proceedings of the Conference of Human Factors in Computing Systems, 1995.

[7] J. Konstan, B. Millr, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, “GroupLens: Applying Collaborative Filtering to Usenet News,” Communications of the ACM, Vol.40, No.3, pp.77-87, 1997.

[8] G. Linden, B. Smith, and J. York, “Amazon.com Recommendations: Item-to Item Collaborative Filtering,” IEEE Internet Computing, 2003.

[9] M. Condliff, D. Lewis, D. Madigan, and C. Posse, “Baysian Mixed-Effect Models for Recommender Systems,” Proceedings of the Recommender Systems Workshop at SIGIR-99, 1999.

[10] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, “An Algorithmic Framework for Performing Collaborative Filtering,” Proceedings of the ACM SiGIR-99, 1999.

[11] J. Herlocker, J. Konstan, C. Tervin, and J. Riedl, “Evaluating Collaborative Filtering Recommender Systems,” Proceedings of the ACM Transactions on Information Systems 22, 1, 2004.

[12] H. Kim, J. Kim, and J. L. Herlocker, “Feature-Based Prediction of Unknown Preferences for Nearest-Neighbor Collaborative Filtering,” Proceedings of the 4th IEEE International Conference on Data Mining, 2004.

[13] A. Poposcul, L. Ungar, D. Pennock, and S. Lawrence, “Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments,” Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence, 2001.

[14] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, “Item-based Collaborative Filtering Recommendation Algorithms,” Proceedings of the 10th International WWW Conference, 2001.

[15] P. Chirita, W. Nejdl, and C. Zamfir, “Preventing Shilling Sttacks in Online Recommender Systems,” Proceedings of the 7th annual ACM International workshop on Web Information and Data Management, 2005.

[16] B. Mobasher, R. Burke, R. Bhaumik and C. Williams, “Effective Attack Models for Shilling Item-based Collaborative Filtering Systems,” Proceedings of the 2005 WebKDD Workshop,

2005.

[17] G. Adomavicius and A. Tuzhilin, “Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions,” IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.6, pp.734-749, 2005.

[18] M. O’Mahony, N. Hurley, N. Kushmerick and G. Silvestre, “Collaborative Recommendation: A Robustness Analysis,” ACM Transactions on Internet Technology Vol.4, No.4, 2004.

[19] R. Burke, B. Mobasher, and R. Bhaumik, “Limited Knowledge Shilling Attacks in Collaborative Filtering Systems,” Proceedings of the 3rd IJCAI Workshop in Intelligent Techniques for Personalization, 2005.

[20] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, “Classification Features for Attack Detection in Collaborative Recommender Systems,” Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2006.

[21] C. Dellarocas, “Immunizing Online Reputation Reporting Systems against Unfair Ratings and Discriminatory Behavior,” Proceedings of the ACM Conference on Electronic Commerce, 2000.

◐ 저 자 소 개 ◑

김 용 욱

1999년 동국대학교 컴퓨터공학과 학사
2001년 동국대학교 대학원 컴퓨터공학과 석사
2001년~현재 동국대학 컴퓨터공학과 박사과정
관심분야 : 기계학습, 추천시스템, 의료영상, 인공지능
E-mail : yukim@dongguk.edu

김 준 태

1986년 서울대학교 제어계측공학과 공학사
1990년 Univ. of Southern California, Computer Engineering M.S.
1993년 Univ. of Southern California, Computer Engineering Ph.D.
1994년 Univ. of Southern California, Computer Science and Engineering Postdoc
1995년~현재 동국대학 컴퓨터공학과 교수
관심분야 : 인공지능, 기계학습, 데이터마이닝, 정보검색
E-mail : jkim@dongguk.edu