

# Weighted Window Assisted User History Based Recommendation System

Sungmin Hwang<sup>†</sup> · Rajashree Sokasane<sup>\*\*</sup> · Hiep Tuan Nguyen Tri<sup>†</sup> · Kyungbaek Kim<sup>\*\*\*</sup>

## ABSTRACT

When we buy items in online stores, it is common to face recommended items that meet our interest. These recommendation system help users not only to find out related items, but also find new things that may interest users. Recommendation system has been widely studied and various models has been suggested such as, collaborative filtering and content-based filtering. Though collaborative filtering shows good performance for predicting users preference, there are some conditions where collaborative filtering cannot be applied. Sparsity in user data causes problems in comparing users. Systems which are newly starting or companies having small number of users are also hard to apply collaborative filtering. Content-based filtering should be used to support this conditions, but content-based filtering has some drawbacks and weakness which are tendency of recommending similar items, and keeping history of a user makes recommendation simple and not able to follow up users preference changes. To overcome this drawbacks and limitations, we suggest weighted window assisted user history based recommendation system, which captures user's purchase patterns and applies them to window weight adjustment. The system is capable of following current preference of a user, removing useless recommendation and suggesting items which cannot be simply found by users. To examine the performance under user and data sparsity environment, we applied data from start-up trading company. Through the experiments, we evaluate the operation of the proposed recommendation system.

**Keywords :** Recommendation System, Content-Based Filtering, User History, Window

## 가중 윈도우를 통한 사용자 이력 기반 추천 시스템

황 성 민<sup>†</sup> · Rajashree Sokasane<sup>\*\*</sup> · Hiep Tuan Nguyen Tri<sup>†</sup> · 김 경 백<sup>\*\*\*</sup>

## 요 약

온라인에서 물품을 구매하고자 할 때, 추천 시스템은 사용자에게 맞춘 추천을 하게 되며, 사용자가 관심을 가질만한 새로운 물품까지 추천해 준다. Collaborative filtering 등, 여러 모델들이 보다 정확한 추천을 위해 제안되었으며, 활발히 연구되고 있다. 그중 Collaborative filtering은 사용자 선호도를 예측하는 데 좋은 결과를 보여주지만 사용자 개체 및 데이터가 부족한 환경에서는 사용자들끼리의 비교를 힘들게 하여 collaborative filtering이 적용되기 힘들게 한다. 새로 시작하는 시스템이거나 사용자 개체 수가 적은 경우, 문제가 발생하며, 이와 같은 상황에서는 content-based filtering이 사용된다. 하지만 content-based filtering은 비슷한 물건만 추천해주거나, 사용자 성향 변화를 제대로 반영하지 못하는 등의 여러 단점을 가지고 있다. 이러한 한계점들을 극복하기 위해서 사용자 구매 기록에 가중 윈도우를 적용하고, 사용자 구매 기록 분석을 통한 윈도우 가중치 조정을 수행하는 시스템을 제안한다. 사용자 성향의 변화에 민감하게 반응할 수 있고, 이를 기반으로 무의미한 추천을 제거하며, 사용자가 찾기 어려운 관련 물품 추천이 가능한 새로운 상품도 추천하는 시스템을 제시하며, 언급된 사용자 개체 및 데이터가 부족한 환경에서의 동작을 검증하기 위해, 스타트업 무역업체에서 제공된 상품정보 기반 실험을 통해 제안된 시스템의 동작을 검증하였다.

**키워드 :** 추천 시스템, 콘텐츠 기반 필터링, 사용자 이력, 윈도우

## 1. 서 론

오늘날, 물품을 구매하기 위해 상업적 웹사이트를 방문하

게 되면, 각각의 웹사이트는 사용자의 구매 성향에 맞추어 적절한 물품을 추천해준다. 이들 추천 시스템은 사용자에게 관심 물품을 찾는 수고를 덜어줄 뿐만 아니라, 사용자가 관심을 가질만한 아이템을 제안하여, 사용자가 잘 알지 못하는 연관된 아이টে까지 발견하여 구매할 수 있게끔 한다. 이렇게 상업적 수익과 밀접하게 관련된 추천 시스템은 현재까지 content-based filtering, collaborative filtering, hybrid 모델 등[8] 여러 방향의 연구가 진행되고 있으며[5], 각 상황에 맞는 추천 시스템 모델들이 다양하게 제시되고 있다[2].

※ 이 논문은 교육과학기술부의 LINC사업의 지원을 받아 수행되었음(2014-C-7246-010115). 이 논문은 2014년 미래창조과학부의 재원으로 한국연구재단의 지원을 받아 수행되었음(NRF-2014R1A1A1007734).  
† 준 회원 : 전남대학교 전자컴퓨터공학부 석사과정  
\*\* 준 회원 : 전남대학교 전자컴퓨터공학부 박사과정  
\*\*\* 종신회원 : 전남대학교 전자컴퓨터공학부 조교수  
Manuscript Received : January 14, 2015  
First Revision : February 27, 2015  
Accepted : February 27, 2015  
\* Corresponding Author : Kyungbaek Kim(kyungbaekkim@jnu.ac.kr)

이러한 모델들은 널리 사용되고 있으며, 좋은 결과를 보여주고 있지만, 모든 환경에 적용할 수 있는 것은 아니다. 여러 문제점 중 가장 널리 거론되는 문제점은 데이터 부족으로서, 유저 및 아이템에 대한 데이터가 충분하지 않을 때, 이러한 문제가 발생하며, 주로 새 사용자가 가입하거나 사용자 개체 수가 충분하지 않을 때 발생하게 된다.

새로운 사용자가 가입을 할 경우, 그 사용자가 여러 물품에 대해서 점수를 매기거나 관심 물품으로 등록을 한 정보가 존재하지 않으므로, 그 사용자가 어떤 물품을 선호하는지, 또는 어떤 사용자와 비슷한 구매 성향을 보이는지를 판단하기가 어려우며, 따라서 물품에 대해 매긴 점수와 그를 통한 사용자들 간의 유사도를 통해 관심 물품을 예상하고 추천하는 collaborative filtering 방법은 사용하기가 어렵게 된다. 한편 사이트 관점에서 바라볼 때는 사이트가 시작되지 얼마 되지 않거나, 소규모 사이트라서 충분한 사용자 개체 수를 확보하지 못한 경우 또한 사용자들 간의 비교를 힘들게 만들므로 CF 방법이 사용되기 어렵다. 이러한 사용자 데이터 부족으로 인해 발생하는 cold-start 문제로 인해, 이와 같은 환경에서는 각 사용자 개개인의 기록을 사용하여 물품을 추천하는 방법을 사용할 수밖에 없다.

앞의 경우에서는 Content-based를 기반으로 사용자 각각에 맞추어 물품을 추천하는 방법이 사용될 수밖에 없는데, content-based 시스템의 경우, 몇 가지 문제들과 한계점이 있어, 정확한 예상을 힘들게 한다. 첫째, 사용자의 구매 이력을 기반으로 물품을 추천할 경우, 사용자의 이력과 비슷한 물품만 추천하게 되는 경우가 있다. 사용자의 이력에 있는 물품을 기반으로 그와 가장 흡사한, 거리가 가까운 물품들을 주로 추천하게 되기 때문이다. 둘째, 사용자의 발전, 혹은 구매 성향 변화를 반영하기 어렵다. 사용자가 물품을 구매하기 시작한 뒤, 시간이 지나 관심 분야 및 선호 분야가 달라질 수 있지만 사용자의 이력을 그대로 반영하게 되면, 예전 기록에 주로 영향을 받아서 전과 비슷한 물품을 추천하게 된다. 이들 문제는 사용자 요구사항을 반영하기 어렵게 만들 뿐만 아니라, 사용자가 관심을 가질만한 새로운 영역의 물품을 추천하기 어렵게 만든다. Content-based filtering의 경우, cold-start 문제의 해결방안으로 처음 가입하는 사용자에게 선호 및 관심 분야를 선택하게 하는 방법이 있는데, 사용자의 선호도는 시간에 따라 변하므로 앞의 경우와 마찬가지로 적합한 물품을 추천하는 데 어려움이 있을 수 있으며, 또한 시장의 트렌드가 급격하게 변하고 사용자가 이에 반응하여 구매 성향이 변하는 경우에는 대응하기 어렵다.

Content-based filtering의 문제점들과 한계점을 극복하기 위해, 우리는 weighted-window 기반 추천 시스템을 제안한다. 이 시스템에서는 아이템들 사이의 유사도 그래프를 여러 feature들을 적용하여 구성한 뒤, 사용자 개인의 이력 및 이력의 부분을 관리하는 window와 그 그래프와의 연계를 통해 문제를 해결하고자 한다. 사용자의 구매 이력을 사용하여 아이템을 추천하는 이 시스템은 기존 content-based 방법과는 다르게, weighted-window를 기반으로 하여 사용

자의 최신 구매 성향을 반영할 수 있게 하였으며, 단지 사용자 구매 이력 각각의 아이템과 비슷한 물품을 추천하는 것에서 벗어나, 종합적으로 연관된 물품을 새롭게 추천할 수 있게 하였다. 이를 위해, 우리는 window를 사용자 구매 기록에 적용하여 사용자 구매 기록의 일부분을 시간, 또는 구매 횟수에 따라 저장하고, window 안의 각각의 물품마다 구매 시기 혹은 순서에 따라 서로 다른 비중을 두고, 이를 이용하여 사용자의 구매 성향 변화 및 시장 트렌드에 따른 변화에 대응할 수 있게 하였다.

Weighted-window를 추천 시스템에 적용하여, 이 시스템이 사용자가 쉽게 스스로 찾아낼 수 있는 비슷한 아이템들 뿐만 아니라, 사용자의 관심을 종합적으로 판단하여 기존 시스템과는 다른 새로운 아이템을 추천할 수 있음을 확인할 수 있었다. 또한, 사용자의 구매 성향 변화 및 시장의 변화에 따라 추천하는 아이템이 달라지는 기능을 실험 결과를 통해 확인할 수 있었다.

본 논문은 다음과 같이 구성된다. 2절에서는 데이터가 부족한 환경에서 추천 시스템을 적용할 때 발생할 수 있는 cold start 문제와 content-based filtering 관련 연구 및 사용자 이력을 윈도우를 사용하여 사용자 기록을 분류하는 관련 연구에 대해서 살펴본다. 3절에서는 전체 시스템 구조와 각 부분별 상세 작동 과정을, 4절에서는 실제 데이터를 사용하여 진행한 시스템 적용 및 실험에 대해서 알아본다. 마지막으로 5절에서 논문의 결론을 내린다.

## 2. 관련 연구

### 2.1 Cold Start 문제

Cold start 문제는 새 사용자나 물품이 들어왔을 경우, 이에 대한 데이터가 부족해서 발생하는 문제들을 말한다. Collaborative filtering에서는 사용자에게 추천할 물품을 예상하기 위해 사용자가 물품에 대한 점수를 매긴 것을 이용하여 그 사용자와 같은 혹은 유사한 물품에 비슷한 점수를 매긴 다른 사용자를 확인하고 비교함으로써 현재 대상 사용자의 구매 방향을 예측하고 추천한다. 이와 같은 방식으로 작동하기 때문에 사용자가 새로 가입을 하여 아직 아무 물건에도 점수를 매기지 않았거나, 혹은 새로운 아이템이 생겨서 그에 대한 다른 사용자들의 점수가 없을 경우에 이에 대한 비교가 이루어질 수 없어서 cold start 문제가 발생하게 된다. 새로운 시스템이 만들어졌을 때 또한 사용자 개체 부족으로 인해 비교를 할 수 없으므로 cold-start 문제가 발생한다[5]. 이를 극복하기 위해, 사용자의 소셜 네트워크 정보를 이용하여, 소셜 네트워크상에서 가까운 사용자와 비교를 하는 방법[4], 랜덤 워크를 통한 비교[7], CF와 content-based filtering을 합친 hybrid 접근방식[5, 6] 등의 방법이 새롭게 제시되고 있다. 하지만 아직까지는 제한된 해결책만 존재하며, 이들을 적용할 수 없는 경우 또한 다수 존재한다.

Content-based filtering의 경우도 이와 유사하게, 새로운

사용자가 생겼을 경우에는 사용자의 이력이 존재하지 않으므로 그 사용자에게 대한 성향 및 경향을 처리하지 못하게 된다. 따라서 이에 대한 해결책으로 사용자가 가입을 할 때, 관심 분야를 물어 이를 기반으로 가장 관심을 가질만한 물품을 추천하는 방식이 널리 사용되고 있다. 하지만 이러한 해결책들은 시간의 변화에 따른 사용자의 변화를 따라가기 어려운 실정이다.

### 2.2 User History Window

사용자의 이력을 사용할 때, 사용자 전체의 기록을 동일하게 보는 것이 아닌, 이력의 발생 시간 및 순서에 따라 분류하여 이를 사용자 선호도 변화 인식에 사용하는 연구가 있다[1]. 이 연구에서는 사용자가 직접 점수를 매기는 등의 노력이 없는 상황에서 사용자의 선호도 변화를 알기 위하여 현재 지점을 기준으로 얼마나 많이 떨어져 있는지를 확인한다. 최대 며칠까지의 기록을 사용할지를 고려하고, 시간별로 비중을 얼마나 둘지를 고려하여 이를 맞춤 추천을 위한 사용자 프로필을 작성하는 데 사용하였다. 이와 비슷하게 음악 추천 및 온라인 시장에서의 분석에서도 사용자 행동을 예측하는 데 window 및 가중 윈도우 개념이 많이 사용되며[9, 10], 시간 및 발생 횟수 등 다양한 기준으로 적용되고 있다.

앞의 연구들과 마찬가지로, 우리가 제안하는 시스템에서는 어떤 시간 혹은 구매 횟수 틀 안에서의 사용자의 구매 이력에 주목하고 시간의 변화에 따른 사용자 관심사 변화를 weighted window를 통해 찾아내고자 하였다.

## 3. User History Aware Recommendation System

사용자 이력을 충분히 반영하면서, 사용자의 변화에 맞추어 물품을 추천할 수 있는 시스템이 content-based filtering의 한계를 극복하기 위해서 요구되고 있다. 다른 논문에서도 언급되었던 것처럼[1], 사용자는 일정 시간이 지나거나 구매 횟수가 증가함에 따라 다른 단계의 사용자로 발전할 수 있다. 이 경우 사용자는 자신이 지금까지 구매한 물품과는 다른 물품에 흥미를 보이거나, 구매를 희망할 수 있다. 이외에도 사용자의 관심이 다른 분야로 옮겨갈 가능성도 있다. 또한 시간에 따른 시장의 트렌드 변화의 경우, 사용자가 주로 관심을 갖는 분야가 패션 등과 같이 트렌드에 영향을 많이 받는 분야라면 이는 더욱 변화가 클 것으로 예상된다. 따라서 이런 변화에 적응하여 사용자에게 의미 있는 물품을 추천할 수 있는 시스템이 필요하다.

Fig. 1은 제안된 시스템의 전체 구조를 보여준다. 그림에서 확인할 수 있듯이, 이 시스템은 크게 물품 유사성 그래프 생성 모듈과 사용자 이력 기반 랭킹 모듈의 두 가지 모듈로 나누어볼 수 있다. 첫 번째 모듈은 물품들 사이의 유사도를 구하는 모듈로서, 각각 물품들 사이의 모든 유사도를 구하여 그래프로 구성하는 모듈이다. 여러 feature

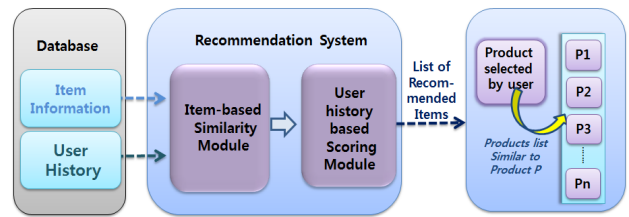


Fig. 1. System Structure

들을 적용하여 유사도를 조사하였으며, 여기에서 작성된 그래프는 두 번째 모듈에서 사용되게 된다. 두 번째 모듈은 작성된 물품들 사이의 유사도 그래프와 사용자 이력 및 윈도우를 사용하여 실제 추천되는 물품을 구하는 모듈이다. 사용자 이력에 있는 아이템들과 다른 물품들 사이의 관계를 확인하고, weighted window를 사용하여 모든 물품에 대해 추천 점수를 연산한 뒤 가장 높은 점수를 갖는 n개의 물품을 추천하게 된다.

### 3.1 Feature 기반 유사성 그래프 생성 모듈

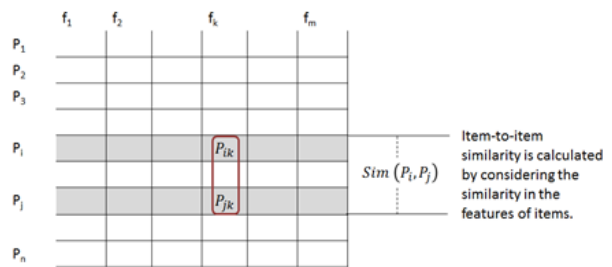


Fig. 2. Item-feature Matrix

물품들 사이에서의 유사성 그래프를 생성하기 위해서 weighted feature와 Jaccard 유사도가 사용되었다. 또한, 다양한 관점에서의 유사도를 얻기 위하여, 각 물품마다 그 물품의 중요한 속성을 대표하는 여러 feature들을 두고, 각 feature마다 중요도에 따라 서로 다른 weight를 두었으며, numeric, content-based, keyword-based feature들이 사용되었다. Keyword-based feature는 물품의 데이터에 존재하는 키워드를 바탕으로, content-based feature는 그 물품에 대한 설명을 바탕으로, numeric feature는 현재 그 물품이 속해있는 상위 카테고리 및 하위 카테고리를 표현하는 카테고리 번호를 바탕으로 계산되었으며, 각각의 feature들 weight에 따라 각 물품들 사이의 유사도에 적용되었다.

I를 m개의 물품들 집합, P<sub>i</sub>를 물품 i의 l개의 feature들의 집합이라고 했을 때,

$$I = \{P_1, P_2, \dots, P_m\} \tag{1}$$

$$P_i = \{f_1, f_2, \dots, f_l\} \tag{2}$$

$p_{ik}$ 가 물품  $p_i$ 에 대한 feature  $f_k$ ,  $A$ 가 feature들의 weight 들의 집합이고  $a_k$ 는  $f_k$ 에 대한 weight라고 하면,  $p_i$ 와  $p_j$ 의 Jaccard 유사도  $J_k(p_i, p_j)$ 는 다음과 같이 구할 수 있다.

$$J_k(p_i, p_j) = \frac{|P_{ik} \cap P_{jk}|}{|P_{ik} \cup P_{jk}|} \quad (3)$$

이를 이용하여 Fig. 2와 같이 물품  $p_i$ 와  $p_j$ 의 유사도  $Sim(p_i, p_j)$ 를 얻을 수 있으며,

$$Sim(p_i, p_j) = \sum_{k=1}^l a_k J_k(p_i, p_j) \quad (4)$$

Equation (4)를 모든 경우에 수행하여 전체 유사도 그래프를 얻을 수 있다.

### 3.2 사용자 이력 기반 랭킹 모듈

사용자는 물품을 구매할 때마다 이력을 남기게 되는데, 이 사용자의 구매 이력이 content-based 추천 시스템에서 물품을 추천하는 데 핵심 데이터가 된다. 사용자 개체마다 갖는 각각의 구매 이력을 바탕으로 사용자에게 맞는 물품을 추천하게 되는데, 이렇게 단순하게 이력을 이용하게 되면 몇 가지 문제점이 생길 수 있다.

첫째, 사용자가 오래전 구매했던, 즉 사용자가 현재 사용하지 않거나 구매하지 않는 물품까지도 추천에 반영되게 된다. 의류 및 가전제품을 예로 들면, 현재 시점으로부터 오래전에 유행했던 스타일의 의류나, 지금은 더 이상 사용되지 않는 전자제품이 추천에 반영될 수 있다. 따라서 이렇게 현 상황에 의미가 없는 정보는 제외하는 과정이 필요하다.

둘째, 사용자가 자주 찾던 항목만 추천하게 된다[8]. 사용자가 비슷한 물품을 여러 번 구매할 경우, 이 이력에 비슷한 항목의 물품들만 추천하게 되는데, 이러한 물품은 너무 당연하여 사용자가 이미 알고 있을 확률도 높으며 따라서 시스템의 목적 중 하나인, 사용자가 새롭게 관심을 가질만한 물품에 대한 추천이 제대로 이루어질 수 없다. 사용자에게 새롭지만 지금까지의 구매 이력과 연관이 있는 물품을 추천해줄 수 있는 방법이 요구된다.

두 과제를 해결하기 위해 우리는 weighted window를 사용자 이력에 적용하고, 새로운 유사도 점수 계산 방식을 적용하였다. 제안된 시스템에서는 사용자의 전체 구매 기록이 아닌, 부분의 기록을 사용하며, 각 물품마다 서로 다른 비중으로 추천에 반영되게 하였다.

#### 1) Weighted Window

제안하는 시스템의 초점은 사용자 전체 이력이 아닌 최근 및 최신 몇 개의 구매 기록을 반영하는 것이었으며, 이를 위해 window 개념을 가져와 최근 발생된  $n$ 개의 아이템들만 window  $W$ 안에 들어갈 수 있게 하였다.  $W$ 의 원소  $w_i$ 는 현재 시점에서  $i$ 번째 떨어져 있는 시점에서 구매한 물품을 의미한다.

$$W = \{w_1, w_2, \dots, w_n\} \quad (5)$$

$W$ 는 사용자 이력의 일부분을 추천 시스템에 반영할 수 있게 하며  $W$ 의 원소  $w_i$ 는 구매한 아이템을 의미한다. 이 중에서도 서로 차등 적용을 할 수 있게 원소들마다 구매 시기에 따라서 다른 비중을 갖게 하였다.  $R$ 은  $W$ 의 각 원소에 대응하는 weight의 집합이다.  $r_i$ 는  $w_i$ 의 weight이다.

$$R = \{r_1, r_2, \dots, r_n\} \quad (6)$$

$R$ 은 가장 최근의 기록이 추천에 가장 많이 반영될 수 있게 weight를 주었으며, linear, exponential, log 등의 형식을 사용하여 물품들 각각의 weight를 주게 된다. 이는 물품들이 어떤 영역에 속해있는지에 따라서 변할 수 있으며, 1로 normalize를 하였다.

$$\sum_{k=1}^n r_k = 1 \quad (7)$$

이 개념을 이용하여 오래된 사용자의 기록은 적용하지 않으면서, 적당한 시간 범위 혹은 순서 범위 안의 기록들만 차등 비중으로 적용할 수 있게 하였다. 이러한 weighted window를 적용하여 구성한 랭킹 모듈은 다음과 같다.

Fig. 3에서 보이는 것과 같이 이 모듈은 두 개의 단계로 나누어볼 수 있는데, 하나는 물품들 사이의 유사도 그래프와 사용자의 이력을 바탕으로 window  $W$ 를 생성하고,  $W$ 안의 물품 상황을 파악해  $W$ 의 weight인  $R$ 을 수정하는 단계이며, 다른 하나는  $W$ , 수정된  $R$ , 그리고 유사도 그래프를 사용하여 모든 아이템에 점수를 매겨 최종  $k$ 개의 아이템을 추천하는 단계이다.

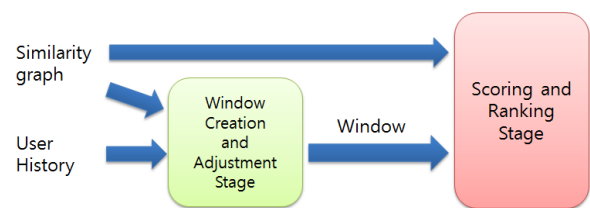


Fig. 3. Two Process Stages in Ranking Module

#### 2) Window 생성 및 조정 단계

사용자의 구매 기록을 확인하고 window를 적용하여  $W$ 를 생성한 후에는 기본적으로 설정된 weight  $R$ 을 현재  $W$  상황에 맞게 다시 조정하는 과정이 필요하다. 이 과정은 다음과 같은 상황을 대비하고, 좀 더 정확한 비중을 설정하게끔 처리하는 과정이다. 사용자가 물품을 구매할 때, 관심사가 새롭게 바뀌어 새로운 물품을 구매할 수도 있지만, 전혀 동떨어진 아이템을 단 한 번 구매하거나 한 번으로 충분한 구매 등 구매 패턴을 벗어나는 경우가 있을 수 있다. 이와 같은 단발성 거래가 최초로 시도되었을 때, 높은 비중을 가지

고 반영되게 되면, 결과적으로 사용자에게 맞지 않는 결과가 나올 수 있으므로, 이에 대응하기 위해 우리는 W안의 물품 구매 기록에 대하여 다음과 같은 과정을 통해 R을 재조정한다.

a) 구매 기록 분석

사용자 구매 기록 중 W안의 구매 기록을 분석한다. 앞서 생성된 아이템들 간의 유사도 그래프를 이용, W안의 한 아이템에 대해서 다른 W안의 모든 아이템들과의 유사도를 합한다. 이를 W안의 모든 아이템들에 의해 각각 수행한 뒤, 이 점수를 기반으로 하여 W안의 물품들 사이에서의 유사도 점수 랭킹을 정리한다.

b) Weight 수정

가장 유사도 점수가 낮은 물품의 weight에 대해서 일정 부분을 차감한 뒤, 그만큼의 weight를 가장 유사도 점수가 높은 물품의 weight에 더하여 사용자의 단발성 거래 물품에 대한 반영 비율을 감소시킨다. Algorithm 1은 이에 대한 계산을 통해 R를 수정하여 변경된 weight의 집합 R'을 생성한다.

**Algorithm 1** Weight adjustment algorithm

```

Input: W, R, AdjustmentFactor(af), Similarity graph
Output: R'
1: for(i=0; i<n/2; i++){
    //rank[i] = product id whose rank is i in W
2:   rrank [i] = rrank [i]+rrank [n-i]*af*(n/2-i)/(n/2);
3:   rrank [n-i-1] = rrank [n-i-1]-rrank [n-i]*af*(n/2-i)/(n-2);
4: }
    
```

위와 같이 window의 weight를 조정하게 되면, 다음의 두 가지 경우에 대응할 수 있다. 맨 처음에는 사용자의 관심사가 급격하게 옮겨간 경우를 반영하기 힘들지만, 그 항목에 대해 추가 구매가 발생할 경우에는 사용자가 그 항목에 대한 관심을 유지한다고 판단하고 이에 대해 더 높은 비중을 두어 추천을 생성할 수 있다. 또한, 단발성 거래가 최근에 발생했을 시, 높은 비중으로 반영되는 것에 대한 역제가 가능해진다.

3) Score and Ranking Stage

실제 아이템별 랭킹을 매기고 k개의 가장 점수가 높은 아이템을 추천하는 단계이다. 전 단계에서 생성된 W와 그 weight인 R 및 물품 유사성 그래프를 사용하여 각 물품별로 점수를 구하고, 이에 따라 랭킹을 매긴 뒤 추천을 생성하게 된다. P를 물품들의 집합, 전체 물품 수가 m, 전체 물품들에 대한 총 점수를 S라고 하면,

$$S = \{s_1, s_2, \dots, s_m\} \tag{8}$$

**Algorithm 2** Similarity scoring for every item

```

Input: W, R
Output: T(top k items)
1: Initialize S with 0;
2: for(i=0; i<n; i++){
3:   for(j=0; j<m; j++){
4:     if(i!=j && pj does not exist in W)
5:       sj=ri*ci,j+sj;
6:   }
7: }
8: Initialize T;
9: for(l=0; l<m; l++){
10:  for(i=0; i<k; i++){
11:    if(si> score of T[i]){
12:      push T down and place pl in T[i];
13:      sort T;
14:    }
15:  }
16: }
    
```

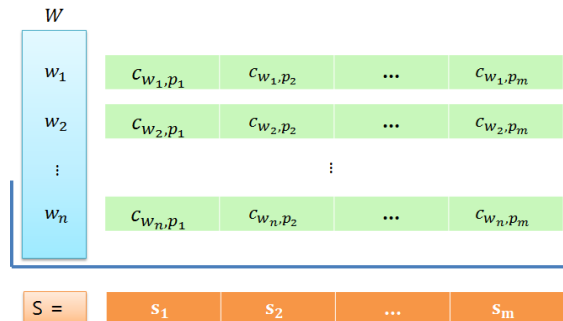


Fig. 4. Scoring Similarity between W and all other Items

물품  $p_u$ 와 물품  $p_v$ 에 대한 유사도값이  $c_{u,v}$ 라고 할 때, S는 다음 Algorithm 2와 같이 계산한다.

Fig. 4와 같이 W의 각 원소마다 다른 물품과의 유사성을 그래프를 통해 비교하게 되며, 이를 그대로 합산하지 않고, R과의 연산을 통해 각 원소별로 나온 유사성의 비중을 결정한다. S의 원소  $s_i$ , 즉  $p_i$ 의 추천 점수는 0부터 1 사이의 값을 가지게 되며, 이 점수의 비교를 통해 가장 높은 k개의 원소를 사용자에게 추천하게 된다.

Window안의 모든 기록들을 한꺼번에 적용하여 계산하는 것은 다음과 같은 이유 때문이다. 사용자가 물품을 구매하고 그 분야에 대해 어느 정도 알고 있을 경우, 비슷한 아이템을 추천하는 것은 사용자에게 의미가 크지 않다. 이미 알고 있는 아이템을 추천하는 경우일 확률이 크며, 이와 같은 경우를 줄이기 위해, 또한 새로운 물품을 추천하기 위해 이 시스템에서는 사용자의 최근 성향을 종합적으로 분석하는 방향으로 가게 되었다. Window 안의 물품들을 한꺼번에 적용하면 사용자가 지금까지 구매했던 물품들을 종합적으로 적용하여 그 물품들에 공통적으로 가까운 물품이 최종적으로 점수를 높게 받게 되므로, 비슷한 물품뿐만 아니라 새로운 물품을 추천받게 되는 경우가 생기게 되며, 이는 사용자가 잘 알지 못했던 물품일 확률이 크다. 따라서 사용자의 새로운 관심 및 연관 상품 추천에 기여를 할 수 있다.



### 4. 구현 및 실험

제시한 시스템과 기존 시스템과의 차이점 검증 및 적절한 추천이 이루어지는지 확인하기 위해, 몇 가지 경우에 대해 실험하였다. 신생 중소기업과의 협력을 통해 기업이 가진 총 313개의 아이템을 대상으로 하였으며, 이는 신생기업이라서 사용자 수가 부족하고 물품의 수가 많지 않아 제시된 시스템을 검증하기에 적합한 데이터라고 할 수 있다. 입력으로는 물품의 ID 및 물품이 속해있는 카테고리, 그리고 사용자 구매 이력을 받았으며, 윈도우의 기본 weight는 가장 최근의 기록이 가장 높은 값을 갖도록 하여 선형으로 분산하였다. 각 실험은 무작위의 경우, 무작위 사용자 기록 및 입력 생성을 통해 생성하였고, 패턴이 있는 입력 및 사용자 기록의 경우, 각 제품 클러스터별로 구분하여 여러 조합을 생성하였다. 각 실험은 1000회 수행 후 평균을 내었다.

#### 1) Window 영향 실험

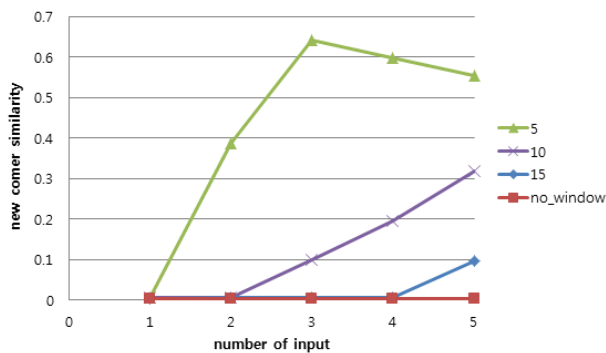


Fig. 5. Similarity between New Input Item and Recommended Items under Patterned History, Patterned Input

Window의 영향을 확인하기 위하여, window를 적용한 경우와 적용하지 않은 경우를 실험하였다. 과거 구매 이력이 일정한 패턴, 즉 비슷한 물건들을 구매하던 상황에서 그와는 다른 물품들을 새로 구매했을 경우, 새로운 구매 이력이 추천된 물품에 미치는 영향을 확인하기 위해서 새로운 물품과 추천된 물품과의 유사도를 확인하였다(이하 new comer similarity). Fig. 5는 이에 대한 실험 결과를 보여주며 윈도우 크기 및 윈도우 적용 여부에 따른 실험을 진행하였다. Window를 적용하지 않고 전체 기록을 사용할 경우에는 새로 들어온 물품들이 추천에 미치는 영향이 거의 없음을 알 수 있다. Window를 적용하였을 경우, 그 크기가 5이면, 하나의 물품이 들어왔을 경우에는 전혀 영향을 미치지 못하지만 2번째 입력부터 높은 비율로 반영되기 시작하는 것을 볼 수 있다. 윈도우 크기가 10, 15인 경우와 비교를 해볼 때, 윈도우 크기가 커질수록 새로운 물품들이 추천에 반영되기 시작하는 입력 수가 커지는 것을 확인할 수 있으며, 이는 전체 window 대비 새로운 물품들 비율이 작아지게 되므로 발생한다.

#### 2) Weight Adjustment 효과 검증

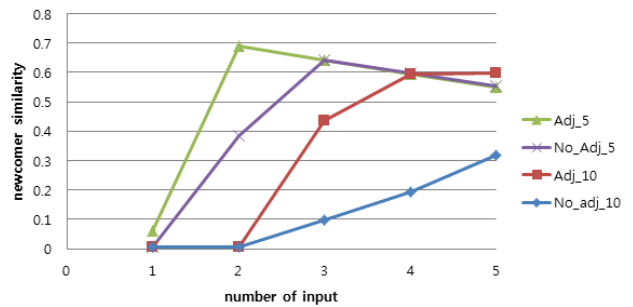


Fig. 6. Patterned-history, Patterned Input with and Without Weight Adjustment Stage

사용자의 구매 성향 분석 및 단발성 거래를 억제하기 위한 단계인 weight adjustment 단계의 효과를 알아보기 위해 Fig. 6에서와 같이, 사용자가 패턴이 있는 기록을 가지고 있는 상황에서 다른 패턴의 구매를 시작할 경우, 이 단계의 적용 유무에 따른 결과를 비교하여 새로 들어온 물품의 영향력 차이를 알아보았다. 사용자가 패턴이 있는 기록을 가지고 있는 상황에서 첫 번째 입력이 발생했을 시, 모든 경우는 비슷한 결과를 보여주지만 두 번째 입력이 발생했을 경우는 조금 달라진다. 윈도우 크기가 5일 때, 두 번째 입력이 발생하면, weight 수정 단계가 있는 경우는 이를 사용자 패턴이 달라졌다고 인식하여, 빠르게 그 패턴에 속해있는 물품들이 추천에 반영되는 비율을 높인다. 조정 단계가 있는 경우와 없는 경우, 초반 그래프 기울기에서 큰 차이를 확인할 수 있다. 윈도우 사이즈가 커질 경우, 3번째 입력부터 새로운 패턴을 인식하기 시작하며, 이 경우 또한 weight 수정 단계가 있는 경우가 적용 비율이 가파르게 증가하는 것을 확인할 수 있다. 이는 사용자의 구매 성향 및 패턴 변화를 인식하고, 이를 빠르게 추천 아이템에 적용하는 것으로 볼 수 있다.

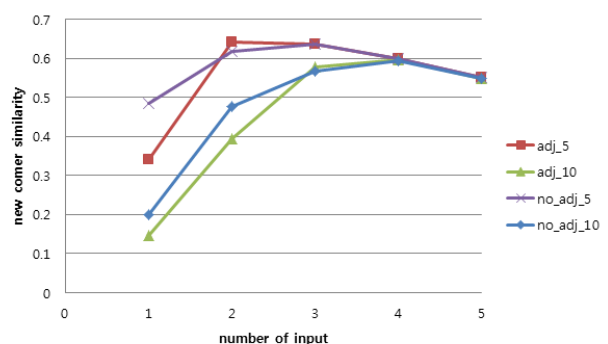


Fig. 7. Random-history, Random-input with and Without Weight Adjustment Stage

Fig. 7에서와 같이 패턴이 없는 사용자 기록에 패턴이 있는 물품들이 들어오게 되면 다음을 확인할 수 있다. Window 크기가 5인 경우, 처음 1개의 입력과 그 추천 반영

를 확인하면 weight 수정 단계가 적용되지 않은 경우가 반영률이 높지만 2번째 입력부터는 적용된 경우가 훨씬 가파르게 증가하는 것을 관찰할 수 있다. 이는 처음 들어온 입력은 단발성 구매로 파악하여 반영률이 낮지만 2번째 입력부터는 물품 구매 성향으로 판단하여 그 반영률을 늘리기 때문이다. 이를 통해 사용자 구매 성향에 변화가 있을 시 빠르게 대응을 할 수 있으며, 구매 성향과 동떨어진 단발성 구매가 있을 경우, 추천에 작은 영향으로 반영하는 것이 가능해진다. 윈도우 크기가 클 경우는 반영할 때까지 필요한 구매 횟수가 더 크지만 반영된 후부터는 비슷한 결과를 보여준다.

3) 다른 분야에 속한 아이템들에 대한 추천 실험

서로 거리가 먼 여러 물품들이 window에 들어있을 때, 들어있는 물품에 대한 각각의 추천이 아닌 그 물품들과 공통적으로 거리가 가까운 몇 개의 물품이 추천됨을 여러 번의 실험을 통해 확인할 수 있었고, 몇몇은 카테고리가 전혀 다른 물품인데도 추천됨을 확인할 수 있었으며, 직관적인 연관성을 확인할 수 있었다.

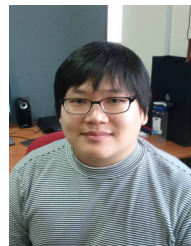
5. 결 론

기존의 content-based filtering의 추천 시스템은 적절한 추천을 방해하는 몇몇 한계점이 있었다. 비슷한 아이템만 추천하는 것과, 시간에 따라서 변하는 유저의 구매 성향 및 시장변화에 적응할 수 없는 것, 이들 문제를 완화하기 위해 우리는 사용자 기록에 weighted window를 적용하여 새로운 아이템 추천 및 사용자의 현재 상태에 맞는 추천 시스템을 개발, 진행하였다. 이 시스템은 사용자 개개인에 맞추어 개인의 변화 혹은 트렌드의 변화에 민감하게 반응할 수 있어 더욱 정확한 추천을 할 수 있을 것으로 기대되며, 사용자 패턴을 확인하여 필요 없는 물품이 추천되는 것을 억제한다. 또한, 전혀 다른 물품이지만 사용자가 관심을 가질만한 물품에 대한 추천도 가능하여, 사용자의 새로운 관심을 자극할 수 있을 것으로 기대된다.

References

[1] K. Sugiyama, K. Hatano, and M. Yoshikawa, "Adaptive web search based on user profile constructed without any effort from users," in *Proceedings of the 13th international conference on World Wide Web*, pp.675-684, 2004.  
 [2] X. Su, T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," in *Advances in Artificial Intelligence*, Vol.2009, Article No.4, 2009.  
 [3] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, "Addressing cold-start problem in recommendation systems," in

*Proceedings of the 2<sup>nd</sup> international conference on Ubiquitous information management and communication*, pp.208-211, 2008.  
 [4] I. Konstas, V. Stathopoulos, and J. M. Rose, "On social networks and collaborative recommendation," in *Proceedings of the 32<sup>nd</sup> international ACM SIGIR conference on Research and development in information retrieval*, pp.195-202, 2009.  
 [5] J. Basilico, T. Hofmann, "Unifying collaborative and content-based filtering," in *Proceedings of the 21st international conference on Machine learning*, 2004.  
 [6] M. Balabanovic, Y. Shoham, "Fab: content-based, collaborative recommendation," in *Communications of the ACM*, Vol.40, pp.66-72, 1997.  
 [7] H. Yildirim, M. S. Krishnamoorthy, "A random walk method for alleviating the sparsity problem in collaborative filtering," in *Proceedings of the ACM conference on Recommender systems*, pp.131-138, 2008.  
 [8] F. Ricci, L. Rokach, B. Shapira, and P. Kantor, "Recommender systems handbook," Springer, 2011.  
 [9] N. Hariri, B. Mobasher, and R. Burke, "Context-aware music recommendation based on latent topic sequential patterns," in *Proceedings of the sixth ACM conference on Recommender systems*, pp.131-138, 2012.  
 [10] J. Wang, B. Sarwar, and N. Sundaresan, "Utilizing related products for post-purchase recommendation in e-commerce," in *Proceedings of the fifth ACM conference on Recommender systems*, pp.329-332, 2011.



황 성 민

e-mail : sungmin1511@gmail.com  
 2013년 홍익대학교 정보컴퓨터공학부(학사)  
 2013년~현 재 전남대학교 전자컴퓨터공학부 석사과정  
 관심분야 : Social network, data analysis, recommender system



Rajashree Sokasane

e-mail : sokasaners@gmail.com  
 2005년 Shivaji University, Department of Electronics(학사)  
 2007년 Shivaji University, Department of Computer Science(석사)  
 2013년~현 재 전남대학교 전자컴퓨터공학부 박사과정  
 관심분야 : Distributed systems, location based applications



### Hiep Tuan Nguyen Tri

e-mail : tuanheip1232@gmail.com  
2010년 Hanoi University of Science  
and Technology, Electronics  
and Telecommunications(학사)  
2013년~현 재 전남대학교 전자컴퓨터공학부  
석사과정  
관심분야: Software Defined Network,  
big data



### 김 경 백

e-mail : kyungbaekkim@jnu.ac.kr  
1999년 한국과학기술원 전자전산(학사)  
2001년 한국과학기술원 전자전산(석사)  
2007년 한국과학기술원 전자전산(박사)  
2007년~2011년 University of California,  
Irvine, 박사후연구원  
2012년~현 재 전남대학교 전자컴퓨터공학부  
조교수

관심분야: 분산시스템, 미들웨어, 피어투피어 네트워크, 오버레이 네트워크, 소셜 네트워크, 모바일 클라우드 시스템