

# 사용자 평점 기반 게임 추천 시스템

## (Game Recommendation System Based on User Ratings)

김 종 현<sup>1)</sup>, 조 현 정<sup>2)</sup>, 김 병 만<sup>3)\*</sup>

(JongHyen Kim, HyeonJeong Jo, and Byeong Man Kim)

**요 약** 최근 게임 산업의 발달과 게임 방송에 대한 사람들의 관심이 많아짐에 따라 기존 게이머들이 아닌 사람들도 게임에 관심을 많이 보이고 있고, 게임 구매로 이어지고 있다. 하지만, 일반 사용자가 매일 수십 개씩 발매되는 게임 중에 어떤 게임이 자신이 재밌게 즐길 수 있는 게임인지를 판단하기 어렵다. 따라서 게임 판매 플랫폼에서 게임 추천 기능을 갖추고 있지만 그들의 매출 증가를 위한 수단으로 사용되어 그들의 할인 제품이나 신제품에 초점을 맞춰 추천을 해주기 때문에 추천 시스템의 정확도가 낮다. 이러한 이유 때문에 본 논문에서는 사용자에게 추천 만족도를 높이고 사용자 경험을 적절히 반영한, 사용자가 남긴 평점을 기반으로 한 게임 추천 시스템을 구성하였다. 시스템에서는 협력 필터링을 이용한 예상 평가 점수 기능과 나이브 베이지안을 이용한 게임 추천 기능을 구현하여 사용자에게 빠르고 정확한 추천을 할 수 있도록 구현하였다. 결과적으로 예상 평점 알고리즘의 경우 2.4초의 처리 속도와 평균 72.1퍼센트의 정확도를 얻었고, 게임 추천 알고리즘의 경우 75.187퍼센트의 정확도를 얻어 사용자에게 빠르고 정확한 추천 결과를 제시 할 수 있었다.

**핵심주제어** : 게임 추천, 협력 필터링, 나이브 베이지안 분류기

**Abstract** As the recent developments in the game industry and people's interest in game streaming become more popular, non-professional gamers are also interested in games and buying them. However, it is difficult to judge which game is the most enjoyable among the games released in dozens every day. Although the game sales platform is equipped with the game recommendation function, it is not accurate because it is used as a means of increasing their sales and recommending users with a focus on their discount products or new products. For this reason, in this paper, we propose a game recommendation system based on the users ratings, which raises the recommendation satisfaction level of users and appropriately reflect their experience. In the system, we implement the rate prediction function using collaborative filtering and the game recommendation function using Naive Bayesian classifier to provide users with quick and accurate recommendations. As the result, the rate prediction algorithm achieved a

\* Corresponding Author : bmkim@kumoh.ac.kr

+ 이 논문은 2018년 금오공과대학교 연구비 지원에 의해 연구되었음.

Manuscript received October 4, 2018 / revised October 26, 2018 / accepted November 6, 2018

1) 금오공과대학교 컴퓨터소프트웨어공학과, 제1저자  
2) 금오공과대학교 컴퓨터소프트웨어공학과, 제2저자  
3) 금오공과대학교 컴퓨터소프트웨어공학과, 교신저자

throughput of 2.4 seconds and an average of 72.1 percent accuracy. For the game recommendation algorithm, we obtained 75.187 percent accuracy and were able to provide users with fast and accurate recommendations.

**Key Words** : Game Recommendation, Collaborative Filtering, Naive Bayesian Classifier

## 1. 서론

게임 산업의 발달과 게임 방송에 대한 사람들의 관심이 많아짐에 따라 게임을 이용하는 사람들의 특징이 다양해지고 있다. 이에 따라 게임 개발사들은 다양해진 이용자들을 위해 새로운 게임들을 출시하고 있고, 인디 게임 개발사들도 참신한 아이디어를 가지고 게임을 출시하고 있어, 하루에도 수십 개의 게임이 발매되고 있다. 이로 인해, 게임을 구매하고자 하는 사람들은 자신이 구매하려고 하는 게임이 정말 자신이 재밌게 즐길 수 있는 게임인지 알기 위해 게임 전문 매체의 리뷰기사나 유튜브 등과 같은 곳에 업로드 되는 유명 방송인의 리뷰 영상, 혹은 게임 판매 플랫폼에 등록되는 리뷰 등에 의존한다. 이러한 방법들은 구매자의 취향을 반영하기 어려울 뿐만 아니라 게임사에서 광고를 목적으로 리뷰를 게재했다면, 구매자는 정확한 판단하기 어렵다는 단점이 있다.

최근 게임 판매 플랫폼에서 이용자의 게임 구매 목록과 최근 실행한 게임 등을 기반으로 게임 추천을 하는 기능을 탑재하고 있다. 하지만, 이 역시 그들의 매출 증대를 위한 수단으로 사용되고 있어 할인 제품이나, 신제품을 우선적으로 보여주는 등 사용자에게 만족할 만한 기능이 제공되지 않는다. 이렇기 때문에 사용자의 데이터를 기반으로 한 게임 추천 시스템이 요구된다.

본 논문에서는 사용자가 입력한 게임 평가 정보를 기반으로 협력 필터링 [1, 2, 3]을 통해 사용자의 각 게임에 대한 예상 평점을 계산하여, 사용자가 해당 게임을 구매하여 플레이 했을 때 얻을 수 있는 만족도를 수치화 하여 보여주고, 나이브 베이저안 분류기 [4]를 통해 게임 평가

정보를 게임의 장르별 선호도로 계산하여, 사용자의 장르별 취향을 분석할 수 있도록 하였다. 또한, 협력 필터링이 가지고 있는 여러 문제점 중 콜드 스타트, 확장성 문제를 완화 할 수 있도록 방안을 제시하였다. 콜드 스타트 문제를 완화하기 위해 초기 데이터 수집 방안을 수립하여 시스템과 사용자에게 일정 이상의 데이터를 가지고 있을 수 있도록 하였다. 또, 유사도 계산 대상을 평가한 게임이 많이 중복되는 순으로 선정하여 유사도 계산 대상을 대폭 감소하여 확장성 문제를 완화할 수 있었다.

## 2. 관련 연구

### 2.1 기존 추천 알고리즘

추천 시스템을 구현하는 방식에는 크게 두 가지 방식이 존재하는데, 하나는 협력 필터링 방식 [1,2,3]이고, 다른 하나는 내용 기반 필터링 방식[1,2,3]이다. 내용 기반 필터링 방식은 찾고자 하는 정보의 내용과 사용자의 관심사를 비교하여 사용자에게 적절한 정보를 선택해 주는 방식이다. 내용 기반 필터링 방식을 이용하기 위해선 사용자 관심사가 반드시 텍스트 등과 같은 기계가 분류 할 수 있는 형태이어야 하고, 사용자 관심사와 찾고자 하는 대상의 내용이 정확하게 일치해야 하는 용어 불일치 문제가 있다.

협력 필터링 방식은 사용자의 취향을 예측하기 위해 다른 사용자들의 의견을 사용하는 방식으로 세계 최대 온라인 쇼핑몰인 Amazon에서도 사용자의 구매 목록과 검색 기록을 바탕으로 한 협력 필터링 방식을 채용 [5]하고 있으며, 영화 스트리밍 서비스를 하는 Netflix에서는

Netflix Prize라는 이름으로 추천 알고리즘을 공모하였고, 협력 필터링을 기반으로 한 알고리즘이 우수한 성과를 얻어 Netflix에서 사용하고 있다 [6]. 하지만, 협력 필터링 방식에는 몇 가지 문제점을 안고 있는데, 콜드 스타트 문제(Cold Start Problem), 희소성 문제(Sparsity Problem), 확장성(Scalability)이다.

콜드 스타트 문제는 새로운 게임이나 새로운 사용자가 시스템에 등록되었을 때, 게임에 사용자들의 평가가 등록되지 않는 경우나 사용자가 게임에 평가를 등록하지 않은 경우 예측 점수에 기반이 되는 평점이 없으므로 예측 점수를 계산할 수 없다는 문제점이다.

희소성 문제는 지금까지 발매된 게임의 개수가 일반적인 사용자가 소화 할 수 있는 개수보다 훨씬 상회해서 게임-사용자 행렬이 매우 희소한 분포성을 띄게 되는 문제로, 이는 유사도 계산 시에 문제를 유발 할 수 있으며, 정확도를 떨어뜨리게 된다. 또한 일반적으로 인기 게임과 비인기 게임간의 평가 개수도 차이가 크게 날 수 밖에 없는데, 이로 인해 비인기 게임이 인기 게임에 비해 희소성 문제에 처할 가능성이 높다.

협력 필터링 방식에서는 보통 최근접 이웃 알고리즘을 사용하게 되며, 이로 인해 사용자와 게임의 수가 증가하면 할수록 비교 대상이 늘어 계산 시간이 비례하여 증가하게 된다. 이러한 문제가 확장성의 문제이다.

## 2.2 메타 크리틱(Metacritic)

메타 크리틱[7]은 영화, 게임, TV 프로그램 등 멀티미디어 전반의 리뷰를 모아놓은 웹 서비스이다. 메타 크리틱은 어떤 아이টে에 전문 매체의 리뷰가 등록이 되면 리뷰어가 부여한 해당 아이টে의 점수를 메타 크리틱에서 수집하여 평균을 사용자에게 Fig. 1처럼 보여준다. 메타 크리틱에서는 그것을 크리틱 점수라 부르는데, 많은 사람들이 객관성과 전문성을 갖추고 있다고 생각하고 있다. 비슷한 예로 영화에서는 일반적으로 로튼 토마토 점수가 높으면 좋은 영화라고 사람들이 생각하는데, 게임에서는 크리틱 점수

가 그것에 대응된다고 볼 수 있다. 때문에 게임 판매 플랫폼들은 자신들이 가지고 있는 게임에 메타 크리틱 점수를 게재하고 있으며, 사용자들은 그것을 보고 게임의 재미를 어느 정도 예상할 수 있다.

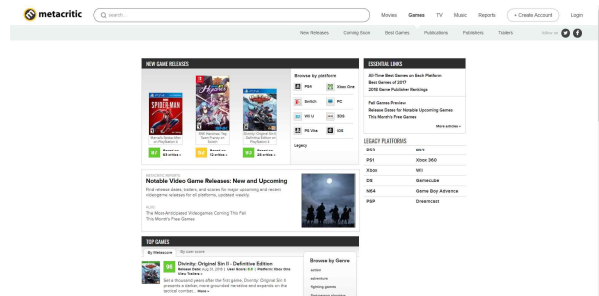


Fig. 1 Metacritic's Web Page

## 2.3 기존 게임 추천 시스템

많은 게임 판매 플랫폼들이 자신들의 매출 상승을 위해 사용자들에게 게임 추천을 하려는 시도가 보이고 있다. 그 중 가장 활발하게 게임 추천을 하려는 곳은 스팀(Steam)[8]이라는 게임 판매 플랫폼으로 전세계 최대 규모를 가지고 있다. 스팀에서는 최근 게임 큐레이터라는 서비스를 시작으로 사용자의 친구가 구매한 게임 목록, 전문성있는 게임 큐레이터의 평가, 사용자가 최근 많이 플레이한 게임 등을 바탕으로 게임을 추천하고 있다. 하지만, 많은 사용자들이 스팀의 추천 시스템에 불만을 가지고 있다. 그 이유는 Fig. 2에서 처럼 스팀의 추천은 정말 사용자가 즐겁게 플레이 할만한 게임을 추천해주는 것보다 자신들이 현재 진행하고 있는 할인 품목들을 대상으로 위에서 언급한 데이터들을 사용하고 있기 때문이다. 이로 인해, 사용자들은 자신이 만족 할만한 게임 보다는 스팀에서 어떤 게임이 할인하고 있는지를 더 많이 보게 되는 것이다.

본 논문에서는 기존의 게임 추천 시스템이 가지는 이러한 문제점들을 해결하기 위해 공정한 분석을 통해 도출된 추천 결과를 사용자들에게 제시함으로써 더 나은 유저 경험을 가질 수 있도록 하였다.

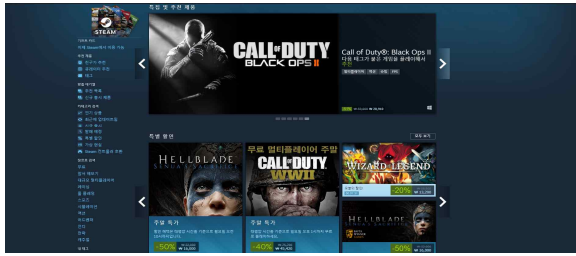


Fig. 2 Steam's Web Page

### 3. 사용자 평점 분석을 통한 게임 추천 시스템

#### 3.1 시스템 구조

Fig. 3은 본 논문에서 제안하는 시스템의 운용도이다. 시스템은 데스크톱과 모바일에서 동일한 사용자 경험을 제공하기 위해서 반응형 웹 디자인으로 만들어진 웹 페이지로 인터페이스를 구축하였다. 본 시스템은 Angular를 사용하여 차세대 웹 개발 방식인 SPA(Single Page Application)로 제작하였다. 또, Node.js와 Express를 이용하여 RESTful API를 작성하여 웹 페이지의 새로 고침 없이 페이지에 데이터를 반영할 수 있도록 제작하였고, 이에 따라 사용자에게 끊임 없는 웹 서비스를 제공하여 보다 빠르고 쾌적한 서비스를 제공할 수 있게 하였다. 또한, 알고리즘은 자바스크립트로 작성되어 Node.js에서 실행이 되는데 Node.js는 싱글스레드로 수행이 되므로 계산의 속도를 더욱 높이기 위해 비동기 데이터 처리 라이브러리인 Rxjs를 이용하여 멀티 스레드로 구현된 것 같은 효과를 얻을 수 있다[9].

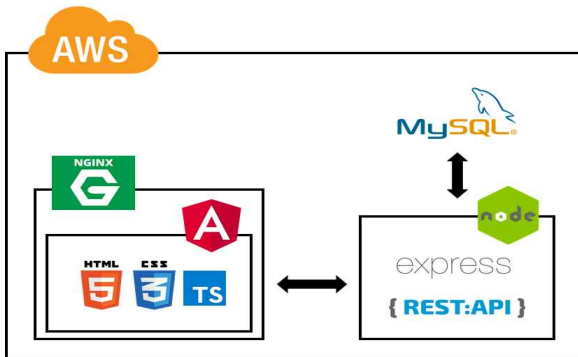


Fig. 3 Diagram of System Operation

Fig. 4는 본 논문에서 제안하는 시스템의 주요 기능에 대한 구조도이다. 사용자가 입력한 평점은 데이터베이스에 저장되었다가 사용자가 특정 게임의 예상 점수를 원하면 데이터베이스에서 전처리과정을 통해 이웃을 선정하고 선정된 이웃과의 유사도 점수를 계산한 뒤, 예상 점수를 산출하게 된다. 또한, 게임 추천은 사용자가 등록한 게임 평가 정보를 이용해 장르 선호도를 계산한다. 그리고 나서, 선호 장르에 속한 게임을 나이브 베이지안을 통해 선호 게임인지를 판별한 뒤, 사용자에게 만들어진 선호 게임 리스트를 제시한다.

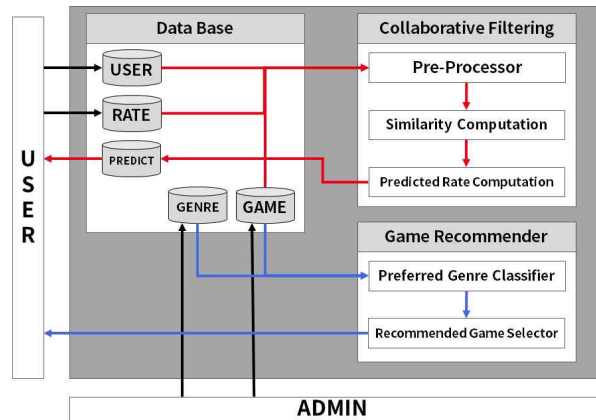


Fig. 4 Diagram of System Structure

#### 3.2 초기 데이터 수집

초기 데이터 수집은 협력 필터링의 문제점 중 하나였던 콜드 스타트 문제를 해소하기 위해 실행하였다. 초기 데이터 수집은 게임과 사용자 두 가지로 나뉘어서 진행하였다. 먼저 게임에 대한 데이터 수집은 해외 게임 전문 웹 서비스인 Metacritic의 평점 데이터를 수집 하였다. 게임 정보는 3천 여 개, 사용자 정보는 10만 5천여 개, 게임 평가 정보는 26만 2천여 개, 장르정보를 140여 개를 수집하여 데이터베이스에 저장하였다.

또, 새로 추가되는 사용자의 경우에는 회원 가입 시 사전에 데이터 수집을 실시하게 되는데, 먼저, 평소 선호하는 게임의 장르를 선택하게 되면 시스템에서 해당 장르의 게임을 선별하

여 사용자에게 보여주고, 사용자는 리스트에서 실제 플레이 해본 게임에 대해 평가를 하게 된다. [5]에서는 사용자가 20개 이상 평점 정보를 입력했을 때, 정확한 유사도 측정이 가능하다고 분석했기 때문에, 시스템에서도 해당 개수 이상의 평점 정보를 입력 했을 시에 시스템에서 예상 점수를 계산 하도록 하였다.

### 3.3 예상 점수 측정

예상 점수 측정은 협력 필터링 방식을 이용한다. 협력 필터링 방식은 이웃이 입력한 게임 평가 정보와 사용자가 입력한 게임 평가 정보에 대한 유사도를 계산하고, 계산된 유사도를 이용해 이웃의 평가 특징을 통해 사용자의 예상 점수를 얻는다.

#### 3.3.1 유사도 계산

유사도를 계산하는 방식으로 일반적으로 피어슨 상관관계 공식[10]과 보완 코사인 유사도 측정 방법[11]이 사용된다. 두 방법은 사용자의 평가 성향을 파악하여 그 유사성을 비교하는데 주로 사용된다. 때문에, A라는 사용자가 대부분의 영화에 대해 1 ~ 4점만을 주는 성향이 있고, B라는 사용자가 대부분의 영화에 대해 2 ~ 5점만을 주는 성향이 있을 때, 두 사용자가 같은 영화에 대해 상대적으로 비슷한 평가를 주었다면, 위의 방법을 통해 유사도를 계산한다면 A, B는 비슷한 유사도를 가지게 된다.

두 식의 차이점은 피어슨 상관관계 공식은 사용자가 평가한 평균을 통해 사용자의 평가 성향을 파악하고, 코사인 유사도 공식은 게임에 평가된 평균을 통해 사용자의 평가 성향을 파악한다. [2]의 결과를 보았을 때 피어슨 상관관계 공식과 보완 코사인 공식의 결과 값은 비슷하였기 때문에 본 논문에서는 보완 코사인 유사도를 채택하였다.

$$sim(k, l) = \frac{\sum_{i=1}^m (k_i - \bar{r}_i)(l_i - \bar{r}_i)}{\sqrt{\sum_{i=1}^m (k_i - \bar{r}_i)^2} \sqrt{\sum_{i=1}^m (l_i - \bar{r}_i)^2}} \quad (1)$$

여기서,  $sim(k, l)$ 은 사용자  $k$ 와  $l$ 간의 유사도를,  $k_i$ 는 게임  $i$ 에 대한  $k$ 의 평점을,  $l_i$ 는 게임  $i$ 에 대한  $l$ 의 평점을,  $r_i$ 는  $i$ 의 모든 사용자에게 대한 평점 평균을,  $m$ 은  $k$ 와  $l$ 이 동시에 평가한 게임들을 뜻한다.

[2]에서는 대상 게임에 대한 사용자의 평가 예측을 위한 이웃을 선정할 때 모든 사용자에게 대한 유사도를 구한 뒤 top-N을 선발해 평가 예측에 사용하였다. 이는 앞서 말한 확장성 문제에 큰 영향을 주는 문제였고, 26만개가 넘는 평가 정보를 활용하여 평가 예측을 할 때 계산 시간에 너무 많은 시간이 소요된다. [12, 13]에서는 협력 필터링에서 유사도를 사용할 때 이웃과 사용자가 중복되는 평가 게임이 많을수록 가중치를 더 많이 부여하는 자카드 계수 유전 알고리즘을 통해 구하는 방식을 제안하였는데, 본 논문에서는 여기서 아이디어를 얻어 top-N을 선발하는 과정에서 평가한 게임이 많이 중복되면 유사도에 대한 신뢰도가 높다고 판단하여 중복되는 평가 게임이 많은 N명을 선발 하여 유사도를 계산하였다.

위의 방법은 사용자 수에 대한 확장성을 높이기 위한 방법이다. 하지만, 예상 점수 측정 시간에 많은 영향을 주는 요소에는 평가된 게임 수도 있기 때문에 그것에 대한 확장성도 고려를 해야한다. 본 논문에서는 예상 점수 측정에 사용되는 대상 사용자의 평가 데이터에 기간 제한을 두어 평가된 게임 수에 대한 확장성을 고려했을 뿐만 아니라, 사용자의 최신 취향에 대한 정보를 더 정확하게 파악할 수 있게 하였다.

#### 3.3.2 협력 예측

사용자  $k$ 에 대해 게임  $i$ 에 대한 예상 평점을 구하기 위해서 미네소타 대학에서 개발한 Group Lens 방식을 이용했다[2]. (1)에서 얻은  $sim(k, l)$ 와  $l$ 의 평가 성향을 이용해  $k$ 의 예상 점수를 도출해낸다.

$$P_{k,i} = \bar{R}_k + \frac{\sum_{l=1}^n (R_{l,i} - \bar{R}_i) \times sim(k,l)}{\sum_{l=1}^n |sim(k,l)|} \quad (2)$$

여기서,  $P_{k,i}$ 는 게임  $i$ 에 대한 사용자  $k$ 의 예상 점수이다.  $R_k$ 는 사용자  $k$ 의 평균 평점을,  $n$ 은 사용자  $k$ 에 대해 선정된 이웃들을,  $R_{l,i}$ 는 게임  $i$ 에 대해 사용자  $l$ 의 평점을 나타내고  $R_l$ 은  $l$ 의 평균 평점을 나타낸다.  $\bar{R}_i$ 는 게임  $i$ 에 대한 모든 사용자에게 대한 평균 평점을 나타낸다.

### 3.4 게임 추천

게임 추천은 사용자가 입력한 평가 정보와 평가가 입력된 게임들의 장르 정보를 이용하여 사용자가 선호하는 장르 정보를 선별하여 선별된 장르에서 게임을 추천 해주는 방식으로 진행된다[14]. 게임 추천의 경우 협력 필터링을 이용해 전체 항목을 계산하여 사용자에게 예상 점수가 높은 게임을 추천 해주는 방법도 있으나, 이 방법은 빠른 응답속도를 보여야 하는 웹 서비스 특성상 맞지 않다고 판단하여, 본 논문에서는 빠른 시간에 많은 게임을 추천 할 수 있는 나이브 베이지안 분류식을 선택하였다.

나이브 베이지안 분류식(이하 나이브 베이지안)은 사용자가 입력한 평가 정보를 이용하여 사용자 개인의 장르 선호도로 계산하는 역할을 하게 된다. 나이브 베이지안은 대상에 속한 특성들이 서로 독립적이란 가정 하에 각각의 특성들을 이용해 해당 대상이 분류 기준에 해당할 확률을 계산할 수 있다. 본 논문에서는 게임에 속한 장르 정보를 이용하여, 사용자가 각 게임에 대해 선호할 게임일지, 비선호 게임일지를 분류한다.

Fig. 5는 나이브 베이지안을 이용한 게임추천 과정을 도식화 한 것이다. 나이브 베이지안은 하나의 게임에 대해 선호, 비선호 구분을 하기 때문에, 전처리 과정을 통해 사용자에게 제공할 게임들을 선별한 뒤, 나이브 베이지안을 각 게임에 적용하여 선호 게임으로 구분된 게임들만

사용자에게 보여진다. 전처리 과정은 먼저 대상 사용자가 입력한 평가 정보를 [4]에서 가정한 것처럼 게임에 내린 점수만큼 게임에 속한 장르 정보에 동일하게 분배한다. 그런 후 각 장르 정보를 평균 평점 순으로 정렬하고 상위 장르에서 전체 사용자에게 대한 평균 평점이 높은 게임들을 선별한다. 이후, 해당 게임들에 나이브 베이지안을 적용하여, 선호 게임으로 구분된 게임들을 사용자에게 보여준다.

$$P(L_{like}|G_x) = \left( \prod_{k=1}^n P(T_k|L_{like}) \right) P(L_{like}) \quad (3)$$

$$P(L_{dislike}|G_x) = \left( \prod_{k=1}^n P(T_k|L_{dislike}) \right) P(L_{dislike})$$

$$P(L_{like}|G_x) > P(L_{dislike}|G_x) ? 'like' : 'dislike' \quad (4)$$

여기서,  $P(L_{like}|G_x)$ 는 게임  $G_x$ 가 선호 ( $L_{like}$ ) 클래스에 속할 확률을,  $P(L_{dislike}|G_x)$ 는 비선호 ( $L_{dislike}$ ) 클래스에 속할 확률을,  $P(T_k|L_{like})$ 는 선호 게임 중에 장르가  $T_k$ 일 확률을,  $P(T_k|L_{dislike})$ 는 비선호 게임 중에 장르가  $T_k$ 일 확률을 뜻한다.  $P(L_{like})$ 는 사용자의 평가 정보에서 선호 확률을,  $P(L_{dislike})$ 는 사용자의 평가 정보에서 비선호 확률을 뜻한다. 하나의 게임에 대해 선호 클래스에 속할 확률과 비선호 클래스에 속할 확률을 계산하여 더 높은 쪽을 택한다. 선호인 기준은 사용자의 평균 평점을 기준으로 한다. 즉 사용자 평점보다 높은 경우를 선호로 판단한다.

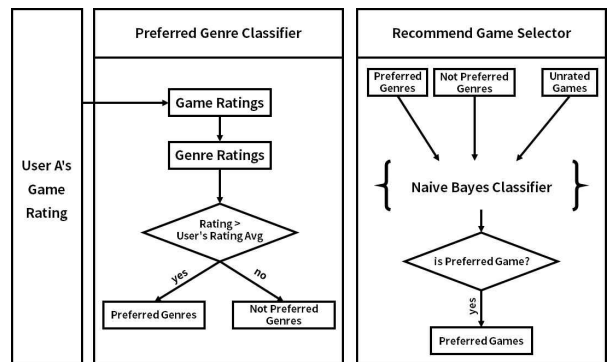


Fig. 5 Process of Game Recommendation

예를 들어 사용자 A가 평가한 게임이 Table 1과 같을 때, 나이트 베이지안을 이용해 게임 D에 대한 선호, 비선호 판별을 하게 되면, D에 속한 장르는 Horror와 Racing이기 때문에, 다음과 같은 식으로 게임 D가 선호일 확률을 계산하게 된다.

$$P(L_{like}|G_D) = P(Horror|L_{like}) * P(Racing|L_{like}) * P(L_{like})$$

$P(T_{Horror}|L_{like})$ 는 사용자 A의 평균 평점인 3점 이상인 게임 2개 중에 1개에서 Horror를 찾을 수 있으므로  $\frac{1}{2}$ 이다. 또,  $P(T_{Racing}|L_{like})$ 는 동일한 방법으로 계산하였을 때  $\frac{1}{2}$ 를 얻을 수 있다.  $P(L_{like})$ 는 평가한 3개의 게임 중에 평균을 넘는 게임이 2개가 있으므로  $\frac{2}{3}$ 을 얻을 수 있다. 따라서  $P(L_{like}|G_D)$ 는  $\frac{1}{6}$ 이란 것을 알 수 있다. 또, 동일하게  $P(L_{dislike}|G_D)$ 를 계산하면, 0을 얻을 수 있기 때문에, 사용자 A에 대해 게임 D는 선호일 것이라 판단할 수 있다.

Table 1 Example of Game Ratings of User A

Game	Genre	Rating
A	Shooting, Horror, Action	3
B	Shooting, Racing	4
C	Action, Fighting	2
D	Horror, Racing	

## 4. 테스트 및 분석

### 4.1 예상 점수 측정

#### 4.1.1 실험 배경

예상 점수 측정 알고리즘은 사용되는 이웃의 수가 변하면 정확도와 계산 시간에 차이가 발생하기 때문에 이웃의 수를 10, 30, 50, 70, 100 명

으로 각각 조절하여 최적의 이웃 수를 결정할 수 있도록 하였다. 이 실험의 실험 대상은 Table 2와 같이 선정하였다. 여기서 희소성 수준은 행렬 내에 평가가 없는 요소의 비율을 나타낸 것이다. 예를 들어 사용자 10명, 항목 10개에 대한 10x10 행렬에서 항목의 수가 10개가 존재 한다면 희소성 수준은 0.9가 된다. 또한, 제한 기간별로 실험을 진행하여 정확도, MAE, 계산 시간을 측정하였다. 실험 데이터는 Table 3과 같으며, 제한 기간을 3개월, 6개월, 1년, 전체 기간으로 각각 조절하여 최적의 제한 기간을 결정할 수 있도록 하였다. 이 때, 이웃 수는 100 명으로 고정하였다.

Table 2 Dataset for Performance Evaluation depending on the Number of Neighbors

Subject	random user with over 100 rating
Matrix Size (Number of User × Number of Games)	105478 × 3781
Rating Range	0 ~ 5 real number
Scarcity Level	0.9993

Table 3 Dataset for Performance Evaluation depending on the Period Limit

Subject	random user with over 20 ratings for each period limit
Matrix Size (Number of User × Number of Games)	105478 × 3781
Rating Range	0 ~ 5 Real Number
Scarcity Level	0.9993

실험 방법은 선정된 평가 대상의 평가 정보의 무작위 20%를 삭제 한 뒤, 예상 점수를 측정하였다. 그 뒤, 예상 점수와 실제 점수를 비교 하였다. 평가 기준은 정확도, MAE, 계산 시간 3

가지로 나누었고, 이 중 정확도는 실제 점수와 예상 점수가  $\pm 0.6$  미만의 차이가 발생했을 때 성공으로 판단하였다. MAE는 (6)과 같은 방법으로 계산하였다. 이 때,  $r_{k,i}$ 는 게임  $i$ 에 대해 사용자  $k$ 가 부여한 평점이고  $r'_{k,i}$ 는 항목  $i$ 에 대해 사용자  $k$ 의 예상 점수이고  $N$ 은 성능 평가에 사용된 대상의 개수이다.

$$MAE = \frac{\sum_i^N |r_{k,i} - r'_{k,i}|}{N} \quad (6)$$

#### 4.1.2 성능 평가

Fig. 6, Fig. 7, Fig. 8은 보완 코사인 유사도 계산식을 이용한 협력 필터링 알고리즘을 사용하였을 때, 각각 이웃 수에 따른 정확도, MAE, 계산 시간 변화를 나타낸 것이다. 먼저 정확도의 경우 70명까지 약 15%의 상승률을 보였지만, 100명에서 0.581%의 상승률을 보였고, 이웃 수 100에서 최저 62.79%, 최고 83.744%의 정확도를 얻을 수 있었다. 실험 결과, 정확도는 대상에 따라 많은 차이를 보였는데 낮은 정확도의 경우 상대적으로 많은 사람들이 평가 하지 않은 게임들을 사용자가 상당수 평가한 경우에 선정된 이웃과 중복되는 게임이 많지 않아 정확도가 낮았다. 또한, MAE의 경우에도 이웃 수 70까지 큰 감소폭을 보이다 100에서 70과 비슷한 MAE 값을 보였다. 이와 같은 결과의 원인은 이웃의 수가 늘어난다 하더라도 이후 추가되는 이웃들과는 중복되는 평가 게임의 수가 많지 않기 때문에 정확도 상승에 많은 기여를 하지 못한다고 볼 수 있다. 또한, 계산 시간도 10에서 30으로 늘어났을 때보다 70에서 100으로 늘어났을 때 덜 증가한 것을 볼 수 있는데 이도 마찬가지로 이후 추가되는 이웃들은 중복되는 평가 게임의 수가 많지 않아 계산시간이 짧기 때문에 덜 증가하는 것으로 볼 수 있다.

다음으로 Fig. 9, Fig. 10, Fig. 11은 제한 기간에 따른 성능을 보인다. 정확도는 3달에서 6달간에 5.903% 상승폭을 보였으며, 6달에서 77.906%에 달했다. 이후 전체 기간에서 71.511%

까지 떨어지며 최저 정확도를 보였다. MAE도 마찬가지로 6달이 0.438로 가장 낮았으며, 전체 기간이 0.479로 가장 높았다. 또한, 전체 기간의 계산시간은 2.89초로 6달에 비해 0.5초 늘어난 것을 볼 수 있다.

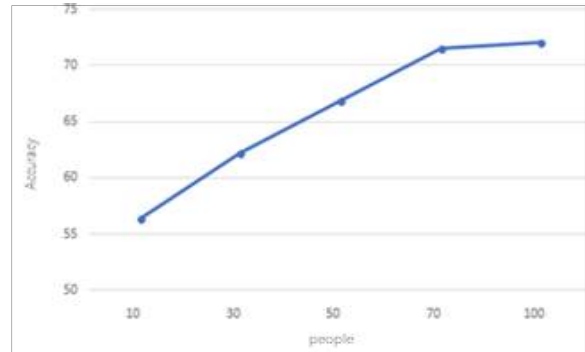


Fig. 6 Accuracy by the Number of Neighbor

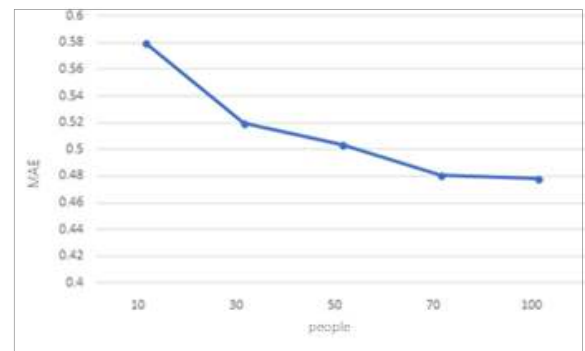


Fig. 7 MAE by the Number of Neighbor

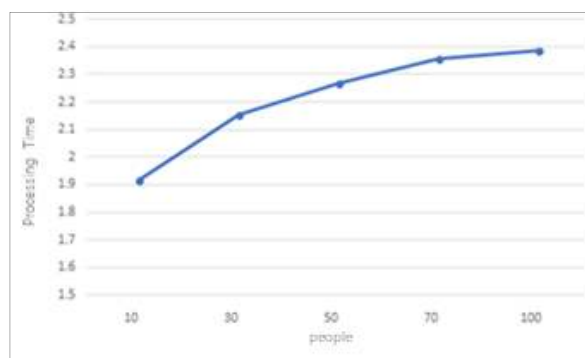


Fig. 8 Processing Time by the Number of Neighbor



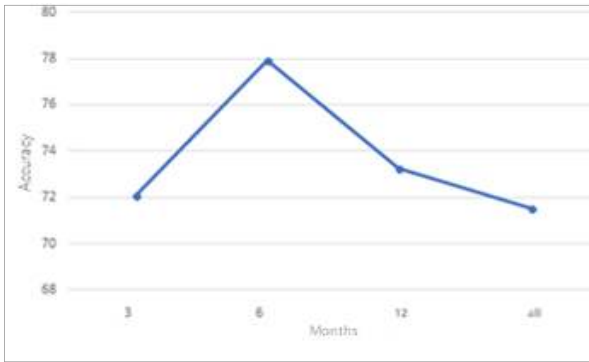


Fig. 9 Accuracy by Period Limit

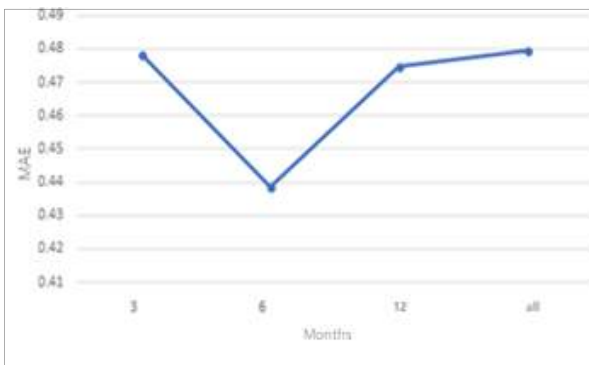


Fig. 10 MAE by Period Limit

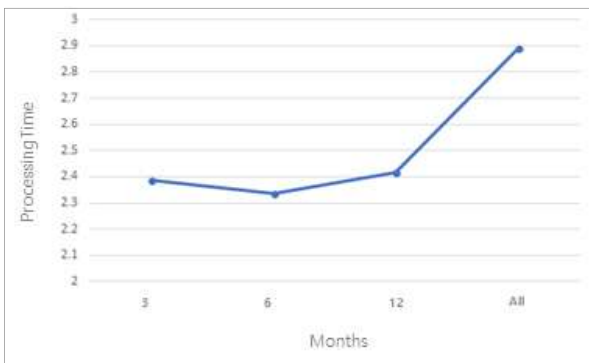


Fig. 11 Processing Time by Period Limit

#### 4.2 게임 추천

게임 추천 알고리즘의 정확도는 선호 기준에 의해 좌우된다. 따라서, 선호 기준에 사용자의 평균 평점이 적합한지를 판단하기 위해, 선호 기준을 1~5점으로 변경하면서 실험 하였고, 그것의 정확도와 평균 평점의 정확도를 비교하였

다. 실험 대상은 Table 4와 같다. 실험 방법은 선정된 평가 대상이 입력한 평가 정보의 무작위 20%를 삭제 한 뒤, 삭제한 게임에 나이브 베이 지안을 적용하여 나온 결과 값을 실제 평가 값 이 평균 평점을 넘었는지 확인하여 비교하였다.

Table 4 Dataset for Naive Bayesian Classification

Subject	random user with over 100 rating
Matrix Size (Number of User × Number of Games)	105478 × 3781
Rating Range	0 ~ 5 Real Number
Scarcity Level	0.9993

실험 결과, Fig. 12와 같이 선호 기준을 고정 으로 설정하였을 때, 정확도가 0.5점에 머무 르는 것을 볼 수 있었다. 이유는 사용자 별로 평 균 평가 점수가 제각각인데 선호 기준을 고정으 로 설정하면 선호 기준에서 너무 떨어진 평균 평가를 가지고 있는 사용자는 나이브 베이 지안 의 결과가 모두 선호이거나 모두 비선호인 결 과를 가졌기 때문이다. 이에 반해 선호 기준을 각 사용자의 평균 평가 점수로 하게 되면, 나이브 베이 지안이 사용자의 평가 특성을 반영할 수 있 기 때문에 고정 기준의 결과 값에 비해 높은 75.187퍼센트의 정확도를 얻을 수 있었다.

#### 5. 결론 및 향후 연구

현재까지 시중에 나온 게임 추천 시스템은 게임 판매 플랫폼의 마케팅 전략을 위해 만들어진 수단으로 사용되어왔다. 하지만, 본 논문에서는 최신 웹 개발 프레임워크인 Angular를 이용하여, 데이터 중심의 웹페이지를 구성할 수 있게 되어, 사용자에게 더 나은 웹 경험을 선사할 수 있게 되었다.

또한, 본 논문에서는 협력 필터링에서 확장성 문제를 해결하기 위해 이웃 수와 최신 평가를

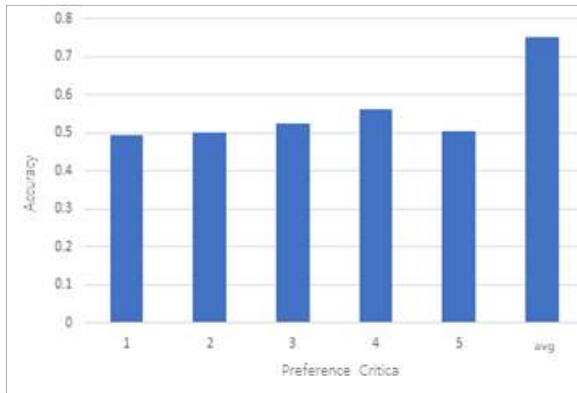


Fig. 12 Accuracy by Preference Criteria

이용하였고, 정확도를 크게 해치지 않으면서 계산 시간을 대폭 감소 할 수 있었다. 또한, 나이브 베이지안 분류식을 통해 장르별 선호도를 통해 선호 게임을 선정할 수 있었다.

향후, 예상 점수의 계산 결과를 추천에 결합하여, 사용자가 더 나은 추천을 받을 수 있도록 할 계획이다.

### References

[1] Francesco, R., Lior, R., and Bracha, S., "Introduction to Recommender Systems Handbook," Recommender Systems Handbook, Springer, pp. 1-35, 2011.

[2] Kim, B. M., Li, Q., Kim, S. G., Lim, E. K., and Kim, J. Y., "A New Approach Combining Content - Based Filtering and Collaborative Filtering for Recommender Systems," Journal of KISS : Software and Applications, Vol. 31, No. 3, pp. 332-342, 2004.

[3] Kim, S. Y., Lee, S. H., and Hwang, H. S., "Design and Implementation of Personalized Recommendation System Using AHP and Hybrid Filtering," Journal of the Korea Industrial Information Systems Research, Vol. 17, No. 7, pp. 111-118, 2012.

[4] Nam, Y. J., Shin, D. I., and Shin, D. K.,

"Sensor Data Classification Using Naive Bayesian Classifier," Proceedings of Symposium of the Korean Institute of Communications and Information Sciences, pp. 90-91, 2015.

[5] Brent, S., Greg, L., "Two Decades of Recommender Systems at Amazon.com," IEEE Internet Computing, Vol. 21, No. 3, pp. 12-18, 2017.

[6] Carlos, A., Gómez, U., Neil, H., "The Netflix Recommender System," ACM Transactions on Management Information Systems, Vol. 6, No. 4, pp. 1-19, 2015.

[7] <https://www.metacritic.com/game>

[8] <https://store.steampowered.com/>

[9] Yun, Y. J., Kim, J. H., Jo, H. J., Yun, G. J., Jeong, K. H., Hong, S. P., Kim, B. M., and Le, H. Y., "Game Recommendation System Using Naive Bayesian Classification," Proceedings of KIIT Summer Conference, pp. 185-187, 2018.

[10] Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P. and Riedl, J., "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," Proc. of ACM Conf. on Computer-Supported Cooperative Work, pp. 175-186, 1994.

[11] Bardrul, S., George, K., Joseph, K., and Riedl, J., "Item-based Collaborative Filtering Recommendation Algorithms," Proc. of WWW 10, 2001.

[12] Lee, S. J., "Optimization of the Similarity Measure for User-based Collaborative Filtering Systems," The Journal of Korean Association Of Computer Education, Vol. 19, No. 1, pp. 111-118, 2016.

[13] Hwang, Y. W., Kim, J. H., Kim, B. M., and Lee, H. A., "Game Recommendation System Based on Collaborative Filtering," Proceedings of the 2017 KISS Conference, pp. 1914-1916, 2017.

[14] Lee, S. J. and Lee, S.W., "A Comparison

Study on Preference Calculate Methods for Content-Based Recommendation,”  
Proceedings of the 2010 KISS Conference,  
Vol. 37, No. 2, pp. 222-227, 2010.



김 중 현 (JongHyen Kim)

- 정회원
- 2015년 ~ 현재 : 금오공과대학교 컴퓨터소프트웨어공학과 학부생
- 관심분야 : 인공지능, 빅 데이터 처리



조 현 정 (HyeonJeong Jo)

- 정회원
- 2015년 ~ 현재 : 금오공과대학교 컴퓨터소프트웨어공학과 학부생
- 관심분야 : 인공지능, 영상처리



김 병 만 (Byeong Man Kim)

- 정회원
- 1987년 : 서울대학교 컴퓨터공학과 학사
- 1989년 : 한국과학기술원 전산학과 석사
- 1992년 : 한국과학기술원 전산학과 박사
- 1992년 ~ 현재 : 국립금오공과대학교 교수
- 1998년 ~ 1999년 : 미국 UC, Irvine 대학 방문 교수
- 2005년 ~ 2006년 : 미국 콜로라도 주립대학 대학 방문교수
- 관심분야 : 인공지능, 정보검색, 정보보안