공학석사 학위논문

희박한 평점데이터를 가진 사용자를 위한 멘토 기반의 영화추천 시스템

2014년 2월

서울대학교 융합과학기술대학원

융합과학부 디지털정보융합전공

천성권

희박한 평점데이터를 가진 사용자를 위한 멘토 기반의 영화추천 시스템

지도 교수 이 교 구

이 논문을 공학석사 학위논문으로 제출함
2014년 2월

서울대학교 융합과학기술대학원
융합과학부 디지털정보융합전공
천 성 권

천성권의 공학석사 학위논문을 인준함
2014년 2월

위 원 장 강 남 준 (인)

부위원장 서 봉 원 (인)

위 원 이 교 구 (인)

초 록

통신 기술의 발달과 스마트 기기의 보급은 영화 콘텐츠를 시청 하는 방식의 다양화를 가져왔다. 그에 발맞추어 영화 콘텐츠의 양도 시간이 지남에 따라 급격하게 증가하고 있다. 그러나 자신의 취향에 맞는 영화를 고르기란 언제나 쉬운 일은 아니다. 보고 싶은 영화가 정해져 있는 경우에는 검색 서비스를 사용하여 관련 정보를 찾을 수 있지만, 그렇지 않은 경우에는 검색에 제한이 있다. 따라서 추천시스템은 검색서비스의 한계점을 개선하고, 사용자의 다양한 니즈를 해결하기 위해 적합한 대안이다. 왜냐하면, 추천시스템은 사용자 및 콘텐츠 자체의 특성(feature)을 기반으로 아이템을 추천 하기 때문에 보다 합리적인 결과를 제시할 수 있다.

지금까지 영화추천시스템에 대한 연구는 상당한 진척을 보여왔다. 하지만 기존 연구들의 특징은 무비렌즈(MovieLens)와 같이 정해진 데이터 집합을 이용하여 연구가 진행되었다는 점이다.

여기서 주목할 점은, 무비렌즈 데이터 집합이 '적어도 각 사용자는 20개의 각기 다른 영화에 대한 평점 정보를 가진다' 라고 밝혔다는 것이다. 다시 말해, 실제 평점개수가 20개 미만인 사용자는 추천 대상에서 완전히 제외가 된다는 이야기다. 하지만 실제 영화 사이트에서 평점을 20개 이상 남긴 사용자는 전체 사용자를 기준으로 매우 적을 것이라 예상된다. 따라서 이 연구에서는 사용자 데이터 희박성(User data sparsity) 해결에 초점을 맞춘 멘토(mentor) 기반의 영화 추천시스템을 제안하고자 한다. 멘토 기반의 영화 추천시스템은 특히 사용자가 1-2개의 극도로 희박한 평점정보를 가지고 있을 때도 의미 있는 영화 추천을 가능하게 해주는 것을 목표로 한다. 멘토(mentor) 기반의 영화 추천을 설계하기 위해 실제 사용자들이 많이 사용하고 있는 다음(Daum) 포털사이트에 영화 데이터(사용자, 영화, 리뷰 정보)를 수집하여 멘토 기반의 추천시스템 알고리즘을 구현한다.

최종적으로 이 연구에서 제안하는 추천시스템과 기존 알고리즘의 추천 결과를 비교 제시하여 시스템 성능 평가를 수행한다.

주요어 : 영화 추천시스템, 협업 필터링, 사용자 데이터 희소성
학 번 : 2011-22771

목 차

표 목차

그림 목차

제 1 장 서 론

제 1 절 연구 배경

통신 기술의 발달과 스마트 기기의 보급은 영상 콘텐츠를 시청 하는 방식의 다양화를 가져왔다. 예를 들어, 이제는 지하철이나 길을 걸어가면서 스마트폰, 스마트 패드 등을 사용하여 온라인 영상 콘텐츠를 시청하는 사람들을 쉽게 목격할 수 있다. 이러한 변화에 발맞추어, 특히 영화 산업은 유·무선 통신을 활용한 온라인 시장에 많은 투자를 하고 있다. 구체적으로, 국내 온라인 영화시장의 규모는 2011년을 기준으로 총 매출 추정 규모가 1,411억 정도로 집계된다. [25] 또한, 다양한 서비스들이 온라인 영화 시장에 자리를 잡기 시작했는데 모바일 서비스, VOD(다운로드, 스트리밍), IPTV 등을 중심으로 점점 확대 되고 있다.

영화 콘텐츠의 양도 시간이 지남에 따라 증가하고 있다. 전 세계적으로 제작된 극장 개봉용 장편 영화는 2010년을 기준으로 총 5,669편으로 하루 평균 15.5편의 작품이 완성되고 있다. [26] 이는 극장 개봉 영화만을 기준으로 집계되었기 때문에 실제 하루 평균 완성되는 작품은 더 많을 것으로 예측된다. 한국영화만을 놓고 보았을 때 2011년 한 해를 기준으로 150편 정도로 2.4일에 1편 정도 영화가 개봉되고 있다. 따라서 공급이 수요를 넘어섰다고 볼 수 있다. 그리고 시간이 갈수록 영화 콘텐츠의 양은 지금보다도 더 급격하게 증가하여 수용격차는 더 커질 것으로 예상된다.

그러나 보고 싶은 영화를 고르기란 언제나 쉬운 일은 아니다. 보고 싶은 영화가 정해져 있는 경우에는 검색 서비스를 사용하여 관련 정보를 찾을 수 있지만, 그렇지 않은 경우에는 검색에 제한이 있다. 이러한 상황에서 대부분의 사용자는 영화 정보사이트에서 제공하는 전체 평점(전체 사용자에 평점을 평균 낸 점수)을 선택의 기준으로 삼거나, 사람들이 가장 많이 본 영화, 박스오피스 등을 참고 하게 된다. 물론, 지인들에게 추천을 받는 경우도 있지만 수많은 영화들 중에서 자신에게 맞는 장르, 배우, 감독, 줄거리 등을 찾는 것은 여전히 쉽지 않다. 요컨대 사용자는 수많은 영화 중에서 기호에 맞는 영화를 찾기 위해 많은 노력과 시간을 투자해야 한다.

추천시스템은 검색서비스의 한계점을 개선하고, 사용자의 다양한 니즈를 해결하기 위한 적합한 대안이다. 왜냐하면, 추천시스템은

사용자 및 콘텐츠 자체의 특성(feature)을 기반으로 추천을 하기 때문에 보다 합리적인 결과를 제시할 수 있다. 구체적으로, 추천시스템이 기준으로 삼는 특성은 아이템의 내용, 사용자 선호 프로필, 시간과 장소 같은 맥락 등이 있다. 그래서 추천시스템을 사용하면 여러 가지 측면으로 사용자에게 적합한 콘텐츠를 효율적으로 제시할 수 있게 되는 것이다.

하지만 추천시스템의 유용성에도 불구하고 국내에서 영화 추천 서비스를 제공하는 업체는 극히 드물고, 기존의 알고리즘에 대한 한계로 인해 다양한 문제점들이 드러나고 있다. 따라서 이 연구에서는 국내 영화 콘텐츠를 포함하고, 다수의 사용자들에게 높은 만족도를 이끌어 낼 수 있는 추천시스템 방법을 제안하고자 한다.

제 2 절 연구 목표

지금까지 영화추천시스템에 대한 연구는 상당한 진척을 보여왔다. 하지만 기존 연구들의 특징은 무비렌즈(MovieLens)[①]와 같이 정해진 데이터 집합을 이용하여 연구가 진행되었다는 점이다. 여기서 주목할 점은, 무비렌즈 데이터 집합이 '적어도 각 사용자는 20개의 다른 영화에 대한 평점 정보를 가진다' 라고 밝혔다는 것이다. 다시

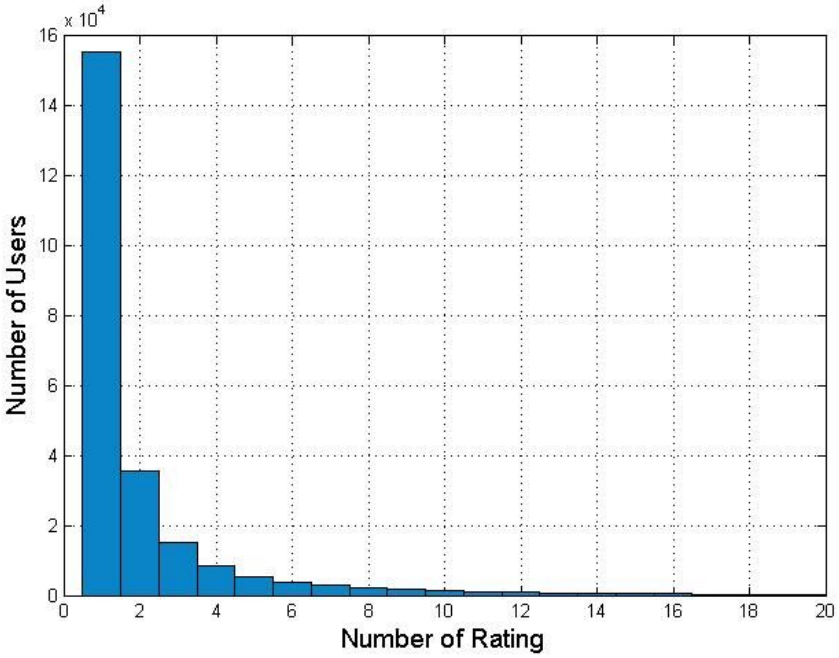
말해, 실제 평점개수가 20개 미만의 사용자는 추천 대상에서 완전히 제외가 된다는 이야기다. 선행연구 제2장에서 자세히 언급하겠지만, 협업 필터링(Collaboriative filtering)을 사용하는 추천 방식의 경우 사용자의 평점 정보의 개수가 일정한 수 이상이 존재 해야 추천 성능이 높아진다.

하지만 평점을 20개 이상 남긴 사용자는 전체 사용자를 기준으로 매우 적을 것이라 예상된다. 실제로, 국내 유명 포털 사이트 중에 하나인 다음(Daum) 영화 데이터를 기준으로 조사한 결과 [그림1]과 같은 그래프가 나타났다.

[①] MovieLens 데이터 집합은 Minnesota 대학의 GroupLens Research Project에 의해 수집된 데이터이며, 다음과 같은 특징을 갖는다

[표 1] MovieLens 데이터 집합의 한 종류 – MovieLens 100k

MovieLens 100k
1682개의 영화는 943 사용자로부터 100,000개의 평점 정보(1–5점) 가진다
각 사용자는 적어도 20개의 다른 영화에 대한 평점 정보를 가진다



[그림 1] 다음(Daum)영화 데이터 (사용자-평점 개수)

[그림1]에서 보이는 것처럼 전체 사용자 241,704명 중에 평점을 1-19개 남긴 사용자는 237,018명으로 전체 98.06%를 차지한다. 이 결과는 바꾸어 말하면 지금까지 진행되어온 대부분의 연구들이 98.06%에 해당하는 사용자를 수용하지 못한다는 이야기이다. 이는 기존 연구들에서 제안하는 방법들이 98.06%에게도 같은 추천 성능을 보일지는 모른다는 뜻이다.

따라서 이 연구는 사용자의 대부분을 차지하고 있는 1-2개의 희박한 평점 정보를 가진 사용자 191161명(79.08%)에 초점을 맞추어 평점 정보가 낮은 사용자에게도 보다 의미 있는 영화 추천 방법을 제안하고자 한다.

또한 이런 추천을 통해 사용자로 하여금 평점의 개수 증가를 이끌어, 협업필터링 알고리즘상에서의 최상의 결과를 도출 하기 위한 중간 과정 역할을 목표로 한다.

이 논문의 구성은 다음과 같다. 연구를 시작하게 된 동기와 연구 목표를 먼저 1장에서 언급한 뒤, 2장에서 선행연구를 살펴볼 것이다. 3장에서는 연구방법을 설명하고 4장에서는 사용자 평가를 이용한 시스템 성능 평가를 수행할 것이다. 성능 평가를 분석, 마지막으로 5장에서는 연구 성과를 정리하면 끝 맺음을 하려 한다.

제 2 장 선행연구

제 1 절 추천 시스템과 영화

추천시스템은 과거의 사용자 선호 정보를 분석하여 사용자가 만족할 만한 새로운 아이템을 예측하고, 그 중에서도 높은 만족도가 예상되는 아이템을 추천해준다.

이러한 추천시스템 연구는 1990년대 중반에 협업적 필터링에 관한 첫 번째 논문이 나온 이후 독립적인 연구분야가 되었다. [1] [2]

그 후 Tapestry 추천시스템이 등장하였는데 이는 협업 필터링(Collaborative filtering) 기반의 추천시스템으로 가장 먼저 구현된 형태 중 하나이다. Tapestry는 작업그룹(office workgroup)과 같은 비교적 작은 형태의 그룹 안에서 사람들의 명확한 의견을 반영한 시스템이다. 하지만 Tapestry는 소규모 네트워크 안에서 다른 사람의 의견과 선호를 반영하는 것은 어렵지 않았으나 큰 커뮤니티에는 적용할 수 없다는 단점을 가졌다. 이후에도 몇 가지의 평점기반의 자동화된 시스템이 개발되었는데 대표적으로 GroupLens [3], Ringo [2], Video Recommender [4] 시스템을 들 수 있다. 각각 간단히 요약하면 GroupLens는 다양한 뉴스 기사들 중에서 사용자가 과거에 만족했었던 기사와 비슷한 기사들을 발견하는데 도움을 주는 시스템이다. Ringo, Video Recommender도 마찬가지로 음악과, 영화 분야에서 개인화된 웹 기반 추천시스템을 말한다.

현재까지도 추천시스템에 관한 연구는 무수히 많이 진행되고 있다. 뿐만 아니라 전자 상거래에서도 제품의 판매 촉진을 위하여 추천시스템을 도입하고 있다. 대표적으로 Amazon.com, IMDb.com 등을 들 수 있다.

이 밖에도 동영상 전문 사이트로 유명한 유튜브(Youtube) 도 추천시스템을 적용하여 많은 사용자의 호응을 얻고 있다. [5] 유튜브 추천시스템의 특징은 사용자가 본 동영상과 같은 명백한 콘텐츠 데이터 뿐만 아니라 얼마나 오랫동안 보았는지에 관한 사용자활동 데이터, 즉 암묵적인 데이터도 추천 과정에 포함하여 사용자의 추천 만족도를 크게 높였다. [그림 2]는 실제 유튜브 추천시스템을 이용하여 나온 추천 결과이다.

[그림 2] 유튜브 추천시스템을 이용한 추천결과

이렇게 이미 추천시스템은 우리 주변에서도 쉽게 찾아 볼 수 있을 정도로 널리 사용되고 있고, 이와 함께 사용자에게 더 큰 만족도를 주기 위한 연구는 계속되고 있다.

추천시스템은 그 동안 추천 접근 방식에 따라 세가지 방법으로 분류되었으나, 최근에 상황 인지 추천방법이 등장하면서 크게 네 가지 형태로 분류되고 있다. 각 추천방법에 대해 간단한 설명을 [표 2]로 요약했다.

[표 2] 추천 접근 방법

추천 접근 방법	설명
내용기반 추천 (Content-based)	사용자가 과거에 선호했던 아이템과 내용이 비슷한 아이템들을 추천
협업 필터링 추천 (Collaborative filtering)	사용자는 과거에 좋아했었던 선호와 취향이 비슷한 다른 사용자의 아이템을 추천
하이브리드 추천 (Hybrid)	내용기반 방법과 협업적 필터링을 결합하는 방법
상황 인지 추천 (Context-Aware)	사용자의 상황 정보를 고려하여 아이템을 추천

다음에서는 표에서 요약된 추천 접근 방법에 대해 세부적인 설명과 각 방법을 영화에 적용한 사례를 제시하고자 한다.

1. 내용 기반 추천(Content-based recommendation)

내용 기반 추천시스템은 사용자가 과거에 선호했던 아이템의 내용에 관한 공통점을 파악하여 사용자 선호 프로필을 생성한다. 그 후 새로운 아이템들에 대해 내용 분석 과정을 거쳐 사용자 선호 프로필과 유사한 아이템을 추천해주는 방법이다. 이러한 내용기반 접근은 정보 검색(Information retrieval)과 정보 필터링(Information filtering) 연구에 뿌리를 두고 있다. [6]

그 이유는 내용기반 접근 자체도 아이템 안에 존재하는 원문(Textual) 분석에 집중하기 때문이다. 따라서 정보 검색에서 연구되었던 텍스트 분석 기술들이 내용기반 분석에 중요하게 이용한다.

하지만 내용 기반 추천이 텍스트 기반의 분석만 존재하는 것은 아니다. 대표적으로 음악 추천시스템 같은 경우, 음악의 원 자료(raw data)를 바탕으로 신호 분석(Signal Processing)을 사용하여 음악과 음악을 비교한다.

영화 추천시스템에 경우 영화 아이템의 긴 영상 자체를 분석하고 비교하는 것이 어렵기 때문에 주로 텍스트 기반 분석을 주로 사용한다. 영화의 대표적인 텍스트로는 영화 메타 데이터(meta-data)가 존재한다. 여기서 영화 메타데이터는 영화 제목, 감독, 배우, 장르, 나라, 줄거리, 시간, 평점, 연령 등으로 구성된다. 이렇게 텍스트 기반의 메타데이터를 가지고 영화를 설명 할 수 있기 때문에 내용 기반 방법을 적용할 수 있다.

이처럼 영화의 내용기반 방법을 적용한 연구사례는 대표적으로 Li와 Mak가 있다. Li [7]는 사용자가 평가한 영화와 관련해서 영화 메타 데이터의 속성값을 적용하여 귀납적 학습(Inductive Learning)중에 하나인 의사결정나무(Decision Tree)로 사용자 선호를 나타냈다. 이 의사결정나무를 통하여 새로운 아이템에 대해 사용자 선호 평가 값을 분류했다. Mak [8]는 사용자가 평점 정보를 부여한 영화의 줄거리(Synopsis)를 기계 학습하여 텍스트 분류기를 만들고, 그 분류기를 통해 새로운 영화를 분류하게 한다. 분류된 아이템 중 높은 점수를 얻은 영화를 사용자에게 추천하는 방법의 웹 기반 영화추천시스템을 제안했다.

2. 협업 필터링 (Collaborative filtering)

내용기반 추천과 달리 협업 추천시스템 (또는 협업 필터링 시스템)은 다른 사용자가 이전에 평가 점수를 남긴 아이템을 기반으로 추천 받는 사용자를 위해 여러 아이템들의 가치를 예측한다. [9] 영화추천분야에 적용해보면, 사용자는 영화를 추천 받기 위해서 협업 추천시스템은 동료(peers)를 먼저 찾는다. 이때 동료의 정의는 영화 취향이 비슷한 다른 사용자를 뜻하는데, 영화 취향이 비슷하다는 것은 같은 영화에 비슷하게 점수를 남겼다는 것을 근거로 한다. 이렇게 찾은 동료들을 기반으로 대부분에 동료들이 좋아하는 영화를 추천해주는 방식이다.

협업 필터링에 대한 보다 자세한 연구는 2절에서 진행하기로 한다.

3. 하이브리드 기반 추천(Hybrid recommendation)

하이브리드 기반 추천시스템은 내용기반의 추천시스템(Content-based Recommendation)과 협업 필터링 추천시스템(Collaborative filtering)을 결합한다. 하이브리드 기반 추천시스템은 협업 필터링의 단점인 콜드 스타트 문제(Cold Start Problem)와 해리포터 문제(The Harry Potter Problem)를 내용기반 분석을 통해 보완하고 내용기반의 문제점인 내용분석의 한계를 집단지성을 이용한 협업 필터링으로 보완한다. Christakou [10]는 협업적 필터링의 문제중의 콜드 스타트 문제 (Cold-Start Problem)를 해결하기 위해 하이브리드 기반의 영화 추천을 제안했다. 이때 내용기반의 추천은 영화 메타데이터인 영화의 종류, 줄거리, 배우, 감독, 작가 등을 특성으로 신경 네트워크 알고리즘(Neural network)에 적용하였고 피어슨 상관계수 공식을 사용하여 협업 필터링을 구현하였다. Debnath [11]는 소셜 네트워크 그래프(Social Network Graph)로부터 얻어진 사용자들의 판단 정보를 영화 메타데이터의 속성 값 (배우, 감독 등)들에 대한 가중치를 부여하여 속성값의 차이를 만들어 추천하는 방법을 제안하였다.

4. 상황 기반 추천(Context-aware based recommendation)

상황(Context)이란 한 독립체(entity)의 환경을 특징 짓기 위해 사용될 수 있는 어떤 정보를 말한다. 이때 독립체는 사람, 장소, 물리적 또는 컴퓨터 객체가 될 수 있다. [12] 이처럼 상황 기반 추천은 사용자 사이에 둘러 쌓인 시간, 감정, 장소 등과 같은 여러 가지 환경을 고려하여 사용자에게 상황에 맞는 아이템을 추천하는 방법이다. 이러한 상황 기반 추천을 반영한 대표적인 영화 추천 사례는 Ono [13]가 제안한 베이지안 네트워크(Bayesian Newtork)를 적용한 상황-인식 영화 선호 모델이다. 이 시스템은 사용자의 상황정보 (동행 인물, 장소, 감정)를 입력 받고 등록 되어 있는 사용자 정보와 결합, 베이지안 네트워크 추론 엔진을 사용하여 후보 영화들의 평점을 확률적으로 계산하는 방법을 제시하였다.

5. 협업 필터링 시스템의 우수성

이처럼 추천시스템에서 추천이 이루어지는 방법은 크게 네 가지가 존재한다. 하지만 내용기반 분석의 경우 콘텐츠를 분석하고 비교하는 것이 까다롭고, 상황 기반 추천의 경우 사용자의 복합적인 상황 정보를 추출하고 각 상황마다 적절한 추천을 하는 것이 어렵다. 따라서 이 연구에서는 추천시스템에서 가장 많이 연구되고 있는 협업 필터링을 이용한다. 협업 필터링은 아이템 내용 정보를 파악할 필요가 없고 실제 사용자가 평가한 점수를 바탕으로 추천해주기 때문에 사용자의 만족도가 높다. 이러한 이유로 여러 전자상거래 사이트 Amazon.com, CDnow 등에서도 협업 추천시스템을 적용하여 성공을 거두고 있다. 이 연구 또한 협업 필터링 방식을 적용한다. 따라서 2절에서는 협업 필터링 방식에 대해 자세히 살펴보고 협업 필터링에 대한 한계도 설명한다. 마지막으로 이러한 한계를 개선하기 위한 노력을 살펴본다.

제 2 절 협업 필터링 영화추천 시스템

협업 필터링 시스템은 크게 두 가지 방법이 존재한다. 하나는 사용자 기반의 협업 필터링 방식과 다른 하나는 아이템 기반의 협업 필터링 방식이다. 두 가지 방법은 대상(사용자, 아이템)의 차이가 있을 뿐 전체적인 추천 과정은 비슷하다. 아이템 기반의 협업 필터링은 아이템들 사이에 유사도를 계산한 다음 사용자가 좋아했던 아이템과 연관된 아이템들을 추천한다. 이와 달리 사용자기반의 협업 필터링은 사용자들 사이에 유사도를 계산하고 비슷한 사용자들이 좋아했던 아이템을 추천해주는 형태이다.

[그림 3] 사용자−아이템 행렬을 통한 유사도 표

그 중 사용자 기반의 협업 필터링 과정을 살펴보면 아래와 같이 세 단계로 설명 할 수 있다. 아래의 방법은 Herlocker [14] 에 의해 제안된 이웃−기반 알고리즘(In neighborhood−based algorithms) 을 사용한 순수한(Pure) 협업 필터링 방법이다.

첫 번째로 추천을 받을 사용자는 모든 사용자들과 유사도를 계산한다. 보통 사용자 사이의 유사도는 사용자가 부여한 점수 벡터 사이의 피어슨 상관계수(Pearson correlation)로 측정 된다

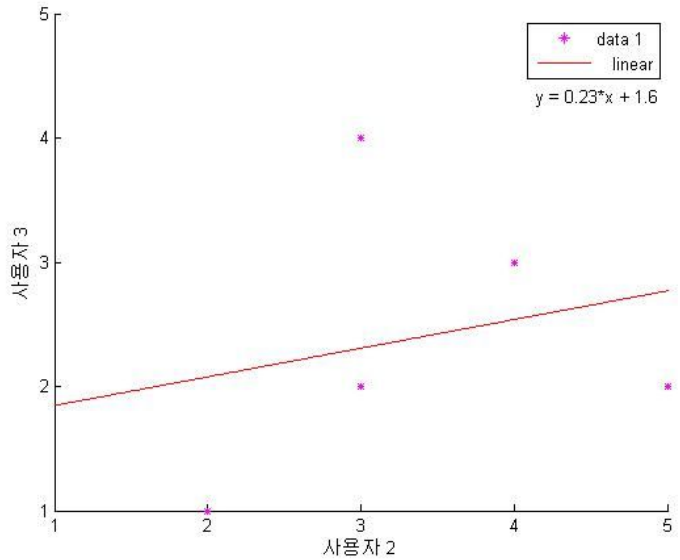
피어슨 상관 계수는 아래 수식 (1)으로 나타낼 수 있다.

여기서 $r_{a,i}$ 는 사용자 a가 아이템 i에 대해서 부여한 점수를 나타낸다. 또한 $\bar{r_a}$ 는 사용자 a의 전체 점수의 평균이다. 마지막으로 m은 아이템의 전체 개수를 나타낸다.

$$P_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i} - \bar{r_a}) \times (r_{u,i} - \bar{r_u})}{\sqrt{\sum_{i=1}^{m}(r_{a,i} - \bar{r_a})^2 \times \sum_{i=1}^{m}(r_{u,i} - \bar{r_u})^2}} \quad (1)$$

이때 결과 값 $P_{a,u}$는 피어슨 상관계수의 범위는 −1과 1사이로 1에 가까울수록 두 사용자는 비슷하다고 본다. −1로 갈수록 두 사용자는 정반대의 경향을 보이고 0은 비슷하지 않다는 것을 뜻한다.

예를 들어 아래 [그림 4]와 같이 [그림 3]에 나타나있는 사용자-아이템 점수 행렬를 기준으로 사용자 2, 사용자 3에 피어슨 상관계수를 구해보았다. [그림 4]에 직선의 기울기가 피어슨 상관계수를 뜻하고 사용자2와 사용자 3은 0.23에 유사도를 가진다고 본다.



[그림 4] 사용자2와 사용자3에 피어슨 상관계수

두 번째로 추천을 받는 사용자와 높은 유사성를 가지는 n명의 사용자를 선택한다. 선택된 n명의 사용자는 이웃(neighborhood)을 형성한다.

세 번째는 선택된 이웃들의 점수 정보의 가중치 결합으로부터 예측 점수를 계산한다. 이때 예측은 이웃의 평균으로부터의 편차의 가중

평균으로 계산된다. 여기서 $P_{a,i}$ 는 사용자 a와 사용자 u의 유사도를 뜻하고 n은 이웃의 수를 뜻한다. (2)

$$P_{a,i} = \bar{r_a} + \frac{\sum_{u=1}^{n}(r_{u,i} - \bar{r}_u) \times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}} \quad (2)$$

이렇게 사용자는 새로운 아이템에 대한 점수를 예측할 수 있다. 예측된 아이템 중에서 가장 높은 점수를 받은 아이템 N (Top-N) 개를 추천해 준다. [15]

1. 협업 필터링 영화 추천 연구

협업 필터링을 이용한 영화 추천 연구는 다음과 같다. Golbeck[16]은 의미론적인 웹 기반(Semantic Web)의 소셜 네트워크(Social Network)를 사용하였다. 사용자는 자신의 친구들을 선택 할 수 있고 이렇게 선택 된 친구들이 부여한 신뢰 있는 평점 정보를 바탕으로 영화 추천 방식을 제안했다. Ding [17]은 협업 필터링 방식에 시간에 대한 가중치를 추가하였다. 최근에 사용자가 남긴 영화에 대한 평점은 이미 오랜 전에 남긴 영화에 평점보다 사용자의 영화 선호를 예측을 하는데 있어서 더 큰 영향력을 차지해야 한다는 내용이다. 시간이 사용자의 선호를 예측하는데 중요한 요인임을 강조하였다. Ungar [18]은 협업 필터링을 위한 클러스터링 방법을 연구했다. 협업 필터링에 문제점인 희박한(sparse) 정보로 인해 모든 영화들 중에 작은 영역의 영화들만 추천해주는 문제점을 가지는데, 비슷한 영화를 중심으로 클러스터(cluster)들 속에 사용자를 그룹화함으로써 좀 더 정확한 예측이 가능하다는 연구이다. Choi [19]는 영화 데이터베이스 안에서 각 영화들에 장르 조합을 기반으로 장르간의 상관관계를 구했다. 그리고 특정한 장르를 선호하는 사용자에게 장르 상관 관계의 협업 필터링 접근을 제안했다. 위와 비슷한 접근으로 Kim [20]은 모바일 환경에서 사용자의 과거 영화 선호 기록을 분석하고 선호장르와 장르 사이의 유사도를 이용하여 아이템을 추천해주는 방식을 제안했다.

2. 협업 필터링 한계

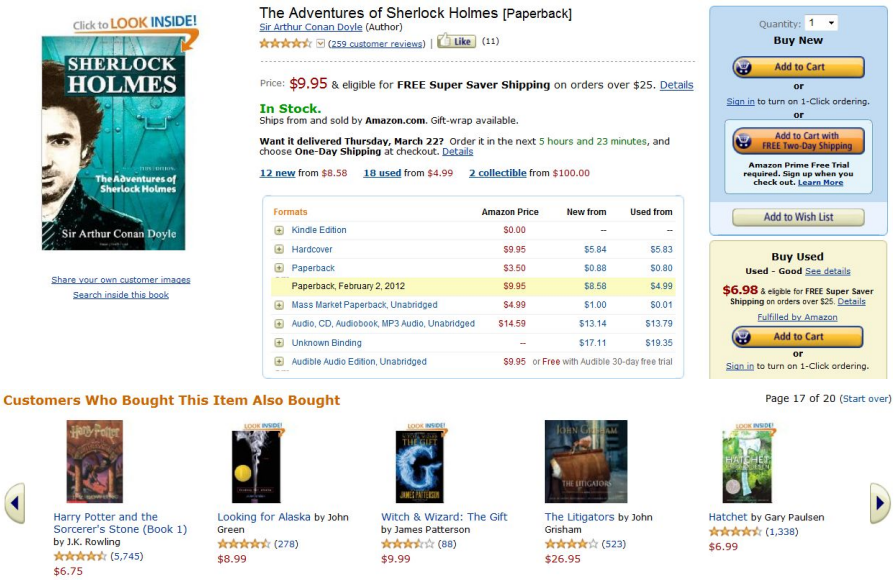
협업 필터링 추천시스템은 전자상거래 사이트 예를 들면 아마존(Amazon), IMDb 에서도 이용될 만큼 추천의 만족도가 높고 널리 사용된다. 하지만 이런 협업 필터링 추천시스템은 사용자들의 아이템 평가 점수를 바탕으로 추천이 이루어지기 때문에 아래와 같은 몇 가지 문제점이 발생한다.

2.1 인기 편향성(Popularity bias)

대중적으로 인기가 많은 아이템에 대해서는 그렇지 않은 아이템에 비해 상대적으로 많은 선택과 평가가 이루어진다. 협업 필터링 시스템은 이렇게 사용자가 아이템에 평가한 점수를 근거로 동작하기 때문에 전혀 연관성이 없는 아이템들이 인기가 높다는 이유로 높은 유사성을 가지게 된다. 따라서 인기가 많은 아이템을 시스템이 추천해주고 사용자는 이런 아이템을 선택하고 평가 하게 된다. 이와 같은 추천이 반복되면 시스템은 결국 대중적으로 인기가 있는 아이템들만 추천해주게 된다. 이는 추천시스템의 기본 목적인 개인화된 추천이 이루어질 수 없다. 또한 사용자들에게 새롭고 유명하지는 않지만 내용적으로 훌륭하고 참신한(novelty) [21] 아이템을 추천해줄 수 없다. 이는 추천시스템의 장점이 사라지게 되어 추천시스템에 존재의미를 퇴색시킨다.

인기편향성에서 언급되는 대표적인 문제로 해리포터 문제(The Harry Potter Problem)를 들 수 있다. 이 문제는 압도적으로 대중적 인기가 있는 아이템이 대부분의 아이템 추천 결과에 나오는 사례이다. 당시 해리포터 책은 남녀노소, 아이부터 어른까지 높은 인기가 있었고 그 결과 다른 책에 비해 해리포터 책은 압도적인 판매량을 보였다. 해리포터를 구입하는 과정에서 해리포터와 관련이 없는 책을 같이 구매하는 경우가 많았는데 이때 추천시스템은 관련이 없는 책과 해리포터 책을 연관 지었다. 그로 인해 대부분의 책을 선택할 때 해리포터 시리즈가 추천되는 현상이 발생하였다.

이 문제는 실제로 [그림 5]에서 확인 할 수 있다. 셜록홈즈 (The Adventures of Sherlock Holmes) 책은 밀리언 셀러로, 해리포터와 같은 높은 대중적 인기가 있다. 따라서 종종 같이 구매되는 경우가 있는데 그때 협업 필터링 시스템은 이 두 책을 연관 짓고, 후에 그 책을 추천 해준다.



[그림 5] 아마존(Amazon.com)에서 셜록홈즈 검색 후 추천결과

결과적으로 참신한(novelty)아이템을 추천하지 못하고 개인화된 추천서비스의 유용성을 떨어뜨린다.

2.2 콜드 스타트 문제(Cold Start Problem)

콜드 스타트 문제는(Cold Start Problem) 초기 평가 점수의 부족이 사용자에게 유의미한 추천을 하지 못하는 상황을 말한다.

대표적으로 두 가지 형태로 문제가 발생하는데 이 문제는 결국 협업 필터링 시스템의 추천 성능 저하를 야기한다.

1) 새로운 사용자 문제

새로운 사용자가 추천시스템을 처음 가입하거나 사용할 때 아이템에 대한 평가 점수 기록이 존재하지 않는다. 따라서 협업 필터링 과정에서 자신과 유사한 이웃(neighborhood)를 찾는 것 자체가 불가능하고 따라서 개인화된 추천을 할 수 없다.

2) 새로운 아이템 문제

협업 필터링 추천 시스템에 새로운 아이템이 추가될 경우 새로운 아이템은 평가점수를 가지지 못한다. 따라서 새로운 아이템이 적절한 수에 사용자들에 의해서 평가될 때까지 그 아이템을

추천해 줄 수 없다.

2.3 사용자 평가 점수 희소성 (User Rating Sparsity)

사용자 평가 점수 희소성은 앞에서 다룬 차가운 시작 문제는 (Cold Start Problem)의 연장선으로 볼 수 있다.

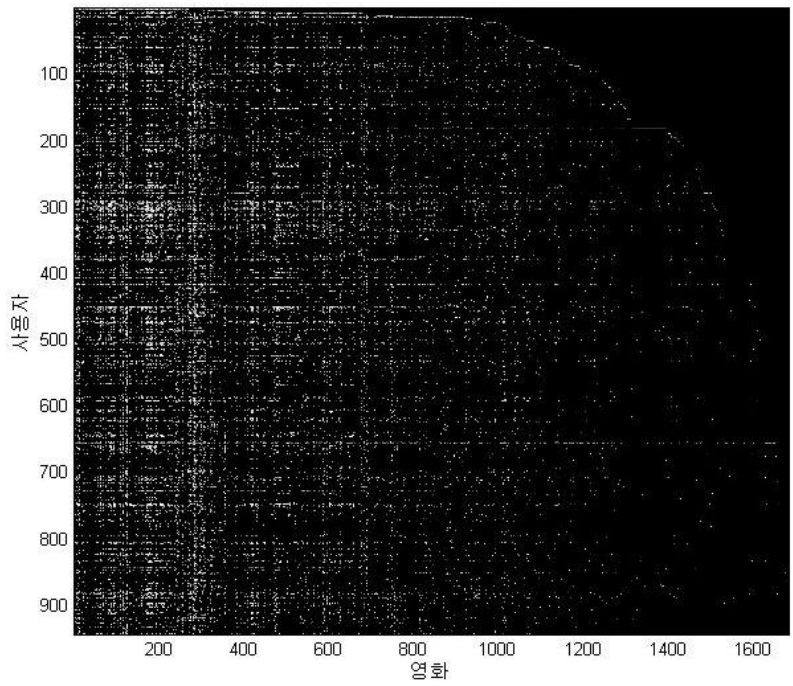
사용자는 전체 아이템 중에 평가한 점수보다 평가하지 않은 아이템의 수가 훨씬 많다. 그리고 사용자 마다 아이템에 평가 개수 는 상이하다. 보통 대부분의 사용자는 아이템에 대한 평가개수가 거의 없거나 소수에 불과할 것이다.

이는 몇 가지 문제를 야기하는데 첫 번째로 내용적으로 괜찮은 아이템이 추천 되지 못할 가능성이 있다. 그 이유는 해당 아이템이 대중적 인지도가 떨어져 소수의 사용자들에 의해서만 평가가 이루어지면 다른 인기 있는 아이템에 비해 낮은 상관 관계를 갖게 된다. 그로 인해 추천이 드물게 이루어지거나 추천되지 못할 수도 있다.

두 번째로 협업 필터링 추천 과정 중에 자신과 비슷한 사용자나 아이템을 발견하기 위해 각각의 유사도를 구하는 과정이 존재한다.

이 때 사용자와 사용자 사이의 유사도는 각 사용자들에 의해서 평가된 동일한 아이템들을 기준으로 계산된다. 이때 같이 평가한 아이템들의 수가 적을 경우 사용자 평가와는 상관없이 높은 상관 관계를 가지게 된다. 이렇게 잘못된 유사도는 추천 과정에서 반영 되어 추천 성능을 떨어트리게 된다.

실제로 무비렌즈(MovieLens)의 사용자 평가 점수 희소성 (User Rating Sparsity)을 [그림 6]에서 보여 주고 있다.



[그림 6] 무비렌즈(MovieLens)에 사용자×영화 행렬

위 그림은 y축은 각각의 사용자 ID를 나타내고 x축은 각각의 영화 ID를 나타낸다. 즉 1-943명의 사용자와 1-1682개의 영화가 존재한다. 검은색은 평가 점수가 없다는 것을 나타내고 하얀색은 사용자가 평가한 점수가 존재한다는 사실을 뜻한다.

연구 목표에서도 언급하였듯이 무비렌즈(MovieLens)는 적어도 한 명의 사용자가 각각 다른 영화에 대해 적어도 20개의 평가점수를 가지고 있다고 설명했다. 따라서 실제 영화 추천 사이트에 사용자 희소성은 위 그림과 차이가 더 클 것으로 예상 된다. 사용자 평가 점수의 희소성은 다음과 같은 방식으로 계산된다.

$$\text{희소성} = \left(1 - \frac{\text{실제 평점수}}{\text{영화수} \times \text{사용자수}}\right) \times 100$$

$$= \left(1 - \frac{100{,}000}{1682 \times 943}\right) \times 100 \approx 93.6953\%$$

결과적으로 무비렌즈(MovieLens) 사용자 평가 점수의 희소성은 93.7%이다.

3. 사용자 데이터 희소성에 집중한 영화 추천

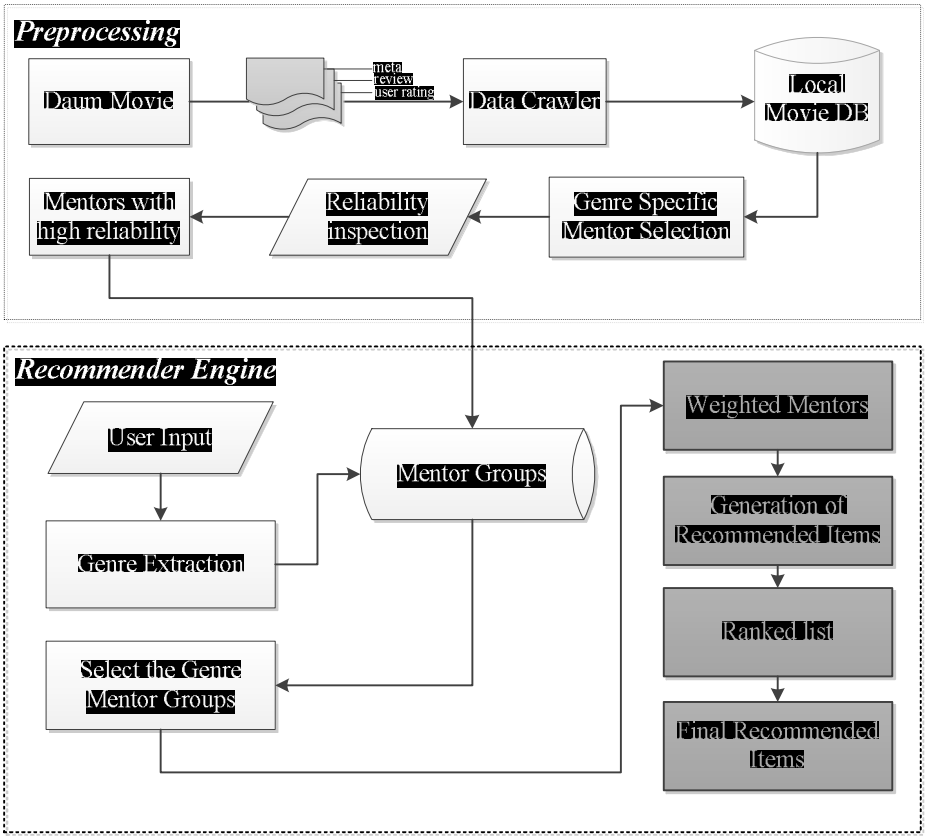
사용자 데이터 희소성 문제를 해결하기 위해 다양한 접근들이 이루어졌다. 그 중 Billsus [22]는 특이값 분해(Singular Value Decomposition)를 사용하여 사용자×아이템 행렬의 차원을 줄이기 위해 중요하지 않거나 대표적이지 않은 사용자나 아이템을 제거함으로써 희소성 문제를 해결 하려 했다. Melville[23]은 희박한(sparse) 정보를 가진 사용자×영화 행렬 정보의 문제점을 해결하고자 내용기반(content-based) 방법을 적용하여 값이 없는 평점 정보를 예측하여 전체 행렬를 구성하였다. 이 전체 행렬을 바탕으로 협업 필터링을 적용하였다. Choi [19]는 전통적인 협업 필터링 기반은 사용자 선호 유사도를 구하기 위해서 많은 작업이 요구된다는 것을 지적했다. 따라서 전통적인 협업 필터링의 문제점인 희소성(Sparsity)과 초기시행자문제(Cold Start Problem)를 해결하기 위해 특정한 장르를 선호하는 사용자에게 카테고리 상관관계 기반의 협업 필터링 접근을 제안했다.

사용자 데이터 희소성을 해결하기 위해 다양한 접근이 이루어졌지만 각각의 문제점 또한 발견되었다. 우선 Billsus [22]는 특정한 사용자나 아이템을 제거함으로써 추천에 필요한 유용한 정보를 잃게 되어 추천의 질을 떨어뜨릴 수 있다. 그리고 Melville [23]은 기본적으로 협업 필터링의 장점인 아이템의 내용정보를 파악하지 않아도 된다는 이점을 무너트린다. Choi [19]는 영화적 특성 중에 장르와 전체 평점을 사용했다는 점이 새롭고 참신한 영화를 기대하기 어려울 것으로 보인다.

하지만 이 연구에서 제안하는 멘토 기반의 영화 추천 알고리즘은 기본적으로 내용기반분석이 필요 없는 협업 필터링의 장점을 취하고 또한 사용자중에서 멘토를 선정하기 때문에 추가적인 제약조건이 필요 없다. 그리고 멘토는 장르를 기준으로 선정되기 때문에 해당 장르에 참신하고 새로운 영화를 추천 받을 수 있다. 이는 기존의 협업 필터링을 사용하여 평점 개수가 적은 사용자에게 유의미한 아이템을 추천해줄 수 없는 것과 비교했을 때 우수성이 발견되며 따라서 이 연구에서는 멘토 기반의 영화 추천시스템을 제안하고자 한다.

제 3 장 멘토기반의 영화추천 시스템

제 1 절 시스템 구성



[그림 7] 멘토기반 영화 추천 시스템 구성도

이 연구에서 제안하는 멘토(Mentor) 기반의 영화 추천시스템에 전체 구성도는 [그림 7]의 형태와 같다. 먼저 다음(Daum) 포털 사이트에서 영화 정보-영화 메타(meta) 데이터, 사용자 평점 데이터, 사용자 리뷰데이터를 데이터 크롤러(Crawler)를 이용하여 로컬 데이터베이스의 각각에 테이블에 저장한다.

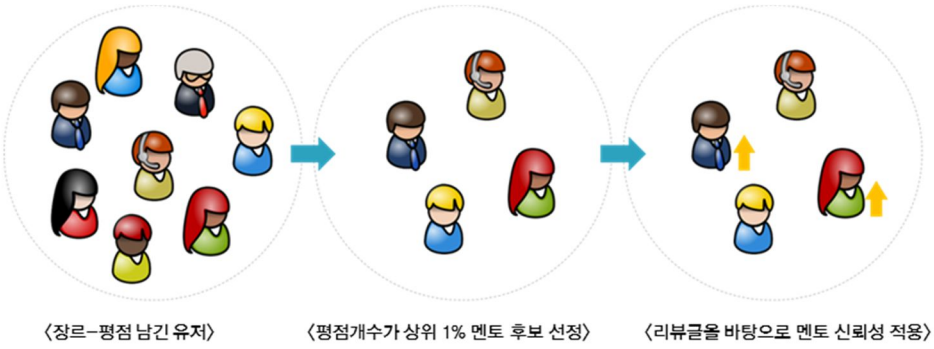
3개의 테이블 중 사용자 정보 테이블에서 각 장르별 조건에 부합하는 멘토를 선정 한다. 여기까지가 시스템의 전처리 단계로 희박한(Sparse) 평점 정보를 가진 사용자(추천 받을 사용자)에 데이터 입력을 기다린다.

입력 데이터는 사용자가 평점을 남긴 영화가 사용되며 해당 사용자의 영화에서 장르 정보를 추출하여 장르별 멘토 그룹을 선택하게

된다. 이렇게 선택된 멘토 그룹안에서도 사용자가 남긴 평점정보와 비교하여 비슷한 성향을 가진 멘토 그룹들만 재선정하게 된다.

최종적으로 선정된 멘토들은 사용자와의 유사도 정도와 신뢰성을 근거로 가중치가 부여되며 이 가중치는 멘토×영화 아이템 행렬에 적용된다. 이 행렬을 바탕으로 영화추천 리스트가 생성되고, 단일 장르일 경우 이 결과를 사용자에게 돌려주고 복수 장르일 경우 장르마다 추천된 리스트들을 병합하여 사용자에게 보여주게 된다.

제 2 절 멘토(mentor)의 정의



[그림 8] 멘토 선정 과정

[표 3] 멘토의 정의

멘토(Mentor)의 사전적 정의
1) 경험 없는 사람에게 오랜 기간에 걸쳐 조언과 도움을 베풀어 주는 유 경험자・선배 [27]
2) 현명하고 신뢰할 수 있는 상담 상대, 지도자, 스승, 선생의 의미로 쓰이는 말 [28]

영화의 멘토(Movie’s Mentor)
1) 유 경험자 (영화를 많이 본 유저 = 평점을 많이 남긴 유저)
2) 현명하고 신뢰할 수 있는 사람 : 리뷰글(추천수+댓글수+조회수)
본 논문에서는 1의 정의를 만족하고 2의 조건에 의해서 발생되는 신뢰성을 토대로 가중치가 적용된 멘토를 사용한다.

멘토의 사전적 정의를 찾아보면 대표적으로 두 가지로 요약할 수 있다. 위의 두 가지 정의를 바탕으로 영화의 멘토를 정의해 보았다.

첫 번째로 이야기하는 유 경험자 및 선배에 대한 정의는 영화 속 멘토로 바꾸어 말하면 얼마나 많은 영화를 본 경험이 있는가로 표현될 수 있다. 영화를 많이 본 유저는 영화에 평점정보를 몇 개 남겼는가로 확인 할 수 있다.

따라서 영화의 멘토에 첫 번째 정의는 각 장르마다 평점정보를 많인 남긴 상위 유저 1%를 기준으로 선정하였다. 다음은 각 장르마다 평점을 남긴 사용자와 평점을 많이 남긴 상위 1%에 해당하는 멘토의 수이다.

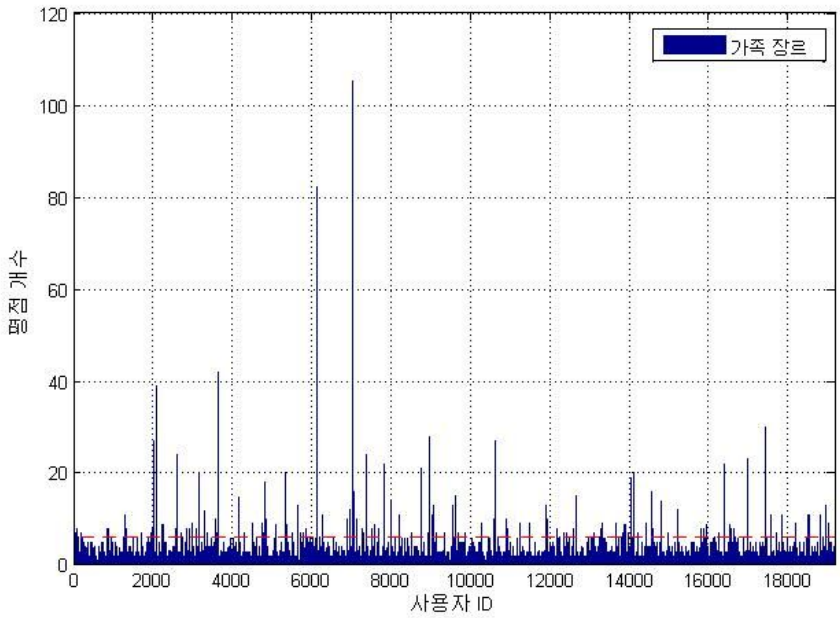
[표 4] 영화 정보와 평점을 남긴 사용자 및 멘토 수

장르	영화수	평점개수	사용자 수	멘토수
가족	1,235	24,480	19,152	191
액션	5,005	185,389	91,742	917
공포	2,836	38,822	23,218	232
스릴러	3,959	105,781	57,133	571
코미디	8,839	148,247	78,300	783
서부	406	4,182	3,906	39
무협	179	3,460	2,877	28
드라마	19,081	260,848	126,107	1,261
로맨스/멜로	4,604	118,112	65,879	658
SF	1,783	65,113	42,960	429
판타지	1,635	59,305	39,218	392
미스터리	1,009	27,480	19,755	197
어드벤처	1,984	78,469	48,101	481
애니메이션	4,000	34,544	22,666	226
성인	651	962	805	8
전쟁	1,022	22,240	16,903	169
다큐	4,733	9,877	8,417	84
뮤지컬	557	5,244	4,509	45
시대극	538	21,489	18,389	183
범죄	2,049	44,094	28,519	285

[표 4]의 영화 장르는 총 20종류로[②] 분류된다. 영화 수는 각 장르 마다 영화의 개수를 나타내며, 복수 장르일 경우 중복 적용된다. 평점개수는 각 장르에 평점을 남긴 개수를 뜻하며 사용자수는 장르마다 평점을 남긴 실제 사용자 수를 뜻한다. 마지막으로 멘토 수는 평점 개수를 기준으로 상위 1%에 해당하는 사용자를 뜻한다. 예를 들어 가족장르를 기준으로 설명하면, 가족 영화는 총 1,235개로 이에 해당하는 영화에 평점을 남긴 개수는 24,480개 존재한다. 이 평점을 남긴 사용자는 19,152명이 존재한다. 다음 [그림 9] 에서 이 내용을 확인해보면 x축은 19,152명에 해당하는 각각의 사용자를 뜻하며 Y축은 평점의 개수를 나타낸다. 각각의 사용자가 다양한 평점 개수를 가지는 것을 확인 할 수 있다. 이 중에서 멘토는 평점 개수가 상위 1%로 [그림 9]에서 붉은 점선 위에

② 다음(Daum) 포털 사이트 영화 정보를 기준으로 20개의 장르로 분류

평점개수를 가진 사용자가 멘토에 해당하고 가족장르의 총 멘토수는 191명이 존재한다.

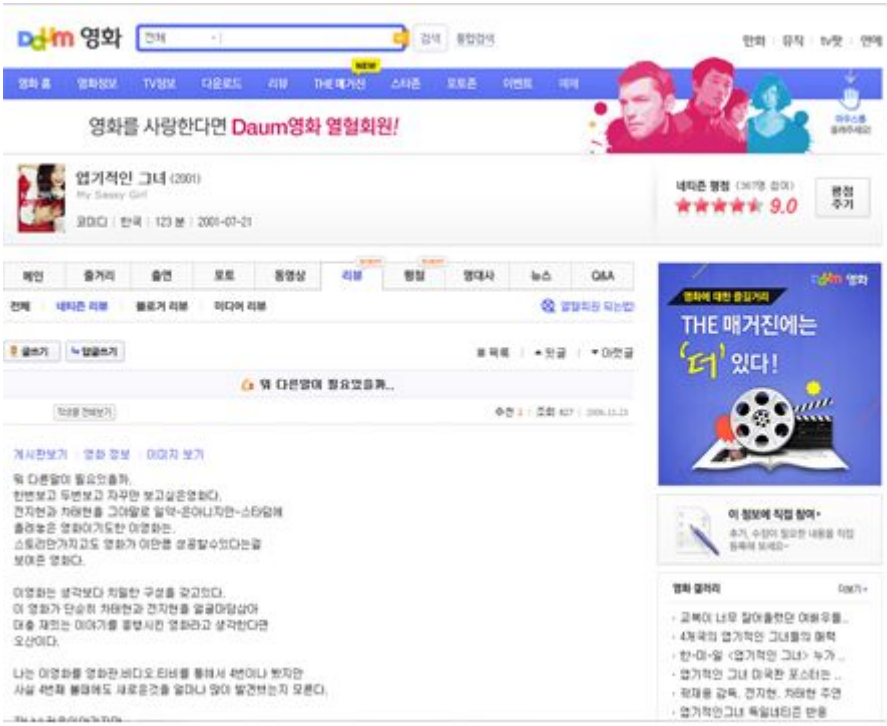


[그림 9] 가족장르 사용자ID에 따른 평점개수

두 번째로 멘토는 유 경험자와 더불어 현명하고 신뢰할 수 있어야 한다고 정의 되어 있는데, 영화 멘토에게 있어서 현명하고 신뢰도를 판단 할 수 있는 근거는 리뷰글을 들 수 있다. 리뷰글은 [그림 10]와 같이 평점을 남길 때 쓰는 작은 코멘트와 다르게 영화에 대한 자신의 생각 및 감상에 대해 따로 작성하게 되어 있고, 해당 글은 다른 사용자들에게 공유될 수 있다.

리뷰글을 바탕으로 멘토가 남긴 리뷰글의 개수와 얼마나 많은 사용자들이 그 내용을 읽었고, 어떠한 반응을 보였는지, 마지막으로 얼마나 많은 호응을 받았는지를 각각 조회수, 댓글수, 추천수로 판단하게 된다.

4가지 조건의 데이터를 바탕으로 각 장르마다 선정된 멘토들은 신뢰도가 주어지게 되고, 영화 추천과정에서 각 멘토 들마다의 경중이 달라지게 되어 추천 아이템의 영향력이 달라지게 된다. 다시 정리하면 신뢰도가 높은 멘토란 리뷰글의 개수가 많고 해당 글에 대한 댓글, 추천, 조회수가 많은 멘토를 뜻하며 그렇지 않은 멘토에 비해 더 높은 신뢰도를 적용 받게 된다.



[그림 10] '엽기적인 그녀' 영화를 본 사용자의 리뷰글

[표 5] 전체 리뷰 데이터

	리뷰글	조회수	추천수	댓글수
총 개수	155,060	113,085,145	150,608	218,634
평균		729.2	0.97	1.41
1/평균		0.0013 (C_c)	1.0309 (C_r)	0.7092 (C_{rp})

[표 5]는 전체 리뷰데이터의 정보를 나타내며 리뷰글의 총 개수는 155,060개가 존재한다. 조회수는 전체 113,085,145번으로 리뷰글 1개당 평균 730번에 조회가 이루어졌다. 총 추천수는 150,608로 1개의 리뷰글에 평균 0.97번 추천되었다. 댓글수는 218,634개로 1개의 글에 평균 1.41 개 정도의 댓글이 달리는 것으로 확인되었다.

$$\mathrm{R}(M) = C_c \sum_{k=1}^{K} N_{k,c} + C_r \sum_{k=1}^{K} N_{k,r} + C_{rp} \sum_{k=1}^{K} N_{k,rp} \quad (3)$$

여기서 R(M)은 멘토의 신뢰성을 나타내며, K는 멘토가 남긴 리뷰글의 수를 나타낸다. $N_{k,c}$ 는 멘토가 남긴 k번째 리뷰글의 조회수를 나타내며 $N_{k,r}, N_{k,rp}$ 도 각각 추천수와 댓글수를 나타낸다.

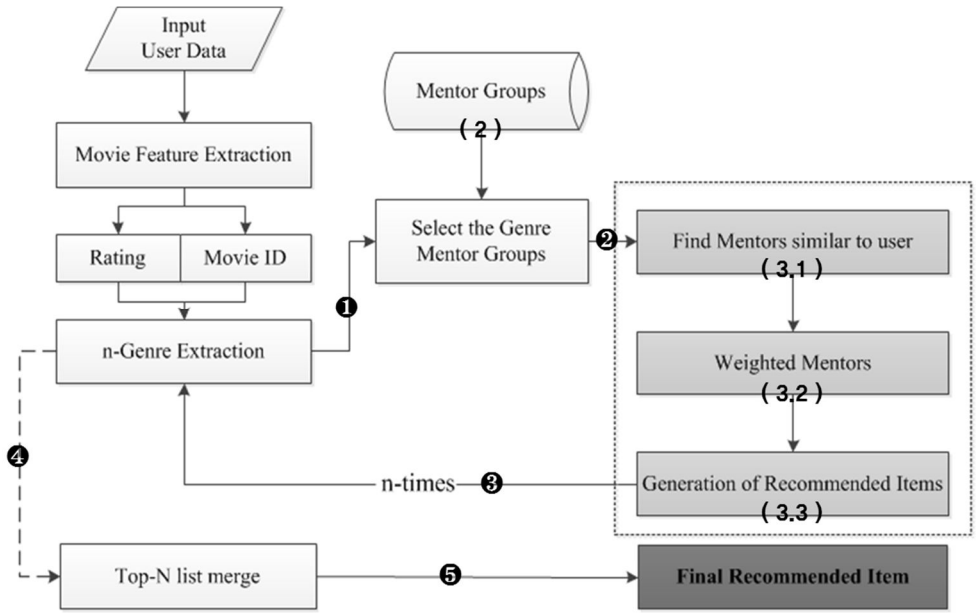
(3)을 바탕으로 실제 가족장르에 해당하는 멘토의 신뢰성을 계산한 결과는 다음 [표 6]와 같다

[표 6] 가족 장르 멘토의 리뷰글 정보

멘토ID	평점 개수	리뷰글 개수	댓글수	추천수	조회수	멘토의 신뢰성
날아*	105	12	11	21	6820	38.7
양쿠*	82	1	0	0	753	1.0
이소*	42	1	1	7	4667	14.3
영**	39	85	56	54	34811	143.0
아이*	30	4	0	0	129	0.1
구이*	28	542	562	212	840868	1769.0
갈바*	27	0	0	0	0	0.0
어릿광*	27	0	0	0	0	0.0
지**	24	148	18	50	23720	96.8
요여*	24	0	0	0	0	0.0
⋮						

[표 6]에서 처럼 멘토ID 날아*는 평점개수를 가장 많이 남긴 멘토임에도 불구하고 리뷰글의 개수는 12개만을 작성하였고 해당 글의 댓글수는 11개, 추천수는 21개, 조회수는 6820으로 멘토의 신뢰성이 38.7이 계산된다. 이와는 다르게 멘토ID 구이*는 평점개수가 28개인 반면에 리뷰글의 개수는 542개로 평점개수보다 약 19배 차이가 난다. 그 결과 멘토의 신뢰성은 날아* 보다 높은 1769.0이 측정되었다.

제 3 절 멘토 기반의 영화 추천 알고리즘



[그림 11] 멘토 기반의 영화 추천 알고리즘

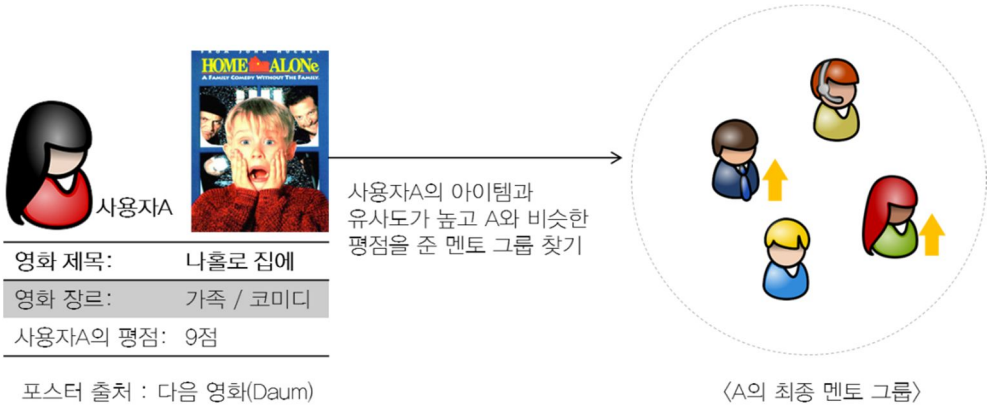
멘토 기반의 영화 추천알고리즘은 [그림 11]와 같다
먼저 추천을 받을 희박한 평점 정보를 가진 사용자에게서 영화와 평점 정보를 추출한다. 추출된 영화와 로컬 데이터베이스에 저장되어있는 영화 메타정보를 비교하여 장르 정보를 추출한다. 추출된 장르 정보를 가지고 전처리 과정에서 수행했던 장르 멘토 그룹들을 선택하게 된다.

선택된 장르 멘토 그룹에서 사용자에게 유사한 멘토 들을 다시 찾고, 가중치를 부여 한 뒤 추천 아이템 리스트를 생성한다. n($n \geq 2$) 장르 인 경우 해당 과정을 n번 반복하여 각 장르마다의 추천리스트를 생성하고 최종적으로 계산된 값을 통해 추천리스트들을 병합 정렬한다.

최종적으로 병합 정렬된 최종 리스트를 사용자에게 추천해준다.

1. 사용자에게 적합한 멘토 그룹 찾기

장르를 기준으로 선택된 멘토 그룹과 사용자의 영화, 평점 정보를 비교하여 장르 멘토 그룹 안에서 사용자에게 적합한 멘토들을 찾게 된다.



[그림 12] 사용자A에 적합한 멘토 그룹 찾기

사용자에게 적합한 멘토를 찾기 위해서 아이템 유사도 행렬이 필요하다. 서론에서도 언급하였듯이 전체 사용자의 평가점수는 극도로 희박하다. 대다수의 사용자가 대부분의 영화에 평점을 남기지 않았기 때문에 유사도를 구하게 되면 유사한 것과 유사하지 않은 것에 차이가 거의 존재하지 않게 된다. 따라서 우리는 전체 다수의 평점을 차지하고 있는 멘토의 사용자×아이템 평점 행렬을 기준으로 아이템 유사도를 구하려고 한다.

1.1 멘토 아이템 유사도 행렬의 SVD 적용

멘토의 사용자×아이템 행렬 또한 평점정보가 희박하기 때문에 행렬 인수분해 중 SVD를 사용하여 비어있는 평점정보를 채운다.

SVD는 Singular Value Decomposition의 약어로 직사각형의 행렬을 분해하는 방법을 뜻한다. 임의의 m × n의 행렬 M이 있다고 가정하면 다음과 같은식으로 SVD를 표현할 수 있다.

$$M = USV^T$$

U는 각 열이 직교(orthogonal)하면서 법선(normal)인 m × m의

유니터리 행렬(unitary matrix)이고 S는 m × n 대각행렬(diagonal matrix)이다. 마지막으로 V는 n × n의 유니터리 행렬 (unitary matrix)이다.

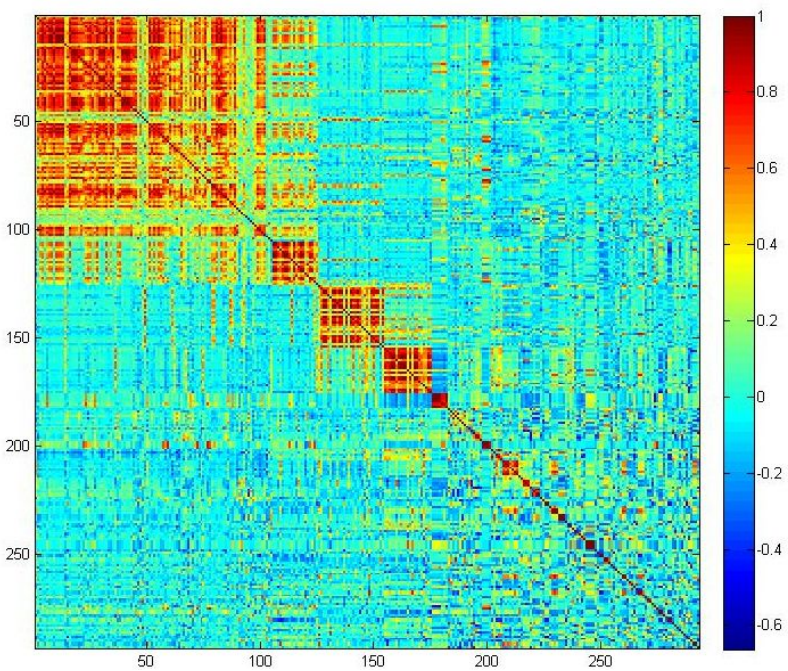
S의 대각선 값들을 특이 값이라고 하고 분해 된 행렬을 재구성할 때 몇 개의 특이 값 k을 사용하는지에 따라 실제 M과 유사해진다. 이를 통해서 한 사용자가 다수의 영화의 평점을 남겼을 때 SVD를 통하여 k차원의 벡터로 축소 할 수 있다.

이 연구에서는 k값을 20으로 설정하였고 피어슨 상관계수를 이용하여 최종적으로 아이템 간의 유사도를 구하였다.

1.2 멘토 찾기

사용자와 멘토들 사이의 유사 정도 차이를 바탕으로 사용자에게 적합한 멘토를 찾는다.

[그림 13]은 실제 가족 장르 영화 아이템의 유사 도로써 X축, Y축은 영화 아이템을 뜻하고 붉은색으로 갈수록 유사도가 높음을 나타낸다

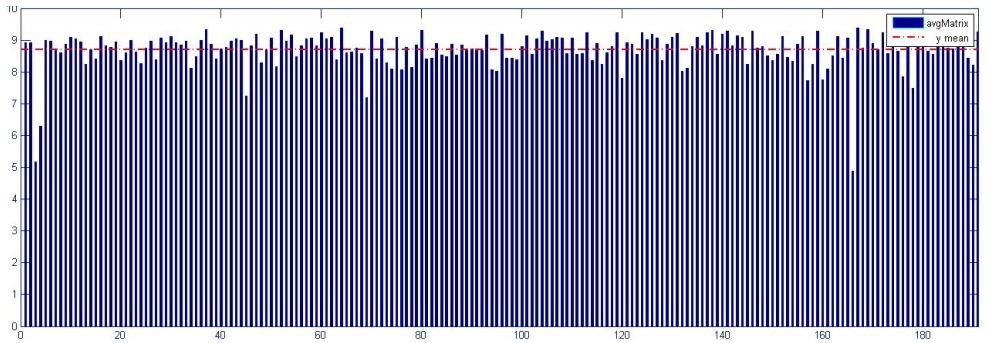


[그림 13] 가족 장르의 영화 아이템 유사도

이 유사도를 바탕으로 사용자의 영화 아이템과 평점을 멘토

아이템들의 평점들과 비교하여 차이 값을 구한다. 이때 차이값이 적을수록 사용자의 영화와 취향이 비슷한 영화에 비슷한 평점을 부여했다는 의미이고 이는 사용자와 멘토 사이의 유사 정도가 가깝다고 말할 수 있다. 이렇게 멘토들 마다 계산된 차이 값을 기준으로 전체 평균을 구하고, 평균 이하의 값을 가진 멘토들을 최종적으로 유저에게 적합한 멘토로 선정한다

실제로 [그림14]는 사용자A와 멘토들 사이의 유사 정도의 차이를 보여주며 붉은선은 멘토 그룹의 평균을 가리킨다. 붉은선 아래에 해당하는 멘토들이 유저 A에게 적합한 멘토로 선정되었다.



[그림 14] 가족 장르 멘토들과 유저 A사이의 거리

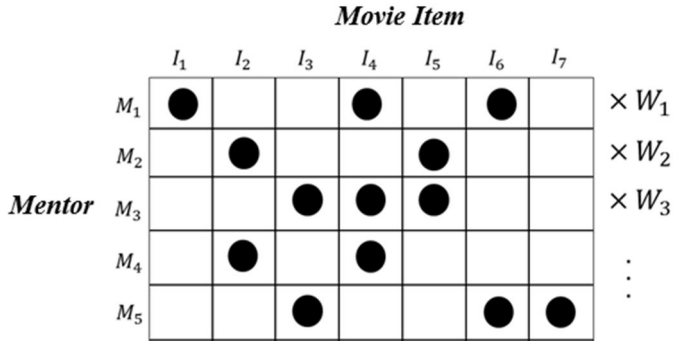
2. 멘토 가중치 부여

멘토마다 추천과정에서의 중요도 차이를 두기 위해서 멘토별 가중치를 부여한다. 가중치는 기존의 계산되었던 멘토의 신뢰도와 함께 사용자A사이의 거리를 기준으로 가중치가 부여된다.

멘토의 신뢰도는 전체 최대값을 기준으로 0.5-1범위로 정규화 하고 멘토와 유저A사이의 거리는 값이 작을수록 큰 값이 되도록 0-0.5로 정규화하였다. 두 값을 더하여 최종 가중치를 구하였다.

영향력 있는 멘토는 신뢰도가 높을수록 또 한 유저와의 거리가 가까울수록 높은 가중치를 적용 받는다.

3. 최종 선정된 멘토와 영화 아이템 추천 과정



[그림 15] 최종 멘토와 영화 아이템 행렬

[그림15]와 같이 최종 선정된 멘토들과 멘토들의 전체 아이템 행렬이 위와 같이 생성된다. 이 행렬에 앞에서 언급한 가중치가 멘토 각각에게 적용된다. 다음으로 이 행렬에 전체 평균값을 구한 뒤에 행렬의 성분에 평균을 빼준다.

이때 각 행렬 성분에 평균값을 빼준 이유는 영화 추천과정에서 멘토별 가중치 적용으로 인해 특정 멘토만의 편향된 추천을 방지하고 사용자에게 적합한 멘토 그룹안에서 공통 분모의 영화 또한 중요한 정보로 간주하기 위함이다.

최종적으로 영화 아이템 열마다의 합을 구하고 결과값이 가장 큰 순서부터 Top-N 리스트를 생성 하게 된다. 단일 장르의 경우에는 이 리스트가 최종 추천 결과로 사용자에게 반환되고 복수장르의 경우에는 장르마다 같은 알고리즘을 적용하여 최종 리스트의 결과를 값에 따라 병합 한 뒤 정렬하여 추천결과를 보여 주게 된다

제 4 장 시스템 성능 평가

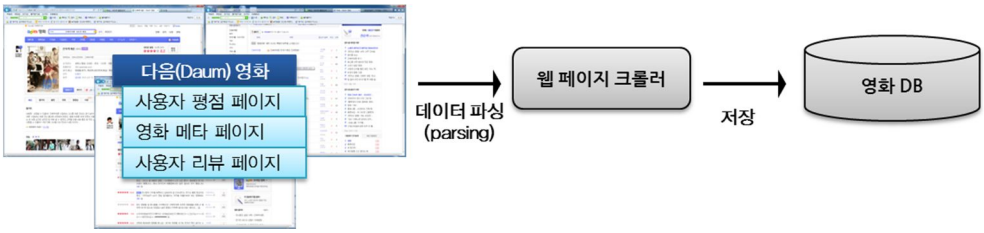
제 1 절 데이터 셋(Data Set)

알고리즘을 설계하고 테스트 및 평가 하기 위해서 영화 데이터와 사용자 데이터가 필요하다. 하지만 멘토(mentor)기반의 알고리즘은 기존의 무비렌즈(MovieLens)와 같은 데이터 집합(Data Set)을 사용할 수 없기 때문에 새로이 영화 데이터를 수집하였다.

수집된 영화는 국내 유명 포털 사이트 중에 하나인 다음(Daum) 영화 데이터로 각종 영화 데이터에 대한 접근 제한이 없어 선정하게 되었다.

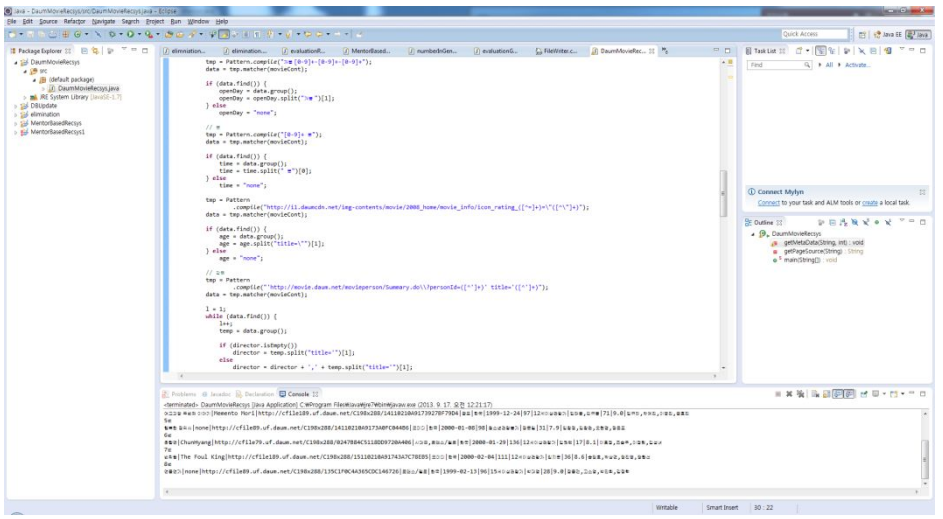
1. 다음 영화 데이터 수집(Daum Movie Data Crawler)

영화 데이터 수집은 [그림 16]와 같은 절차로 진행 하였다.



[그림 16] 영화 데이터 수집 과정

다음(Daum) 영화 API를 사용하여 데이터를 수집하려고 했으나 원하는 정보 모두를 갖고 올 수 없고 일일 접근 횟수에 대한 제한으로 웹 페이지 크롤러(Crawler)를 직접 제작하여 데이터를 수집하였다. 프로그래밍 언어는 자바(Java) 기반으로 인터페이스는 이클립스(eclipse ver.Kepler)를 사용하여 제작하였다.



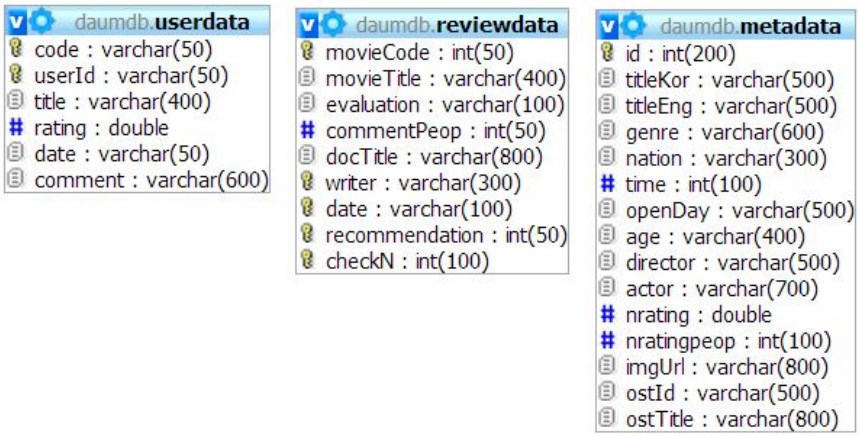
[그림 17] 영화 데이터 수집

웹 페이지 크롤러에 대해 간단히 설명하면 사용자 평점, 영화 메타데이터, 사용자 리뷰 각각의 페이지에 각각 순차적으로 접근하여 웹페이지를 가져온다. 가져온 웹페이지 중에서 필요한 정보를 정규표현식(Regular Express)를 사용하여 접근한 뒤 해당부분만 잘라내기(Parsing)를 통해 가져오게 제작하였다.

[그림 17]처럼 실제 웹 상에 데이터를 가져와 local DB에 저장하고 있다.

2. 다음(Daum) 영화 데이터 베이스

멘토(mentor)기반의 알고리즘에 필요한 데이터는 총 3가지로 사용자 평점 데이터, 사용자 리뷰 데이터, 영화 메타데이터이다. 따라서 이에 맞는 영화 데이터 베이스를 아래 [그림 18]과 같이 구축하였다.



[그림 18] 영화 데이터베이스 테이블

총 3개의 테이블로 데이터베이스를 구성하였고 각 테이블 속성값에 대한 설명은 아래 표와 같다.

[표 7] 사용자 평점정보 테이블

사용자 (userdata)	
code(PK)	다음(Daum) 영화 페이지번호
userId	사용자 ID
Title	영화 제목
Rating	평가 점수
Date	데이터 날짜
Comment	사용자의 논평

[표 8] 영화 메타데이터 테이블

영화 메타데이터 (metadata)	
Id	다음(Daum) 영화 페이지번호
titleKor	영화 한글 제목
titleEng	영화 영어 제목
Genre	장르
Nation	나라
Time	영화 시간
openDay	개봉 날짜
Age	관람가능 나이
Director	감독
Actor	출연배우
Nrating	네티즌 전체 평점
Nratingpeop	네티즌 참여 수
Imgurl	포스터 URL
ostId	OST 페이지 번호
ostTitle	OST 제목

[표 9] 리뷰데이터 테이블

리뷰 데이터 (reviewdata)	
Movieid	다음(Daum) 영화 페이지번호
movieTitle	영화 제목
Evaluation	3가지 평가
commentPeop	댓글 수
docTitle	리뷰 글 제목
Writer	글쓴이
Date	날짜
recommendation	추천수
Check	조회수

웹 페이지 크롤러(Crawler)를 이용하여 총 10일에 걸쳐 전체 데이터(2012. 3.12 기준)를 수집하였다. 수집 결과는 아래 표와 같다.

[표 10] 수집된 데이터의 수

데이터 테이블	레코드 수
영화 메타 데이터 (metadata)	54,021
리뷰 데이터 (reviewdata)	155,060
사용자 데이터 (userdata)	732,046

54,021편의 영화 정보가 수집되었고 사용자 리뷰글은 155,060편의 게시글을 가져왔다. 마지막으로 사용자 평점 데이터는 732,046개의 데이터를 수집하였다.

평점을 남긴 사용자는 241,704명으로 평균 3.02개의 평점을 남기는 것으로 나타났다. 아래 [그림 19]은 실제 평점 데이터 테이블에 저장된 화면이다.

	code	userId	title	rating	date	comment
	64326		화차	8	2012.03.10	탄탄한 연기력, 스무스한 스토리 구성, 순간의 코믹, 박진감 넘치는 미스터리 까지...최고!!!
	64326		화차	9	2012.03.10	김민희의 재발견.. 그리고 이선균의 생활연기 너무 좋았다.. 에스컬레이트위에서 이선균과의...
	44834		세븐 파운즈	10	2012.03.10	이걸 뭐선이나 자기만족이라고 하는 사람도 있는데, 심장이 약한 사람부의 그 내용 한 명의 수...
	60331		파이널 데스티네이션 5	7	2012.03.10	이영화는 원래 보고 기분좋아지는 영화많은 거리가 있는 영화이다. 그냥 사람죽는 장면들을 심...
	64406		원더풀 라디오	1	2012.03.10	재미를 느껴야하나 감동을 느껴야하나 뭐 생각할게 있어야 하나..
	47381		존 카터 : 바숨 전쟁의 시작	8	2012.03.10	전개가 빠르고 내용도 그만하면~~~~^^*
	42375		말암	9	2012.03.10	전도연 또앤
	44392		댓진 하루	9	2012.03.10	별 사업장은 이야기가 무슨 몰입이 이렇게 잘되나
	40309		소년, 천국에 가다	8	2012.03.10	이런 판타지가 난 재밌더라 엄정아 아줌마 이뻐~
	36657		오그빌	9	2012.03.10	독특한 형식,이런 '영화'도 참됐다.탄탄한 차입새와 연기를 지루하지도 않고.
	64326		화차	8	2012.03.10	할인는 대본.
	1902		씨익스피어 인 러브	8	2012.03.10	목롱삼삼력 호감하는 눈
	11367		알프레도 가르시아의 목을 가져 와라	9	2012.03.10	한마디로 정말 독특하다고 밖에는 말할수 없는 작품으로 완성도를 따나 폴킨마말고는 그누구도...
	37592		마지막 한 걸음까지	8	2012.03.10	난 이런 배고픈 영화가 좋더라..
	2135		풍겨사리	10	2012.03.10	저예산 영화라는 것이 믿기지 않을 정도로 자연스러운 특수효과와 연출이 일미다.
	2361		화양연화	7	2012.03.10	감명받지만 재미가 없어..
	31272		더 포드	8	2012.03.10	그래 지구가 멸망한다면 저렇게 차분히 죽어가는거야..
	60324		더 그레이	5	2012.03.10	왜 극중 캐릭터만 무섭하고 보는 나는 안무섭게 만드냐..
	51185		찾구는 못말려 : 오타케베! 카스카베 이생탐국	10	2012.03.10	정말 감동적이고.. 울면 된네요.. 재미있어요!!
	47381		존 카터 : 바숨 전쟁의 시작	3	2012.03.10	평점이 왜 이렇게 높은거... 잘 보긴 했는데 내용이 너무 뻔하고 아동적했어요... 아바타...
	59237		디스 민즈 워	10	2012.03.10	너무재밌어요!!!!!!기대안하고봤다지만, 캐리비안해적 이후로 콘소리로 웃으며 봤네요! 불...
	61883		마릴린 먼로와 함께한 일주일	9	2012.03.10	<철의 여인>에 완벽한 연기를 선보인 메릴 스트립이 있었다면 <마릴린 먼로...
	64473		아티스트	9	2012.03.10	배우의 표정과 영화의 배경음악에 이토록 몰입했던 적이 있었던가!! 2012년 현재, 흑백무...
	50180		러브픽션	5	2012.03.10	기대한것보다별로였어여 ...ㅠㅠ
	64406		원더풀 라디오	8	2012.03.10	재미만 있고만 ㅇㅇ 하는 님들이 왜케 많냐...
	50180		러브픽션	10	2012.03.10	아이디어에 찬사를 보내고, 연애에 대해 한번쯤 다시 생각하게 하고, 맥주소

[그림 19] 실제 평점데이터(userdata) 테이블

제 2 절 시스템 평가방법

추천 시스템을 평가하기 위해서 세가지 방법의 추천시스템을 비교 분석하기로 했다. 첫 번째는 추천시스템에서 가장 많이 사용되고 있는 아이템 기반의 협업 필터링 방식이다. 두 번째는 이 연구에서 제안하는 멘토 기반의 추천시스템이다. 마지막으로 랜덤 방식의 추천시스템이다. 우리는 동일한 평가 데이터 셋을 가지고 세가지 알고리즘에 대해 정확도(Precision)와 재현율(Recall)을 기준으로 평가를 수행하였다.

1. 정확도(Precision)와 재현율(Recall)

추천시스템의 성능을 평가하는 방법은 다양한 방법이 존재한다. 그 중 가장 대표적으로 사용되는 것이 정확도(Precision)와 재현율(Recall)이다.

정확도와 재현율을 설명하기 위해서는 실제 데이터와 예측한(추천한) 데이터 사이의 관계 해석이 필요하다. 실제 사용자가 남긴 평점과 추천시스템에서 추천된 결과를 비교하면 아래 표와 같이 결과를 4가지 형태로 요약 할 수 있다.

[표 11] Reality와 Prediction의 관계

		Reality	
		Actually Good	Actually Bad
Prediction	Rated Good	True Positive (tp)	False Positive (fp)
	Rated Bad	False Negative (fn)	True Negative (tn)

$$\text{Precision} = \frac{tp}{tp + fp} = \frac{|good\ movies\ recommended|}{|all\ recommednations|} \quad (4)$$

$$\text{Recall} = \frac{tp}{tp + fn} = \frac{|good\ movies\ recommended|}{|all\ good\ movies|} \quad (5)$$

정확도(Precision)는 실제 추천시스템에서 추천된 결과들 중에서 사용자가 실제로 만족한 평점(Rated Good)을 준 아이템의 개수가 몇

개 있는지를 뜻한다. 여기서 만족한 평점의 기준은 10점 만점 중에 8점 이상을 뜻한다. 이는 대부분의 추천 시스템 연구에서 이 기준을 5점 만점에서 4점 이상으로 두고 있는 반면에 이 연구에서 수집된 데이터는 10점 척도로 존재하기 때문에 같은 비율을 유지하기 위해 이와 같이 설정하였다.

재현율(Recall)은 실제 사용자가 만족한 평점(Rated Good)을 매긴 영화들 중에서 추천된 결과에 해당 영화들이 얼마나 존재하는 지를 나타낸다.

재현율과 정확도는 추천 TOP-N개의 리스트에서 N이 커짐에 따라 정확도(Precision)는 점차 낮아지고 재현율(Recall)을 높아 지는 경향이 존재한다. 이는 정확도에 경우 N에 따라 분모가 계속 커지기 때문이고 재현율의 경우에는 추천된 결과들이 많을수록 사용자가 만족한 영화가 범위 안에 들어올 확률이 높아지기 때문이다.

이 연구에서는 평가데이터 사용자의 전체 평균 정확도(Mean Precision)와 평균 재현율(Mean Recall)을 가지고 시스템들 간의 성능 비교를 수행한다.

2. 평가를 위한 데이터 셋(Evaluation Data Set)

실제 추천시스템을 평가하기 위해서는 데이터를 학습 데이터(Learning Data)와 평가 데이터(Test Data)로 나누어 성능 평가를 하는데, 앞에서도 언급 하였듯이 데이터 셋 자체가 정형화된 데이터(예를 들어 MovieLens)가 아니라 실제 영화정보사이트에서 가져온 러프한 데이터 (rough data)이기 때문에 대부분의 데이터가 매우 희박성을 띄고 있다. 따라서 이와 같은 경우 데이터를 어떻게 나누느냐에 따라 멘토가 다르게 선정 될 수 있고 이는 결과에 직접적인 영향을 미칠 수 있기 때문에 평가를 위한 데이터 셋을 추가로 수집하였다. 2012년 3월 12일을 기준으로 사용자 평점 데이터 R1을 가지고 알고리즘을 설계하였고 연구 기간 동안 상당한 시간이 경과 했기 때문에 평점데이터 R2를 다시 수집하였다.

수집된 결과는 다음과 같다.

[표 12] 사용자 평점 데이터 R1, R2

데이터 테이블	레코드 수	데이터 수집 날짜
사용자 평점 데이터 R1	732,046	2012. 03. 12
사용자 평점 데이터 R2	845,800	2012. 11. 23

약 1년 동안 113,754건의 평점 데이터가 증가 하였다. 우리는 이 중에서 사용자 평점 데이터 R1에서 극도로 희박한 사용자 즉 평점을 1~2개 남긴 사용자를 찾고 이 사용자 중에 1년이 지난 뒤에 추가로 평점을 남긴 사용자를 찾아 입력데이터와 평가 데이터를 구성하기로 했다. 추가적으로 평점을 10개 남긴 사용자 또한 관찰함으로써 평점에 개수에 따른 각각의 시스템 성능 변화를 관찰 해보고자 했다.

입력데이터와 평가데이터 구성에 따른 제약 조건은 아래 표와 같다.

[표 13] 사용자 입력/평가 데이터 필터링 과정

1	2012년 3월에 평점을 1개 또는 2개 남긴 사용자 A
2	A의 평점개수가 2012년 3월-2012년 11월 사이에 변동이 있는 유저 B
3	평점 개수 변동이 있는 유저 B중에서 평점을 남긴 영화데이터가 2012년 3월 이후 데이터가 아닌 유저 C
본 논문에서는 3단계의 필터링 과정을 걸친 유저 C를 입력/평가데이터로 사용한다.	

위 과정 중에서 평점 데이터(R1)를 기준으로 영화를 추천해주기 때문에 그 2012년 3월 이후에 등록된 영화는 추천할 수 없다. 따라서 평가목록에서 제외되었다.

이렇게 수집된 데이터는 [표 14]와 같다.

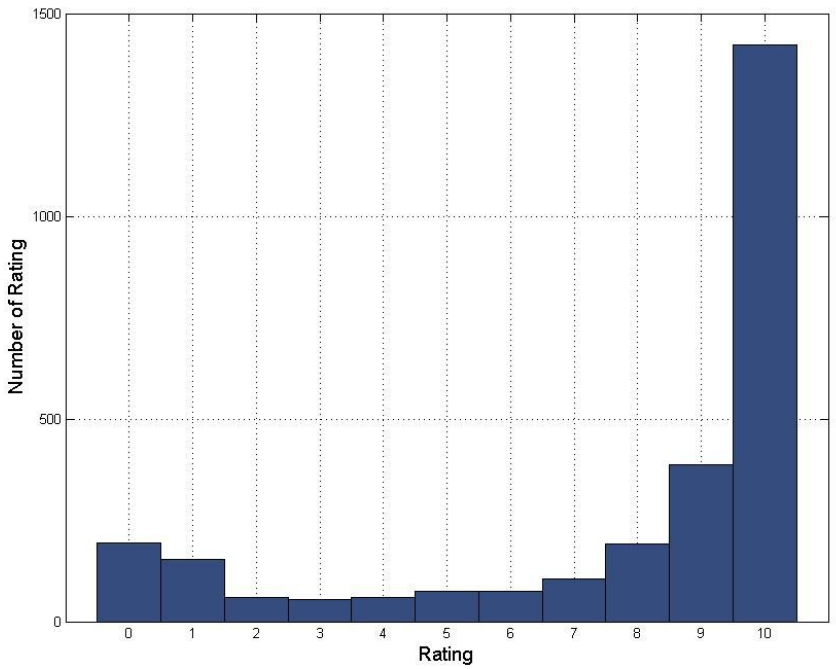
[표 14] 평점데이터 변동 현황

평점데이터 변동	평점점수 분포	사용자	입력 평점개수	평가 평점개수	평점 증가 개수(평균)
1개 -> 2개이상	8점 이상	2002명	2002개	3610개	1.8개
	8점 미만	778명	778개	1140개	1.4개
	전체	2780명	2780개	4750개	1.7개
2개 -> 3개이상		1626명	3252개	3327개	2.04개
10개 -> 11개이상		291명	2910개	5328개	18.3개

[표 14]에서 나타나듯이 평점을 1개 남긴 유저가 약 1년뒤 평점을 추가로 남긴 경우는 2780명으로 평균 1.7개 증가한 것으로 나타났다.

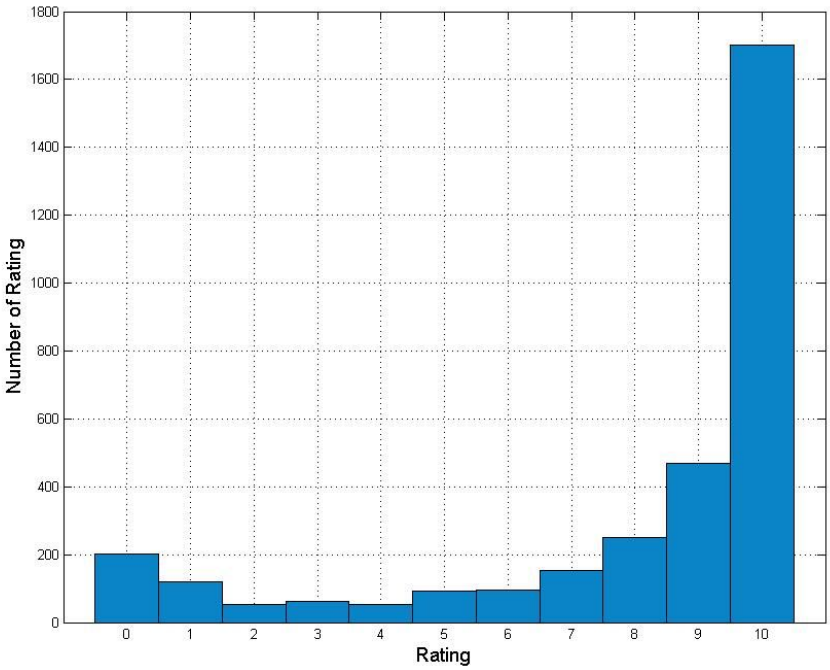
그리고 평점이 2개였던 유저는 1년뒤 평균 2.04개 증가 한 것으로 나타났고 1626명이 이에 해당한다. 마지막으로 평점개수가 10개인 유저는 평균 18.3개를 추후에 더 남겼고 291명이 존재하였다.

다음은 입력데이터의 점수 분포도 이다.



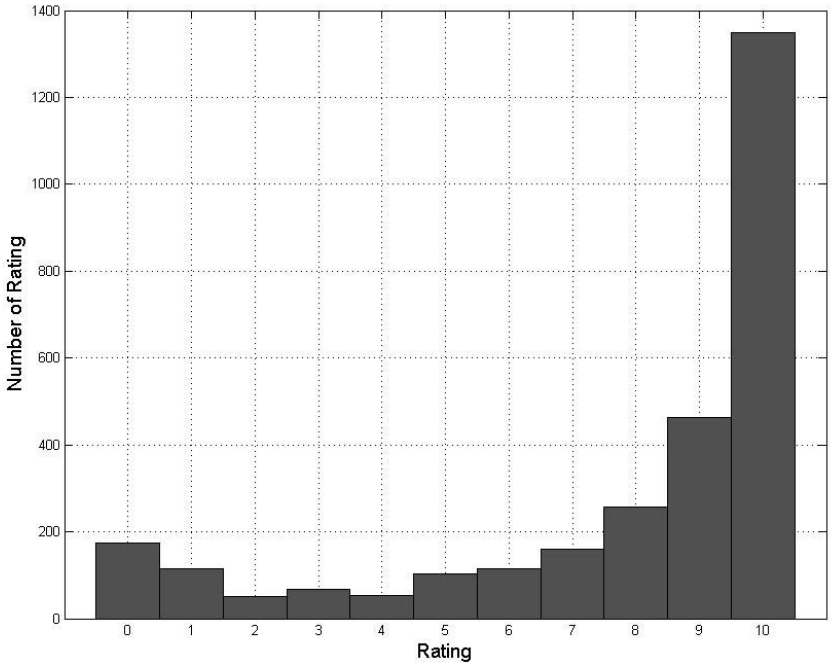
[그림 20] 1개의 평점 입력데이터의 점수 분포도

먼저 [그림 20]는 1개의 평점을 남긴 입력데이터의 점수 분포도이다. 평점을 한 개 남긴 수집된 사용자는 총 2780명이고 이중에서 8점 이상의 평점을 남긴 유저는 2002명에 해당하고 8점 미만을 남긴 유저는 778명 존재한다.



[그림 21] 2개의 평점 입력데이터의 점수 분포도

[그림 21]은 평점을 2개 남긴 사용자의 입력데이터 점수분포도로 사용자는 1626명 존재하고 입력 평점개수는 3252개이다. [그림 20]과 마찬가지로 10점의 점수 분포가 가장 높은 것으로 확인된다.



[그림 22] 10개의 평점 입력데이터의 점수 분포도

마지막으로 [그림 22]는 10개의 평점을 남긴 사용자의 입력 점수분포도로 291명이 존재하고 총 입력 평점개수는 2910개가 존재한다.

성능 분석에 앞서서 평점을 1개 남긴 사용자의 경우에는 사용자를 2개의 분류로 나누었는데, 이는 협업필터링 특성상 아이템 유사도와 평점이 관련이 있기 때문이다. 평점이 2개 이상인 경우는 각각의 아이템들의 평점이 반영되어 계산되지만 1개의 경우는 평점이 영향을 미치지 못하기 때문이다.

따라서 만족한 평점(Rated Good)인 8점 이상을 입력할 때 유사도가 높은 순으로 추천하여 성능 평가를 수행하였고, 동등한 조건하에 다른 알고리즘들을 평가하기 위해 8점 미만은 제외하였다.

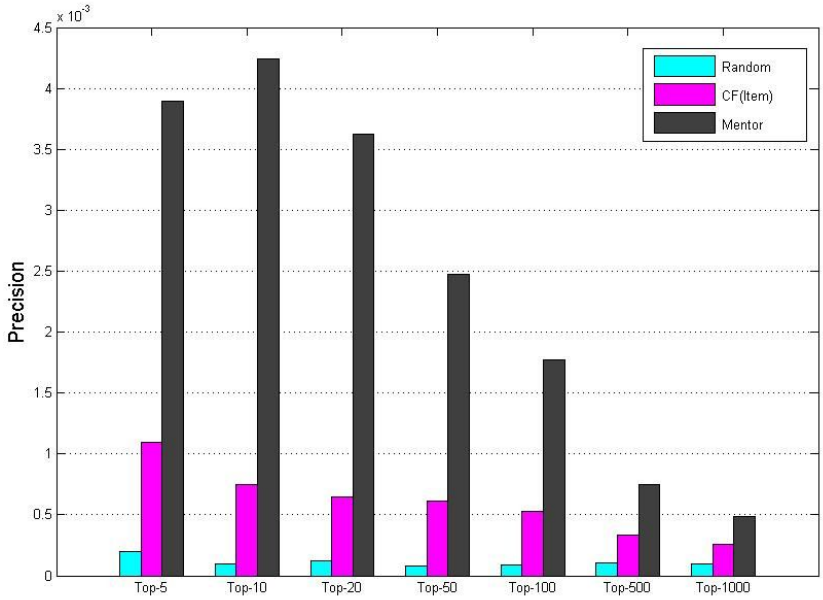
제 3 절 시스템 성능 분석

시스템 성능 분석은 멘토 기반의 추천시스템과 다른 알고리즘과의 비교 평가로 진행하였다. 비교 알고리즘 중에 첫 번째는 아이템 기반의 협업 필터링 방식이다. 선행연구에서도 언급하였듯이 아이템 유사도를 구하여 비슷한 아이템을 추천하는 알고리즘이다. 아이템 유사도를 구하기 위해서 사용자가 남긴 평점정보를 사용하였는데, 실제 평점정보가 매우 희박하기 때문에 유의미한 유사도를 구하기 위해 행렬 인수분해(matrix factorization)중 SVD(k=20)를 사용한 뒤 유사도를 구하였다.

이 때 평점을 한 개 남긴 사용자의 경우에는 앞에서 언급하였듯이 8점 이상일 때 아이템 유사도가 높은 순서로 영화 아이템을 추천해 주는 방식을 채택하였다.

두 번째 비교 방법은 랜덤기반으로 전체 영화 아이템 ID를 랜덤으로 생성하여 추천하는 방식을 택하였다.

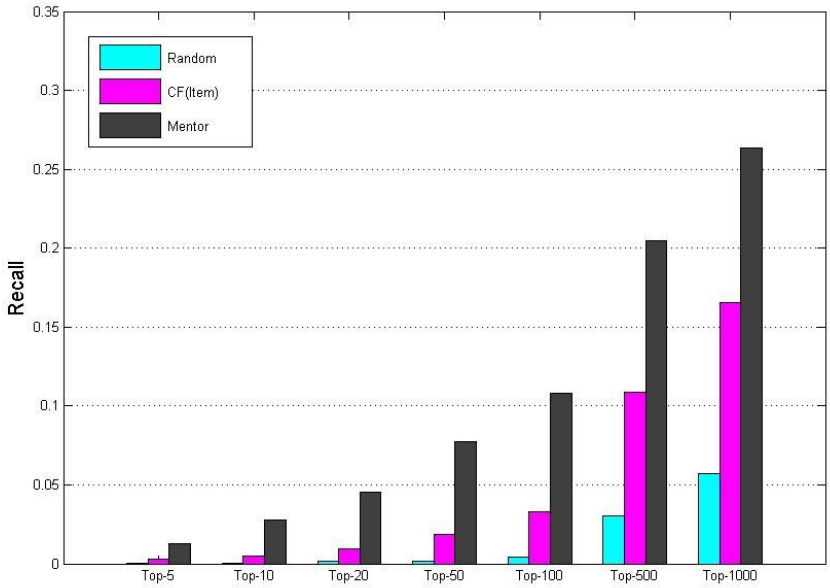
1. 평점개수가 1개인 사용자에 대한 성능 평가



[그림 23] 1개의 평점에 대한 평균 정확도(Mean Precision)

평점을 1개 남긴 사용자 중에서 8점 이상의 입력 점수 가진 사용자

2002명을 대상으로 수행한 평균 정확도는 [그림 23]과 같다. 이 연구에서 제안하는 멘토 기반의 알고리즘이 가장 좋은 성능을 보이고 예상대로 랜덤 기반 보다는 아이템 기반의 협업 필터링이 더 나은 결과를 보였다. Top-10에서 멘토 알고리즘의 정확도가 가장 높은 것으로 확인되고, 추천된 결과가 증가 할수록 알고리즘의 차이는 점점 줄어드는 것으로 확인된다. 이는 정확도의 특성을 반영한 결과라 본다.



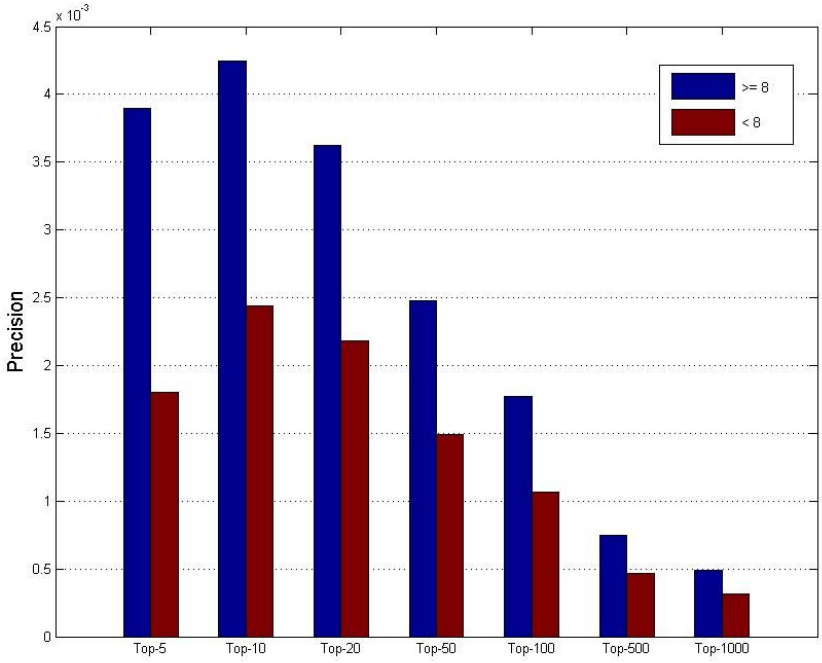
[그림 24] 1개의 평점에 대한 평균 재현율(Mean Recall)

재현율(Recall) 또한 세 알고리즘 중 멘토 기반의 알고리즘이 모든 결과에서 가장 좋은 재현율을 보였다. 더불어 재현율의 특성상 Top-N 값이 증가할수록 재현율 값 또한 커지는 것을 확인할 수 있다.

정확도와 재현율 그래프 상에서 보이 듯이 실제 값에 차이는 그다지 크지 않다. 이는 평가 평점 데이터의 데이터 개수 자체가 적기 때문으로 본다. 따라서 Cremonesi [24]의 연구에서도 언급하였듯이 결과를 절대값으로 해석하지 말아야 하고 같은 데이터 셋에서 다른 알고리즘과의 비교를 통한 결과 해석이 필요하다.

2. 1개의 평점 점수에 따른 멘토기반 추천시스템의 성능 평가

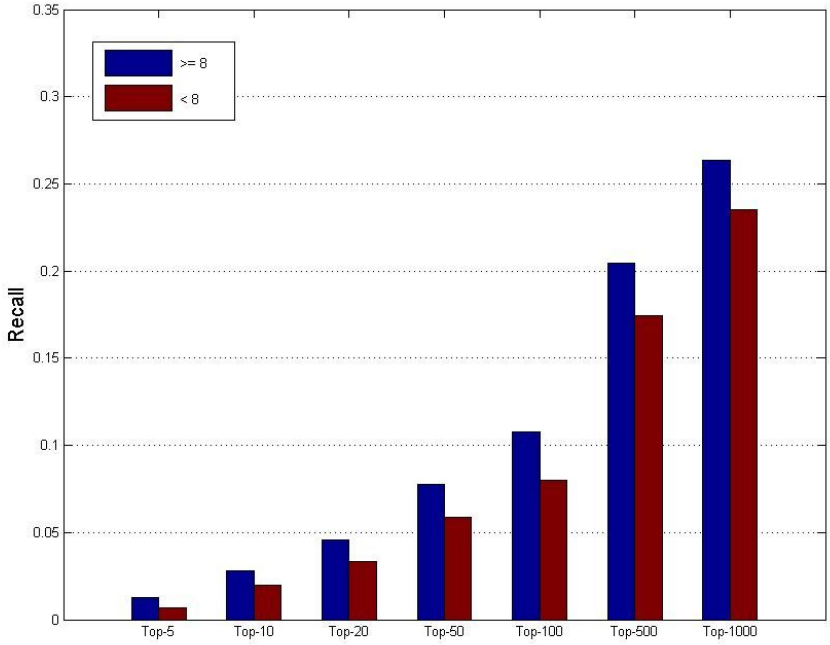
다른 알고리즘과의 비교 평가와 더불어서 입력 평점 데이터가 1개일 때, 평점을 기준으로 8점 이상 일 때에 추천 결과와 8점 미만인 경우에 멘토 기반의 알고리즘 추천 성능도 비교해보았다. 이는 입력된 데이터에 종류에 따라 나타날 수 있는 추천 성능을 비교해 보고자 함이다.



[그림 25] 멘토기반 추천의 평점에 따른 평균 정확도(Mean Precision)

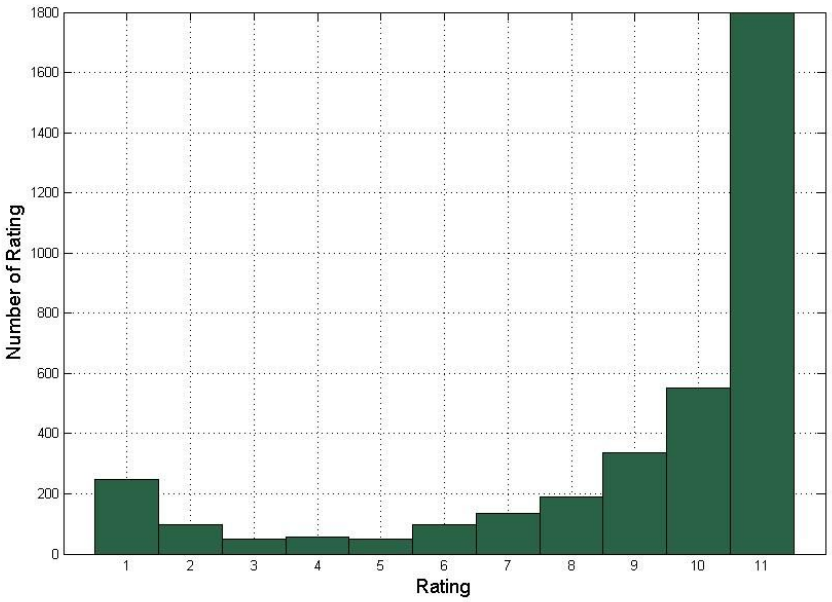
결과는 다음 그래프와 같다. 일반적으로 8점이상 즉 사용자가 만족한 영화(Rated Good)를 입력했을 경우가 그렇지 않은 경우에 비해 정확도 [그림 25]와 재현율 [그림 26]에 있어서 조금 더 나은 결과를 보였다.

이런 결과의 차이는 성능이 저하되었다기 보다는 평가 데이터의 점수 분포차이 때문이라 본다.

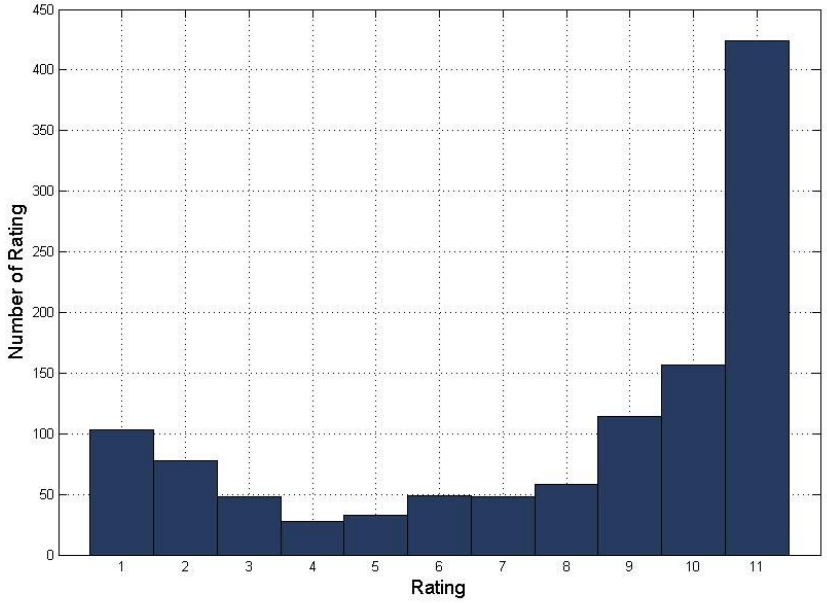


[그림 26] 멘토기반 추천의 평점에 따른 평균 재현율(Mean Recall)

[그림 27]와 [그림 28]에서 점수분포도를 확인하면 정확도와 재현율의 기준이 되는 사용자가 만족한 영화(Rated Good) – 8점 이상의 영화 평점에 대한 전체 비율이, 입력 평점이 8점 이상인 평가데이터에서 3610개 중 2689개로 약 75%를 차지하고 있고, 입력 평점이 8점 미만의 평가데이터에서는 1140개의 데이터 중 695개로 약 61%를 차지하기 때문에, 결과적으로 성능의 차이라기 보다는 평가데이터의 차이 때문에 나타난 결과라 생각된다.



[그림 27] 입력 평점 8점 이상의 평가 데이터의 점수 분포도



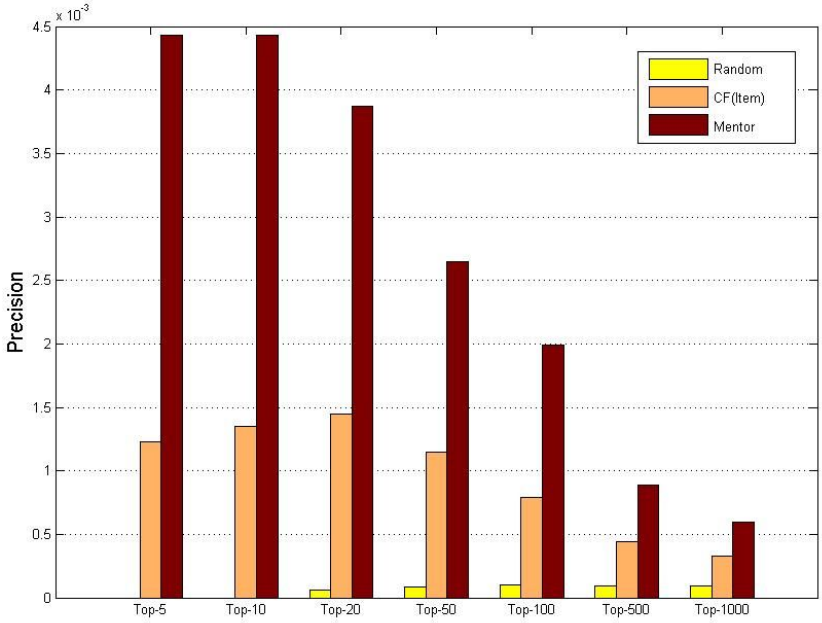
[그림 28] 입력 평점 8점 미만의 평가 데이터의 점수 분포도

3. 평점개수가 2개인 사용자에 대한 성능 평가

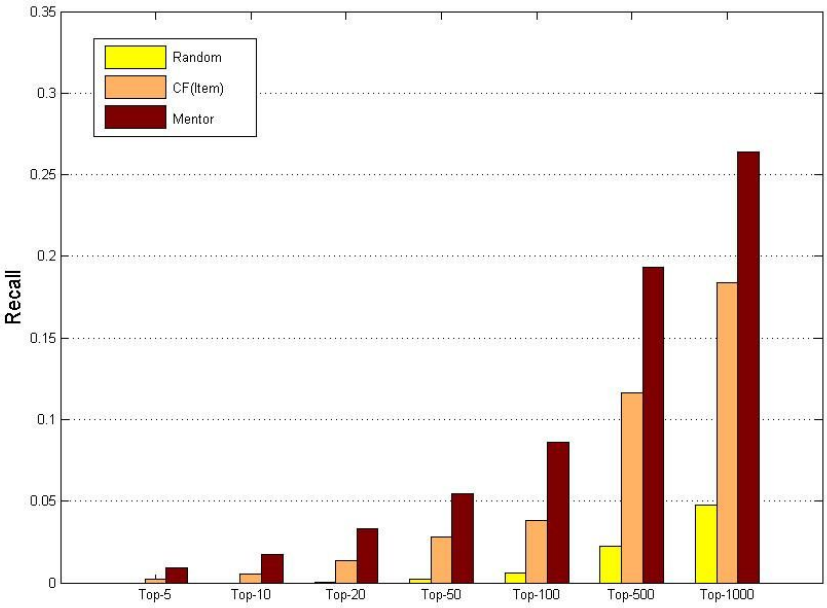
다음으로 평점을 2개 남긴 사용자 중 이후에 평점을 1개 이상 남긴 유저는 1612명이 수집되었다. 평점을 2개 남긴 사용자는 평점이 각 아이템마다 다를 수 있고, 해당 평점은 협업필터링 과정에서 반영되기 때문에 평점에 구분 없이 세 알고리즘을 실험했다.

평균 정확도와 재현율 모두 앞선 결과와 비슷하게 멘토 기반의 알고리즘이 다른 알고리즘보다 더 나은 결과를 보인다.

전체적으로 1개의 평점에 따른 결과와 비슷한 형태의 분포를 가지지만 협업필터링 방식과 멘토기반의 추천 방식에 평균값이 조금 높아진 것이 관찰 된다.

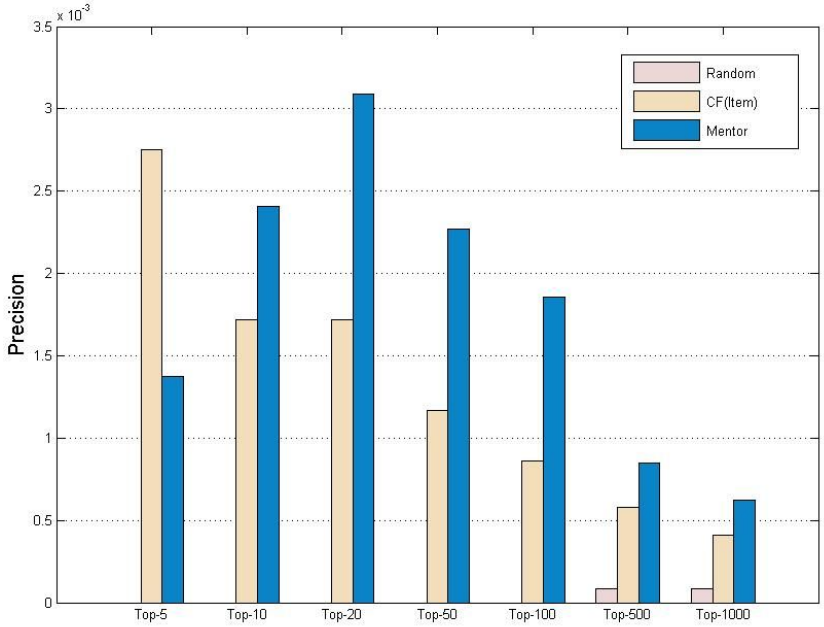


[그림 29] 2개의 평점에 대한 평균 정확도(Mean Precision)



[그림 30] 2개의 평점에 대한 평균 재현율(Mean Recall)

4. 평점개수가 10개인 사용자에 대한 성능 평가

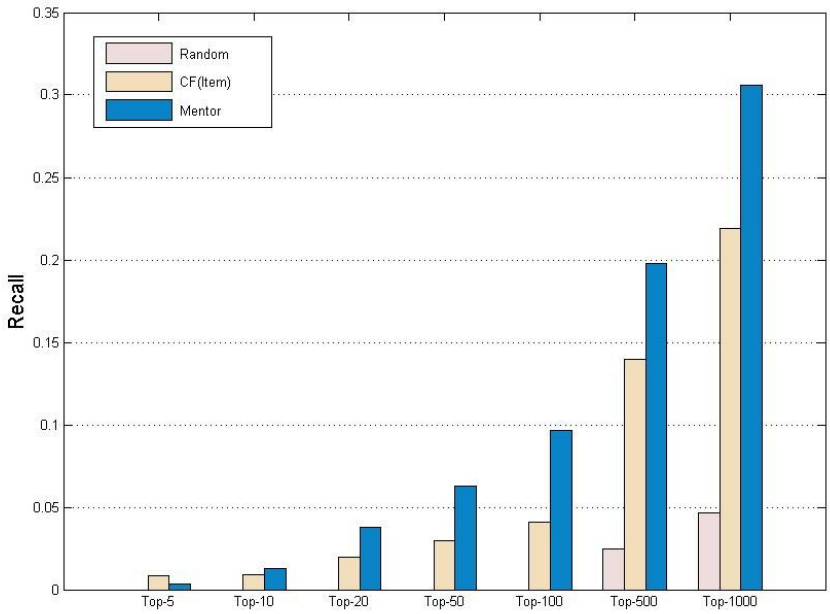


[그림 31] 10개의 평점에 대한 평균 정확도(Mean Precision)

10개의 평점을 남긴 사용자중 이후에 추가 평점을 남긴 사용자는 291명이 존재하고, 입력 평점개수는 총 2910개가 존재하고 평가 평점개수는 5328개 존재한다.

[그림 31]에서 처럼 평균 정확도에 경우 Top-5개 일 때 협업필터링에 성능이 멘토기반의 추천 방식을 추월한 결과가 나타났고 전체적으로 멘토기반과 협업필터링의 차이가 많이 줄어든 것으로 확인된다.

재현율에 경우도 정확도와 마찬가지로 Top-5일 때 협업필터링이 더 좋은 성능을 보였다. 이와 같이 유저가 남긴 평점의 개수가 계속 증가할수록 이와 같은 현상은 계속될 것으로 보인다. 그리고 협업 필터링을 위한 평점개수가 적정 개수에 도달하면 성능은 역전될 것으로 예상된다.

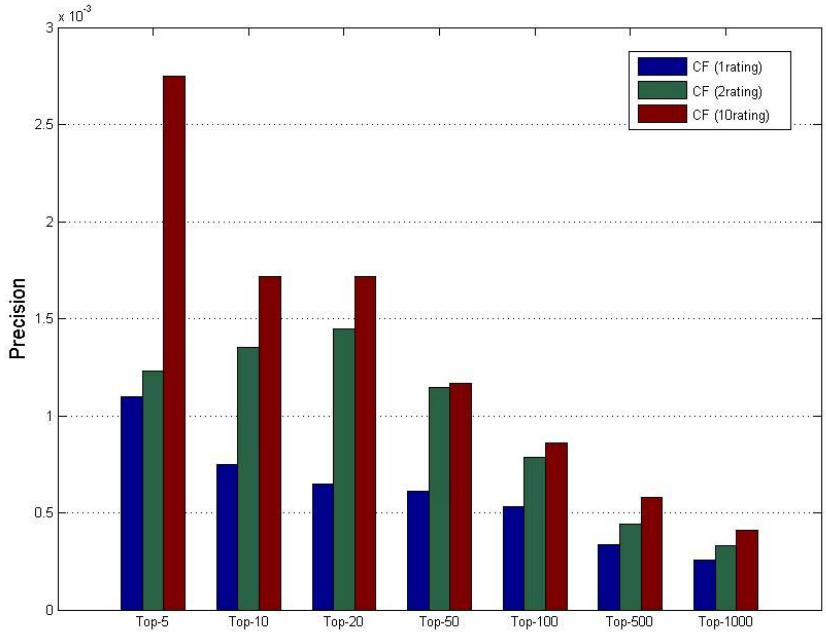


[그림 32] 10개의 평점에 대한 평균 재현율(Mean Recall)

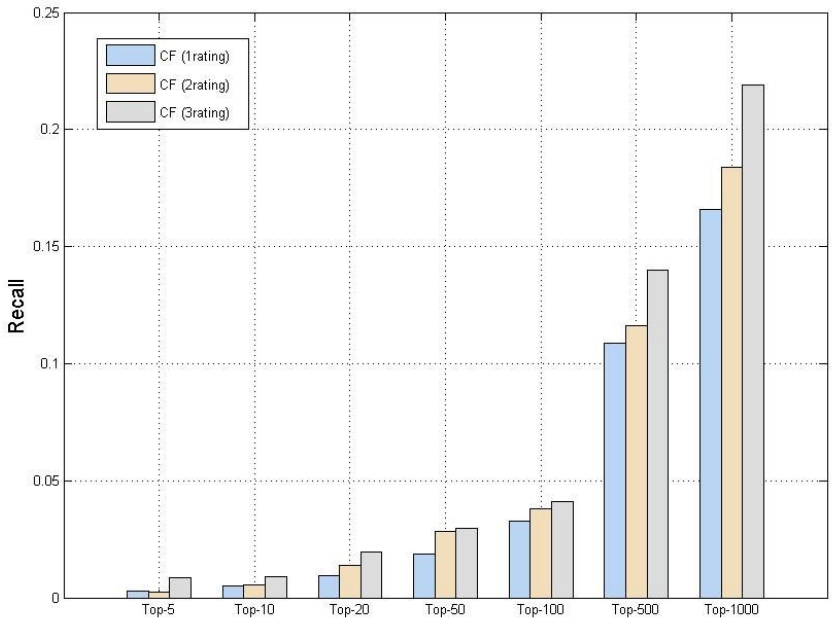
5. 협업필터링 방식의 평점 개수에 따른 성능 평가

협업필터링 방식과 멘토기반의 알고리즘에 평점의 개수에 따른 성능 변화를 보다 명확히 관찰하기 위하여 알고리즘 각각을 분리하여 나타내었다. 먼저 협업 필터링의 경우 예상대로 평점의 개수가 증가함에 따라 전체적인 성능이 높아지는 것을 확인 할 수 있다. [그림 33]와

[그림 34]에서 처럼 평균 정확도와 평균 재현율이 평점개수에 따라 점점 높아지는 것을 관찰 할 수 있다.



[그림 33] 평점 개수에 따른 협업필터링의 평균 정확도

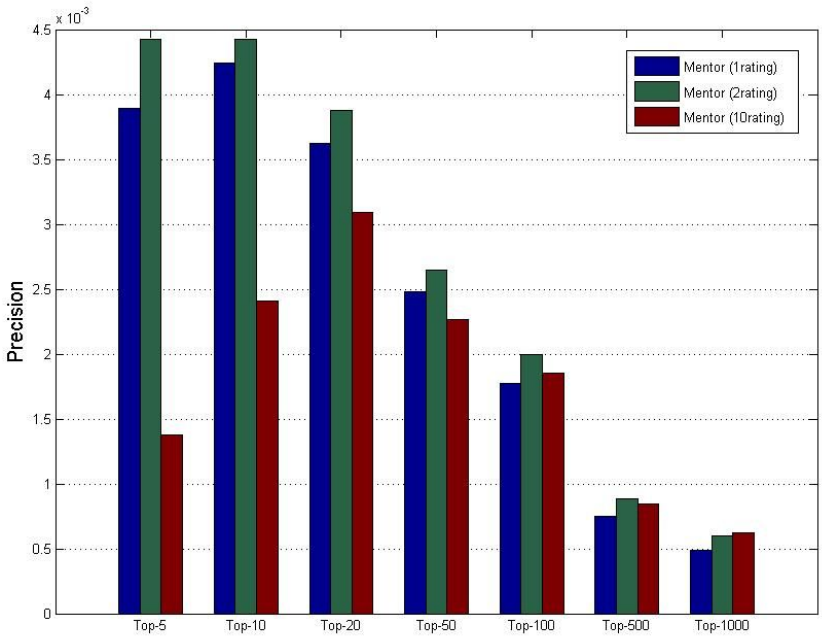


[그림 34] 평점 개수에 따른 협업필터링의 평균 재현율

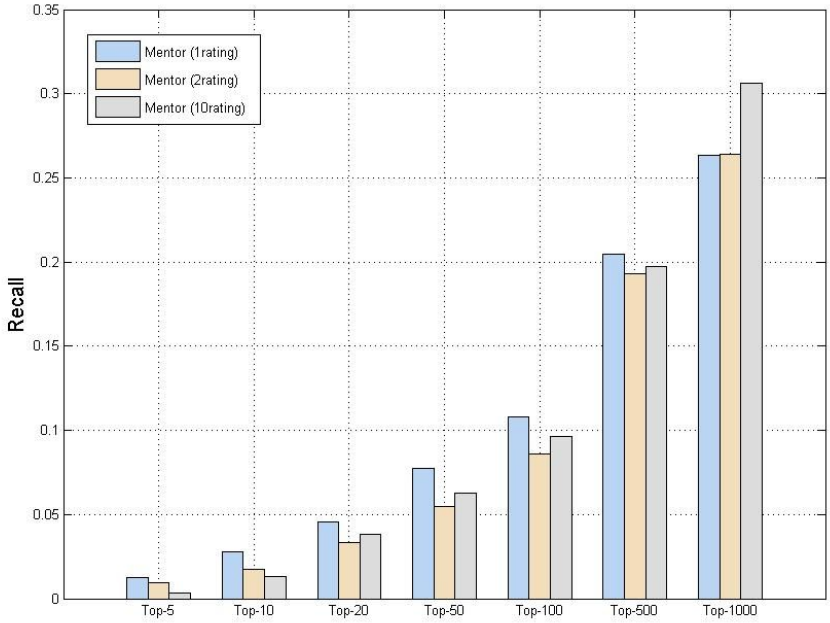
6. 멘토기반 알고리즘의 평점 개수에 따른 성능 평가

협업필터링 방식과는 다르게 멘토기반의 알고리즘은 평균 정확도에서는 평점이 2개 일 때 까지 성능이 높아지다가 평점 10개에서는 성능이 감소되는 것으로 관찰되었다. 재현율에서는 Top-N의 N개의 수가 적을 때는 성능이 점차 감소하는 것으로 관찰되었고 N개의 개수가 커짐에 따라 성능 역전현상이 벌어지는 곳도 존재하였다.

이렇게 멘토기반의 알고리즘의 성능이 점차 감소하는 이유는 아이템 개별의 추천 목록을 각각 생성 후 병합하기 때문이라 생각된다. 아이템의 개수가 증가함에 따라서 아이템 전체의 특성을 파악해서 추천해주는 협업필터링과는 다르게 각각의 아이템만을 기준으로 추천해준 목록을 병합한 결과는 전체의 특성을 파악할 수 없기 때문이다.



[그림 35] 평점 개수에 따른 멘토기반 알고리즘의 평균 정확도



[그림 36] 평점 개수에 따른 멘토기반 알고리즘의 평균 재현율

결과를 요약하면 다음과 같다.

[표 15] 시스템 성능 평가 요약

3. 1. 평점개수가 1개인 사용자에 대한 성능 평가
-멘토기반 알고리즘 방식이 다른 알고리즘에 비해 높은 성능 보임
3. 2. 1개의 평점 점수에 따른 멘토기반 추천시스템의 성능 평가
-평점 점수에 따라서 멘토기반 추천시스템의 성능 차이는 보이지 않음
3. 3. 평점개수가 2개인 사용자에 대한 성능 평가
-전체적으로 1개의 평점에 따른 결과 보다 평균값이 상승하였고 멘토기반 추천시스템이 가장 좋은 성능을 보임
3. 4. 평점개수가 10개인 사용자에 대한 성능 평가
-Top 5에서 협업필터링 방식이 멘토기반 보다 높은 성능을 보임
3. 5. 협업필터링 방식의 평점 개수에 따른 성능 평가
-평균정확도와 재현율이 평점의 개수에 따라서 높아짐
3. 6. 멘토기반 알고리즘의 평점 개수에 따른 성능 평가
-평점개수에 따라서 평균정확도와 재현율이 차이가 있으나 개수에 증가에 따라서 전체적으로 추천 성능이 감소함

전체적인 결과를 종합하면 평점에 개수가 희박한 1~2개의 사용자에게는 멘토기반의 알고리즘이 다른 알고리즘 보다 정확도와 재현율에 있어서 더 좋은 성능을 보장한다. 그리고 평점의 개수가 증가함에 따라 멘토기반의 알고리즘의 성능은 점차 감소하고 협업필터링의 성능은 점차 높아지는 것을 확인할 수 있다. 협업필터링 조건에 만족하는 적정개수의 평점이 입력될 경우에는 멘토기반의 알고리즘의 성능을 추월할 것으로 예상되며 이 연구에 목적에 맞게 멘토기반의 알고리즘은 희박한 평점데이터의 사용자에게 적합한 알고리즘으로 증명되었다.

제 5 장 결　　론

제 1 절 요약 및 시사점

통신 기술의 발달과 스마트 기기의 보급은 영화 콘텐츠를 시청 하는 방식의 다양화를 가져왔다. 그에 발맞추어 영화 콘텐츠의 양도 시간이 지남에 따라 급격하게 증가하고 있다. 이렇게 수많은 영화 콘텐츠 중에서 자신에 취향에 맞는 영화를 사전 지식 없이 선택하기란 많은 어려움이 따른다.

이에 따라 검색서비스의 한계와 영화 추천시스템의 필요성이 대두되면서 다양한 연구들이 진행되고 있다. 하지만 기존의 연구들은 정형화된 데이터 셋 예를 들어 무비렌즈와 같은 데이터를 가지고 연구를 수행한다. 이런 정형화된 데이터 셋은 일반적으로 기본 조건이 명시되어 있는데 '적어도 각 사용자는 20개의 다른 영화에 대한 평점 정보를 가진다' 와 같은 조건이다.

이는 바꾸어 말하면 20개 이하의 평점을 가진 유저들은 제외되었다는 말로 해석 할 수 있다. 그러나 실제 영화 정보 사이트에서 평점을 20개 이상 남긴 사용자는 전체 비율 중에 극히 적을 것으로 예측된다.

따라서 이 연구는 전체 사용자중 극도로 희박한 평점을 남긴 유저에 초점을 맞추어 유의미한 영화 추천시스템을 제안하고자 한다.

이 연구는 경험 없는 사람에게 다양한 조언과 도움을 주는 사람을 일컫는 말인 '멘토'라는 단어를 사용하여 영화 속에서의 사용자의 멘토들을 찾고 해당 멘토 그룹에 아이템중에 사용자가 만족할 만한 영화 아이템을 추천해준다.

이를 평가하기 위해 우리는 추천시스템 성능 평가에 가장 많이 사용되는 정확도(Precision)와 재현율(Recall)을 이용하였다.

희박한 평점을 가진 사용자를 관찰 추적하여 평가 데이터 셋을 만들고 다른 알고리즘과 같은 조건으로 실험한 뒤 이 연구에서 제안하는 알고리즘을 비교 분석하였다.

실험 결과 다른 알고리즘 보다 정확도와 재현율에 있어서 더 나은 결과를 보였으며 희박한 평점을 가진 사용자에게도 의미 있는 영화 추천시스템의 가능성을 확인 할 수 있었다.

제 2 절 연구의 의의 및 한계

본 논문에서는 추천시스템을 설계하고 평가하는데 있어서 정형화된 데이터 셋으로 인하여 제외되었던 극도로 희박한 평점정보를 가진 사용자에게 초점을 맞추어 연구를 진행하였다.

실제 많은 사람들이 이용하는 유명 포털사이트 다음(Daum) 영화 데이터를 기준으로 희박한 평점정보를 가진 사용자는 전체 평점을 남긴 유저 중에 98% 해당하고, 그 중에서도 이 연구에서 초점을 맞춘 1~2개의 평점정보를 가진 사용자는 79%에 해당했다.

이처럼 절대 다수에 해당하는 사용자를 중심으로 추천시스템을 설계하고 개발했다는 점과 절대 다수의 사용자들에게도 참신하고 새로운 아이템을 추천할 수 있다는 점에서 이 연구는 의의가 있다고 본다.

추천 방법에서 연구에 핵심인 멘토를 전체 사용자 가운데에서 일정 조건에 해당하는 유저를 대상으로 선정하기 때문에 다른 제약조건이 존재하지 않아 영화 뿐만 아니라 음악, 책과 같은 다른 분야에서도 적용될 수 있다고 본다.

하지만 이 연구에서 제안하는 멘토는 장르를 기준으로 멘토를 선정하기 때문에 추천 받는 사용자의 영화데이터에 장르가 존재하지 않는 경우 영화추천을 해줄 수 없다는 점이 존재한다.

또한 사용자가 남긴 영화데이터가 극히 희소하여 멘토 그룹 안에 존재하지 않을 경우에도 같은 방식으로 멘토를 발견 할 수 없기 때문에 추천이 불가능하다는 단점이 있다.

이외에도 다른 접근의 추천시스템과는 직접적인 비교를 하지 못한 여 성능평가를 하지 못한 점도 존재한다.

마지막으로 다음(Daum) 데이터 셋 이외에 보다 크고 다양한 영화 데이터 셋에서도 실험하지 못한 점은 본 연구에 한계라고 생각한다.

본 논문에서 제안하는 멘토기반의 영화추천시스템은 협업 필터링에 한계점으로 제시되는 사용자 데이터 희소성에 집중한 영화 추천시스템으로써 대다수를 차지하는 희박한 평점정보를 가진 사용자에게도 유의미한 추천을 가능하게 하였고 추천을 받은 사용자들 또한 더 정밀한 추천을 위해서 평점정보를 더 남길 수 있는 여지를 만들었기 때문에 기존 알고리즘의 활용성을 극대화시킬 수 있는 가능성을 제공했다고 본다.

참고 문헌

[1] P. Resnick, P. Bergstrom, and J. Riedl, “GroupLens : An Open Architecture for Collaborative Filtering of Netnews,” in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175-186.

[2] A. S. Rm and P. Maes, “Social Information Filtering : Algorithms for Automating ‘ Word of Mouth ’,” *Media*, pp. 210-217, 1995.

[3] J. A. Konstan, 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, pp. 77-87, 1997.

[4] W. Hill, L. Stead, and M. Rosenstein, “Recommending and evaluating choices in a virtual community of use,” *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 194-201, 1995.

[5] J. Davidson, B. Liebald, and T. V. Vleet, “The YouTube Video Recommendation System,” in *Design*, 2010, pp. 293-296.

[6] R. Baeza-Yates and B. Ribeiro-Neto, “Modern information retrieval,” vol. 82, 1999.

[7] C. Christakou and A. Stafylopatis, “A hybrid movie recommender system based on neural networks,” in *5th International Conference on Intelligent Systems Design and Applications (ISDA’05)*, 2005, pp. 500-505.

[8] H. Mak, I. Koprinska, and J. Poon, “INTIMATE : A Web-Based Movie Recommender Using Text Categorization,” in *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, 2003, no. Imd, pp. 8-11.

[9] G. Adomavicius and A. Tuzhilin, “Toward the Next Generation of Recommender Systems : A Survey of the State-of-the-Art and Possible Extensions,” in *Knowledge Creation Diffusion Utilization*, 2005, vol. 17, no. 6, pp. 734-749.

[10] C. Christakou and A. Stafylopatis, “A Hybrid Movie Recommender System Based on Neural Networks,” in *Design*, 2005.

[11] S. Debnath, H. I. Storage, and R. Information, “Feature Weighting in Content Based Recommendation,” in *ReCALL*, 2008, pp. 1041-1042.

[12] P. Session, “Towards a Better Understanding of Context and Context-Awareness,” *Computing Systems*, pp. 304-307, 1999.

[13] C. Ono, M. Kurokawa, and Y. Motomura, “A context-aware movie preference model using a Bayesian network for recommendation and promotion,” *User Modeling 2007*, pp. 247-257, 2007.

[14] J. Herlocker, J. Konstan, and A. Borchers, “An algorithmic framework for performing collaborative filtering,” in *Proceedings of the 22nd*, 1999.

[15] M. Deshpande and G. Karypis, “Item-based top- N recommendation algorithms,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 143-177, Jan. 2004.

[16] J. Golbeck, “Filmtrust: Movie recommendations using trust in web-based social networks,” in *Proceedings of the IEEE Consumer communications and networking conference*, 2006, vol. 96.

[17] Y. Ding, “Time weight collaborative filtering,” *Proceedings of the 14th ACM international conference*, pp. 485-492, 2005.

[18] L. H. Ungar and D. P. Foster, “Clustering Methods for Collaborative Filtering,” in *AAAI Workshop on Recommendation Systems*, 1998, pp. 114-129.

[19] S.-M. Choi and Y.-S. Han, “A Content Recommendation System Based on Category Correlations,” in *2010 Fifth International Multi-conference on Computing in the Global Information Technology*, 2010, no. 1, pp. 66-70.

[20] K.-rog Kim, J.-ho Lee, and J.-hee Byeon, “Recommender System Using the Movie Genre similiarity in Mobile Service,” in *Technology*, 2010.

[21] Ò. Celma and P. Herrera, “A new approach to evaluating novel recommendations,” in *Proceedings of the 2008 ACM conference on Recommender systems - RecSys ’08*, 2008, p. 179.

[22] D. Billsus and M. . Pazzani, “Learning collaborative information filters,” in *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, vol. 54, p. 48.

[23] P. Melville, R. J. Mooney, and R. Nagarajan, “Content-Boosted Collaborative Filtering for Improved Recommendations,” in *American Association for Artificial Intelligence (www.aaai.org)*, 2002, pp. 187-192.

[24] P. Cremonesi, R. Turrin, E. Lentini, and M. Matteucci, “An Evaluation Methodology for Collaborative Recommender Systems,” *2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution*, pp. 224-231, Nov. 2008.

[25] ‘2011 년 한국 영화산업 결산 보고서’ 영화진흥위원회

[26]http://biz.heraldm.com/common/Detail.jsp?newsMLId=20111212000043

[27] Oxford Advanced Learner's English-Korean Dictionary
http://endic.naver.com/enkrEntry.nhn?entryId=2067266928334948a87fa0b140f7c3d6

[28] 출처 : 박문각
http://terms.naver.com/entry.nhn?docId=67931&cid=432&categoryId=2264

부 록

실험 결과값

1개의 평점에 대한 평균 정확도(Mean Precision)

	Random	CF	Mentor
Top-5	0.0001998	0.001099	0.003896
Top-10	9.99E-05	0.000749	0.004246
Top-20	0.00012488	0.000649	0.003621
Top-50	7.99E-05	0.000609	0.002478
Top-100	8.99E-05	0.000529	0.001773
Top-500	0.0001049	0.000338	0.00075
Top-1000	9.74E-05	0.000255	0.000486

1개의 평점에 대한 평균 재현율(Mean Recall)

	Random	CF	Mentor
Top-5	0.000583	0.002934	0.012687
Top-10	0.000583	0.004932	0.02784
Top-20	0.00135	0.009487	0.045496
Top-50	0.001924	0.018672	0.077512
Top-100	0.004481	0.032712	0.108004
Top-500	0.03068	0.108814	0.204612
Top-1000	0.057301	0.165809	0.263381

멘토기반 알고리즘의 평점기준에 따른 평균 정확도(Mean Precision)

	>=8	< 8
Top-5	0.0038961	0.001799
Top-10	0.00424575	0.002442
Top-20	0.00362138	0.002185
Top-50	0.00247752	0.001491
Top-100	0.00177323	0.001067
Top-500	0.00075025	0.00047
Top-1000	0.00048601	0.000317

멘토기반 알고리즘의 평점기준에 따른 평균 재현율(Mean Recall)

	>=8	< 8
Top-5	0.012687	0.006962
Top-10	0.02784	0.019571
Top-20	0.045496	0.033572
Top-50	0.077512	0.058444
Top-100	0.108004	0.080264
Top-500	0.204612	0.174501
Top-1000	0.263381	0.235041

2개의 평점에 대한 평균 정확도(Mean Precision)

	Random	CF	Mentor
Top-5	0	0.00123	0.004428
Top-10	0	0.001353	0.004428
Top-20	6.15E-05	0.001445	0.003875
Top-50	8.61E-05	0.001144	0.002645
Top-100	0.00010455	0.000787	0.001993
Top-500	8.98E-05	0.000442	0.000886
Top-1000	9.16E-05	0.000331	0.000598

2 개의 평점에 대한 평균 재현율(Mean Recall)

	Random	CF	Mentor
Top-5	0	0.002381	0.009252
Top-10	0	0.005415	0.017531
Top-20	0.000318	0.013862	0.033389
Top-50	0.002163	0.028353	0.054697
Top-100	0.006271	0.038025	0.086077
Top-500	0.022577	0.11607	0.193342
Top-1000	0.04769	0.183697	0.263943

10개의 평점에 대한 평균 재현율(Mean Precision)

	Random	CF	Mentor
Top-5	0	0.002749	0.001375
Top-10	0	0.001718	0.002405
Top-20	0	0.001718	0.003093
Top-50	0	0.001168	0.002268
Top-100	0	0.000859	0.001856
Top-500	8.25E-05	0.000577	0.000845
Top-1000	8.59E-05	0.000409	0.000625

10 개의 평점에 대한 평균 재현율(Mean Recall)

	Random	CF	Mentor
Top-5	0	0.008591	0.003608
Top-10	0	0.009164	0.013036
Top-20	0	0.019737	0.03795
Top-50	0	0.029732	0.062834
Top-100	0	0.041142	0.096556
Top-500	0.02459	0.139875	0.197607
Top-1000	0.046564	0.218852	0.305971

Abstract

Movie Recommender System Based on Mentor for the Users Who Have Sparse Rating Data

Chun, Sung Kwon
Department of Transdisciplinary Studies
The Graduate School
Seoul National University

The development of communication technology and the spread of smart devices have brought diverse methods for watching movies. Moreover, the amount of film contents has also increased drastically over time. However, it is not always an easy task for a user to select a movie to one’ s appetite. If a user knows the specific movie he/she wants to see, the user can simply use a search service. Otherwise, the search is limited.

Thus, the recommendation system improves the limitation of search services and works as a suitable alternative to resolve a variety of user needs. Since the recommendation system recommends items based on features about the contents or the users, the results may be more reasonable.

Until now, research on the movie recommendation system has shown a significant progress. However, conventional studies conducted researches on structured datasets, such as MovieLens. Such structured datasets generally state underlying conditions of 'each user having rated at least 20 movies'. In other words, users

who have a rating of less than 20 are ruled out from the research target. However, user who left a rating of more than 20 is expected to be very few in the actual movie site.

In this study, a movie recommendation system based on mentor that focuses on solving the sparse user data is proposed. The movie recommendation system based on mentor aims at enabling users who have extremely sparse rating, especially one or two ratings. In order to design the movie recommendation based on mentor, data on users, movies, reviews are collected through DAUM Portal site used many actual users.

Finally, this study performs evaluation to compare existing recommendation algorithms with this suggested recommendation system.

Keywords : movie recommender system, collaborative filtering, sparse user data

Student Number : 2011-22771