댓글-평점 관계를 기반으로
사용자 편향을 보정한 추천 시스템 설계

초 록

본 논문에서는 사용자의 편향을 고려하지 못한 기존의 평점 기반 협업 필터링을 개선하기 위하여 사용자가 작성한 댓글을 이용하여 협업 필터링을 하는 추천 알고리즘을 제안하였다. 댓글에는 감정, 느낌 등의 사용자의 주관적 기준이 반영되어 있으므로 댓글-평점 관계를 CNN으로 학습시켜 비슷한 댓글에 대하여 비슷한 평점을 얻어냈다. 이렇게 도출된 평점을 기존의 추천 알고리즘과 접목하여 예측 선호도를 산출하였다. k-NN 알고리즘과 접목하였을 때 제안한 알고리즘의 RMSE와 MAE값이 전부 낮아졌으며 그 외 다른 7개의 추천 알고리즘 중 5개에 대하여 제안한 알고리즘의 성능이 향상되었음을 관찰할 수 있었다. 본 논문에서 제시한 사용자의 편향을 보정한 새로운 추천 시스템을 통해 목표 고객에 더 특화된 상품을 추천하는데 있어서 큰 도움이 될 것이다.

목차

표 목차

그림 목차

Ⅰ. 서론

21세기 정보화 사회에 접어들면서 개인 PC와 스마트폰이 보편화되었고 데이터 통신망을 이용하여 웹 서비스로의 접근이 용이해졌다. 웹 서비스를 통하여 소비자들은 시간과 장소에 구애받지 않고 원하는 재화, 또는 서비스를 찾고 선택하여 구매한다. 이러한 과정 속에서 과거 소비자들이 남기는 댓글과 평점은 현 소비자에게 그 상품의 가치를 판단할 수 있게 도와주는 기준이 된다. 한편, 셀 수 없이 많은 재화와 서비스가 인터넷에 축적됨에 따라 정보가 많아져 소비자들은 원하는 상품을 찾고 적합한 정보를 얻는 데에 어려움을 겪고 있다. 따라서, 서비스 제공자는 이러한 어려움을 해결하기 위하여 다양한 방법들을 소비자들에게 제공한다. 그 중 대표적인 방법이 추천 시스템(Recommendation System)이다.

추천 시스템은 목표고객이 아직 접하지 않은 상품들 중 선호할 것 같은 상품들을 예상하여 추천하는 서비스로써 대표적인 기법으로 협업 필터링(Collaborative Filtering)이 있다[1,2]. 협업 필터링은 소비한 상품에 대한 각 소비자의 평가를 받아 그 패턴이 비슷한 사람들을 집단으로 묶는 방법으로, 목표고객과 비슷한 취향을 가진 사람들이 선호하는 또 다른 상품을 추천하는 방법이다. 하지만 기존의 알고리즘들은 평점들의 분포, 혹은 거리 등으로만 집단을 형성하여 결정적으로 사용자 개개인이 부여하는 평점의 기준이 다르다는 것을 반영하지 못하였다는 문제점이 있다. 따라서 이러한 문제점을 해결하기 위하여 여러 기법들을 도입하여 사용자의 편향을 고려한 연구가 진행되고 있다. 본 연구에서는 CNN (Convolutional Neural Network)을 이용하여 사용자가 자신의 생각을 표현한 댓글을 분석하고, 비편향 평점으로 도출하여 새로운 추천 시스템을 제안하였고, 검증하였다.

Ⅱ. 관련 연구

협업 필터링의 성능을 향상시키기 위한 많은 연구들이 수행되었다. 콘텐츠 기반(Content-based), 상황 기반(Context-based)등과 같은 여러 추천 시스템을 결합하여 상호 보완적으로 단점을 보완하는 연구들이 주를 이룬다[3,4,5,6]. 최종적으로 상품을 추천하기 전, 장르와 연령, 고객 평가정보 등 더 세밀한 요소들을 고려하여 더욱 더 목표고객에 적합하게끔 추천 시스템을 제안하는 연구들 또한 많이 수행되었다[7,8,9,10,11]. 각각의 요소들이 더욱 더 개개인에 특화된 추천목록을 작성하는데 도움을 주어 알고리즘의 성능을 향상시켰다. 사용자 정보를 전처리를 통하여 k-Means 알고리즘의 단점을 보완하여 선호도가 유사한 사용자들끼리 같은 집단에 속할 수 있도록 방법을 제시한 연구가 있다[12]. 하지만 여전히 평점의 분포만이 유사한 사용자를 정의하는 데에 영향을 준다는 한계를 지닌다.

평점의 편향을 해결하기 위한 다양한 연구들도 수행되었다. 평점을 정규화하여 추천시스템을 설계한 연구와 평점에서 극단적으로 편향된 저평점과 고평점을 노이즈로 간주하고 제거한 연구가 있다[13,14]. 류신(2015)은 리뷰를 이용하여 토픽으로 분류하고 키워드를 이용하여 평점을 보정한 후, 보정편점과 원평점의 오차를 비교하는 방법을 제시하였다[15].

또한 문장 분석을 추천 시스템에 접목시킨 연구들도 많이 수행되었다. 문장의 감성 분석을 통하여 알맞은 음악을 추천하는 연구가 있고, 문장에서 특징적인 단어들을 고려하여 긍정/부정을 판단하는 연구도 수행되었다[16,17]. 본 연구에서도 원평점 대비 보정평점 간의 에러를 비교했을 때 보정평점의 유사도가 개선되었음을 보였다.

2.1 감성 분석 (Sentiment Analysis)

오피니언 마이닝, 특히 감성 분석(Sentiment Analysis)은 텍스트의 정보를 추출하는 텍스트 마이닝과는 다르게 어떤 주제에 대한 주관적인 인사, 감정, 태도, 개인의 의견들을 텍스트로부터 뽑아내는 분석을 말한다. 일반적으로 찬성/반대, 좋음/싫음과 같이 2진형식으로 나타낸다. 더 나아가, 점수가 글쓴이의 감정을 대변할 수 있다는 생각에서 별점이나 평점 같은 스코어들을 예측하는 방향으로 확장되었다.

감성 분석의 정확도는 본래 인간의 판단과 얼마나 일치 하느냐로 결정된다. 실제로 실험에 따르면, 80% 정도가 일반적으로 동의를 한다고 한다[18]. 그러므로 70% 정확도를 갖는 프로그램은 거의 인간과 가깝다고 볼 수도 있으며, 100%의 정확도를 갖는 프로그램은 20%의 사람들은 여전히 동의하지 않을 것이다. 감성분석에서 2진형태보다 스케일형식의 반환값은 타겟값과 예측값이 얼마나 가까운지를 설명한다.

최근에 딥러닝을 이용한 자연어 처리 및 감성분석이 활발하게 연구되고 있으며, 특히 딥러닝 기술은 복잡하고 다양한 자료 및 질적, 양적 변수에 관계없이 모두 분석이 가능하며, 변수들 간 비선형 조합이 가능하여 예측력이 우수하다는 장점이 있다. 따라서 문장 분석에서도 놀라운 성능을 보이며, 관련 연구가 활발하게 진행되고 있다[19].

2.2 협업 필터링 (Collaborative Filtering)

협업 필터링은 크게 다음과 같은 세 단계로 이루어진다.

[1단계] 고객×상품 매트릭스의 준비

m명의 고객들이 n개의 상품에 각각 평점과 댓글을 남겼다. 이를 행렬로 정리하여 M_{mn}의 행렬에 값이 평점과 댓글이 각각 들어가게 한다.

[2단계] 최근접 이웃의 구성

고객들이 상품에 대하여 남긴 정보를 바탕으로 고객들간의 유사도를 계산하여 최근접 이웃을 구성한다. 이 유사도를 계산하는 과정에서 널리 사용되는 여러 측정지수가 있다. 가장 많이 사용되는 측정지수는 Pearson 상관관계 계수식으로 $w(A,B)$가 A와 B 사이의 유사도를 의미할 때 (1)과 같이 계산된다.

$$w(A,B)=\frac{\sum_{i=1}^{q}(R_{A,i}-\overline{R_A})(R_{B,i}-\overline{R_B})}{\sqrt{\sum_{i=1}^{q}(R_{A,i}-\overline{R_A})^2\sum_{i=1}^{q}(R_{B,i}-\overline{R_B})^2}} \quad (1)$$

q는 A와 B가 공통적으로 평가한 상품의 수이고 $R_{A,i}, R_{B,i}$는 각각 A와 B가 i번째 상품에 부여한 평점, 그리고 $\overline{R_A}, \overline{R_B}$는 A와 B가 부여한 총 평점의 평균을 의미한다. 이 유사도는 -1.0 ~ 1.0 사이의 값을 갖고 1.0에 가까울수록 A와 B가 유사하다고 판단된다. Cosine을 이용하는 방법도 있다. A와 B가 공통적으로 부여한 상품의 평점을 q차원 공간에 벡터화 한 후 두 벡터 사이의 코사인 값을 (2)와 같이 계산하여 이용하는 것이다.

$$\cos(\vec{A},\vec{B})=\frac{\vec{A}\circ\vec{B}}{\|\vec{A}\|^2\times\|\vec{B}\|^2} \quad (2)$$

최근접 이웃을 구성하는 방법 또한 여러 가지가 있는데 첫 번째는 k-NN 기법으로써 목표 고객과 유사도가 높은 k명의 사람들로 최근접 이웃을 구성하는 방법이고 두 번째는 threshold based selection 기법으로 목표 고객과 유사도가 미리

선정해둔 threshold 이상인 사람들로 최근접 이웃을 구성하는 방법이다. k-Means 기법은 유사도를 이용하는 것이 아닌 사용자의 평점을 다차원 공간상의 점으로 표시하고 거리를 계산하여 k개의 집단으로 묶는 방법이다.

하지만 부여하는 평점의 패턴이 유사하다고 두 사람을 최근접 이웃으로 묶어 목표 고객에게 상품을 추천하는 과정에서 추천 시스템의 정확도를 하락시킬 수 있는 요인이 존재한다. 두 사람 A와 B는 실제로 상품을 선택하는 패턴이 유사한 사람이라고 가정해보고자 한다. A는 평균 8~10점의 평점을 부여하는 사람인 반면 B는 평균 3~5점의 평점을 부여하는 사람이다. 실제로 이 두 사람은 유사한 사람이지만, 평점의 편차가 너무 커 k-NN이나 k-Means 기법에서는 이웃으로 묶이지 못한다. 또한, Pearson 상관관계 계수식이나 Cosine을 이용할 경우 부여하는 평점의 패턴이 유사한 사람들을 이웃으로 정의하였기 때문에 실제 상품에 대한 개개인의 감정과 평을 반영하지 못한다. 상품에 대한 개개인의 감정은 대체적으로 그 사람이 남기는 댓글 속에 반영되어 있고 이에 본 연구는 평점의 유사한 정도가 아닌, 댓글의 유사도를 이용하여 추천 시스템을 구성하는 방법을 제안하고자 한다.

[3단계] 추천 목록의 생성

목표 고객이 구매하지 않았던 상품들의 평가치를 최근접 이웃의 평가치로 계산하여 그 선호도를 기반으로 추천 목록을 생성한다. 두 고객 사이의 유사도의 가중치에 따라 평점을 가중 평균하여 평가치를 계산한다. 보통 사용되는 평가치의 계산식은 (3)과 같다.

$$R_{A,i} = \overline{R_A} + \frac{\sum_{j=1}^{k} w(A,j)(R_{j,i} - \overline{R_j})}{\sum_{j=1}^{k} |w(A,j)|} \quad (3)$$

이렇게 계산된 평가치를 바탕으로 선호도가 높은 N개의 상품을 추천하는 Top-N기법이 널리 이용된다.

2.3 성능 평가 척도

성능을 평가하는 방법에는 크게 두 가지 지표를 사용할 수 있다. 첫 번째는 MAE(Mean Absolute Error) 지표이다. MAE를 구하는 식은 (4)과 같다.

$$MAE = \frac{1}{n}\sum_{j=1}^{n} |r_j - \hat{r}_j| \quad (4)$$

두 번째는 RMSE(Root Mean Square Error)로 MAE와 유사하지만 편차의 제곱의 합을 제곱근 취한다는 점에서 다르다. RMSE를 구하는 식은 (5)와 같다.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n} (r_j - \hat{r}_j)^2} \quad (5)$$

주어진 선호도와 추천 시스템으로 인하여 계산된 선호도의 차이의 합을 데이터의 수로 나누어 얻어지는 지표들이다. 두 지표 모두 낮게 나올수록 추천 시스템의 성능이 좋다고 평가한다.

Ⅲ. 연구 방법

3.1 알고리즘 설계

댓글을 분석하기 위해 NN(Nerual Network)을 이용한 감성 분석을 하면 각 댓글의 문장이 가지는 보편적인 긍정/부정의 척도를 1에서 5 사이의 값으로 대응할 수 있다. 감성 분석에서 평점과 같은 일정한 스케일 형식의 반환값은 타겟값과 예측값이 얼마나 가까운지를 나타내는 척도로 이용할 수 있다. 거시적인 관점에서, 많은 양의 데이터를 이용하여 학습을 잘 진행한다면 감성분석을 통하여 도출된 결과는 리뷰를 달은 고객 집단 다수의 기준에 부합한다. 즉 리뷰 집단 다수의 타겟값과 예측값이 거의 일치하게 나올 것이다. 하지만 일부 소수의 다른 기준을 가진 사람들은 타겟값과 예측값이 차이가 발생할 것이다. 이 차이는 일부 소수 사용자들이 많은 사용자들과 다른 기준을 가진 그 사용자만의 편향으로 간주할 수 있다.

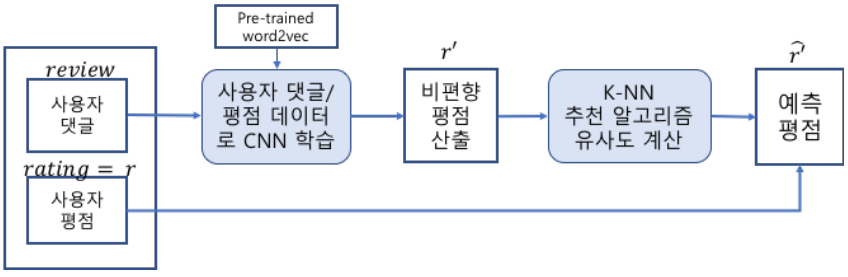
예를 들어, 두 사용자가 '너무 재밌어요'라는 댓글을 달았고, 이 댓글은 4.0 정도의 일반적인 긍정/부정 척도를 보인다고 상황을 가정해 보고자 한다. A 사용자가 5점을 주었고, B 사용자가 2점을 주어도 각각이 틀렸다고 말할 수 없다. 따라서 최근접 이웃을 구성할 때, 특정 사용자와 유사한 사용자들을 찾아야 하는데 단순히 평점의 분포를 사용하여 이웃을 구성한다면, 이러한 개개인의 편향을 전혀 반영하지 못한 이웃을 구성하게 된다.

본 연구에서는 영화에 대하여 유사한 댓글을 단 사람들을 유사하다고 판단하였다. 유사한 댓글을 가진 사람들은 각각의 댓글이 NN을 통과하여 도출된 평점이 유사하다. 따라서 이 도출된 평점을 이용하여 이웃을 구성하는 것이 원평점을 사용하는 것보다 개개인의 편향을 보정한 새로운 추천 시스템이다.

NN에서 댓글을 분석하기 위한 감성 분석기법에는 RNN(Recurrent Neural Network)과 CNN이 있는데, 번역과 같이 문장을 생성하는 경우에는 RNN을 사용해야 하지만, 주어진 분석 단계에서는 2가지 방법 모두 활발하게 연구되고 있다. 특히, CNN을 통한 감성 분석은 단순한 구조로 관심을 받고 있으며 본 연구에서도 CNN을 사용하였다[20,21].

3.2 비편향 추천 알고리즘

<그림 1.>은 본 논문에서 구성한 추천 시스템의 전체 구성도이다. 먼저 사용자 댓글을 이용하여 CNN을 학습시켰고, CNN의 예측 평점으로 추천 알고리즘을 학습시켰다. 자세한 절차는 <그림 2.>에 표현하였다.

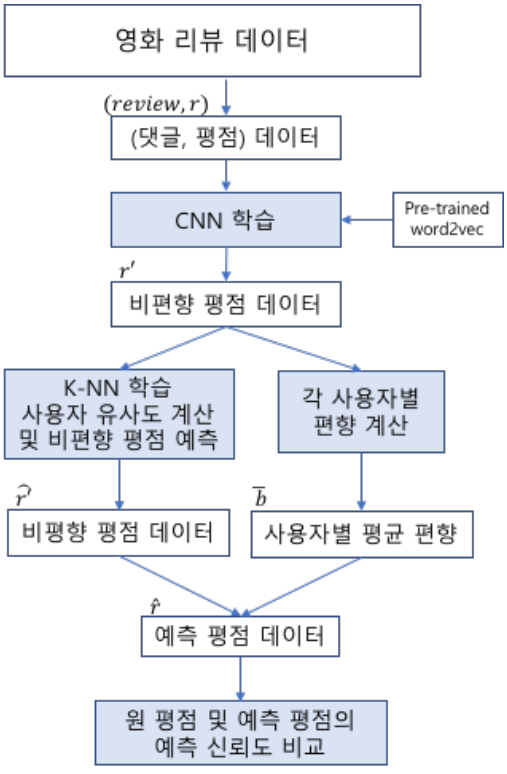


<그림 1.> 추천 시스템의 구성도

CNN 학습에서는, 학습 데이터 (training dataset)에서 모든 사용자 A에 대한 댓글 $review_A$와 평점 r_A을 사용하여 CNN을 훈련하였다. 그러면 훈련된 CNN을 통해서 각각의 사용자별로 새로운 평점 r'_A을 얻을 수 있으며, 이것은 댓글의 의미로부터 얻어진 비편향된 평점으로 생각할 수 있다. 만약 사용자 A가 편향을 가진다면 각각의 댓글에 대한 편향을 b_A라고 하고, r'_A는 r_A에 편향 b_A가 더해진

것으로 생각할 수 있고, 다음과 같은 식으로 표현할 수 있다.

$$r'_A \simeq r_A - b_A \tag{6}$$



<그림 2.> 추천 시스템의 처리 절차

사용자의 평균 편향 $\overline{b_A}$는 다음 식과 같이 사용자의 모든 댓글에 대한 평균으로 표현할 수 있다.

$$\overline{b_A} = \frac{\sum_{i}^{n_A}(r'_{A,i} - r_{A,i})}{n_A} \tag{7}$$

CNN 학습을 통해서, 댓글을 통한 사용자별 고유 평점인 비편향 평점을 예측

할 수 있고, 원 평점과의 차이인 편향을 계산할 수 있다. 다음으로 모든 사용자의 비편향 평점으로 추천 알고리즘을 학습시켰다. 추천 알고리즘은 이 비편향 평점으로 사용자간 유사도를 계산하였다. 다음은 사용자 A와 B사이의 유사도를 비편향 평점으로 계산한 Pearson 유사도 식이다.

$$w'(A,B) = \frac{\sum_{i=1}^{q}(r'_{A,i} - \overline{r'_A})(r'_{B,i} - \overline{r'_B})}{\sqrt{\sum_{i=1}^{q}(r'_{A,i} - \overline{r'_A})^2 \sum_{i=1}^{q}(r'_{B,i} - \overline{r'_B})^2}} \quad (8)$$

비편향 Pearson 유사도를 가지고 사용자 A의 i 영화에 대한 비편향 예측 평점을 다음과 같은 식으로 예측할 수 있다.

$$\widehat{r'_{A,i}} = \overline{r'_A} + \frac{\sum_{j=1}^{k} w'(A,j)(r'_{j,i} - \overline{r'_j})}{\sum_{j=1}^{k} |w'(A,j)|} \quad (9)$$

또한 사용자 A의 i 영화에 대한 편향 예측 평점은 다음과 같이 비편향 예측 평점에 사용자의 평균 편향값을 더한 식으로 계산할 수 있다.

$$\widehat{r_{A,i}} \simeq \widehat{r'_{A,i}} + \overline{b_A} \quad (10)$$

최종적으로 test dataset에 있는 원평점과 이 예측 평점의 예측 신뢰도를 비교함으로써 제안한 추천시스템을 검증하였다.

3.3 CNN을 통한 데이터 학습

3.3.1 데이터 수집 및 준비

수집하고자 하는 데이터는 [사용자 ID, 영화 ID, 평점, 댓글]이다. 추천시스템에서 널리 쓰이는 MovieLens 데이터는 [사용자 ID, 영화 ID, 평점] 정보만 있고

댓글 정보가 없다. 감성 분석에서 많이 사용하는 Pang/Lee 데이터 및 SST 데이터들은 [댓글, 평점]만 있고 사용자, 영화 정보가 없다. 이들 데이터들은 모두 선행 논문에서 사용하여 성능 비교에 용이한 장점이 있지만, 정보가 부족하다.

이에 본 연구에서는 Amazon의 영화에 대한 모든 댓글과 평점 정보를 담고 있는 Stanford 대학교의 SNAP 프로젝트에서 수집한 데이터를 사용하였다. 하지만 이 데이터는 1997년 8월부터 2012년 10월까지의 Amazon의 모든 영화(253,059)편에 대한 사용자(889,176)명의 댓글 7,911,684개를 담고 있으며 크기도 9.3GB로 너무 방대하다. 따라서 다음과 같은 추출 기준에 의거하여 연구용 데이터를 추출하였다.

- 2009년 1월 ~ 2012년 10월
- 댓글 단어가 2개 이하이거나 100개 이상인 댓글 제외
- 댓글이 20개 이하인 영화와 사용자 제외

이 기준에 의하여 추출된 데이터는 131,470개의 댓글과 4,668명의 사용자와 2,365개의 영화를 포함한다. 이렇게 얻어진 데이터로부터 적당한 비율을 학습데이터(training dataset)로, 나머지를 평가데이터(test dataset)로 사용하였다.

3.3.2 미리 훈련된 word vectors

Word embedding은 단어들을 vector로 매핑시켜주는 것을 말하며, vector의 거리를 계산하여 단어들 간의 유사도를 알 수 있다. Word embedding은 단어간의 거리를 기반으로 문장의 유사성을 추출하는데 핵심적인 역할을 하며, CNN 문장 분석에서 매우 중요하다. 따라서 정교한 단어들간의 거리를 가지기 위해서는 가능하면 빅데이터로 훈련되어야 한다. 구글에서는 약 1,000억 개의 단어들로부터 vector들을 이미 훈련시킨 word2vec을 배포하고 있다. 구글의 word2vec은 300차원

vector들로 약 300만 개의 단어들을 가지고 있다. 따라서 본 연구에서는 빅데이터로 미리 훈련된 구글의 word2vec을 사용하였다.

3.3.3 CNN

Kim, Y.(2014)는 CNN을 이용하여 문장에서의 지역적 정보를 보존하되, $k \times p$의 filter(1 <= k <= n)을 사용하여 convolution layer을 구성하는 방법을 제시하였다[20]. <그림 3.>은 n = 4, p = 6인 경우 문장 테이블의 예시를 보여준다.

	Vector1	Vector2	Vector3	Vector4	Vector5	Vector6
Word1	1.3	-2.4	-1.2	-5.4	-2.6	4.5
Word2	6.7	8.6	2.3	-9.8	0.6	0.9
Word3	24.1	3.4	-6.6	8.8	9.9	-0.8
Word4	-19.2	-0.6	-6.3	5.4	7.6	3.2

<그림 3.> n=4, p=6인 경우 문장 테이블의 예시

본 연구에서는 Kim, Y.의 문장 분류용 CNN을 기본 구조로 사용하였으며, 평점의 경우 5차원의 배열로 표현하였다. 예를 들어 1점의 경우 [1, 0, 0, 0, 0]의 배열을 갖고 4점의 경우 [0, 0, 0, 1, 0]의 배열을 갖는다. 이후 Relu와 max-pooling을 통하여 지역적인 대푯값을 추출하고 마지막으로 fully connected layer에서 5개의 클래스 중 가장 적합한 클래스를 선택하였다. 크로스엔트로피오차를 구하고 back propagation 작업을 수행하여 convolution layer filter의 parameter 값을 업데이트하였다. 필터의 크기는 3, 4, 5를 사용하였고, dropout값은 0.5, L2 정규화 람다는 0.1을 사용하였으며, batch size는 128이고, 150 epoch를 학습하여,

accuracy 0.9247까지 학습시켰다.

수집된 131,470개의 데이터에서 95%인 124,897개는 학습데이터(training dataset)로 사용하였고, 나머지 5%인 6,573개는 평가 데이터(test dataset)으로 사용하였다.

3.4 추천 알고리즘의 데이터 학습

본 논문에서는 추천 알고리즘으로 3rd party에서 제공하는 Python scikit for recommendation library, 줄여서 surprise를 사용하였다. 이 라이브러리에는 k-NN, SVD, Baseline 등 다양한 추천 알고리즘들이 구현되어있다. 또한, surprise는 Pearson correlation, cosine 등 다양한 유사도 측정 함수들도 제공한다. 기본적으로 k-NN 알고리즘과 피어슨 및 코사인 유사도 함수를 기본으로 실험을 하였고, 더불어 다른 여러 가지 알고리즘도 비교 평가하였다.

Ⅳ. 연구 결과

에러의 측정에는 RMSE와 MAE를 모두 사용하였다. 실험 결과, 원 평점의 검증 데이터에 대한 MAE보다 비편향 평점에 대한 MAE가 대다수의 추천 알고리즘들에 대해서 낮게 나타났다. 즉, 비편향 평점의 경우 원래의 평점에 비해 평가자의 주관적인 성향이 제거됨으로써 평점의 왜곡이 제거되어 더 작은 오차로 나타나게 되었다.

4.1 k-NN 알고리즘에서의 결과 비교

같은 데이터셋에 대해서 기존의 추천 알고리즘만 사용했을 때의 결과와 본 논문에서 제안한 추천 알고리즘의 결과를 상호 비교하였다.

k-NN 추천 알고리즘을 사용한 실험 결과는 <표 1.>과 같다. Pearson 유사도 함수를 사용한 경우, 기존 k-NN 알고리즘만을 사용하였을 때 RMSE는 0.5080, MAE는 0.3180이었고, 제안된 알고리즘을 사용하였을 때 RMSE는 0.4981로 낮아졌고, MAE도 0.3065로 낮아졌다.

Cosine 유사도 함수를 사용한 경우에도, 기존 k-NN 알고리즘만을 사용했을 때 RMSE는 0.5215, MAE는 0.3257이었으나, 제안된 알고리즘을 사용하였을 때는 RMSE는 0.4997로 낮아졌고, MAE도 0.3071로 낮아졌다. 이것은 CNN을 통한 평점으로 이웃을 찾았을 때 좀 더 가까운 이웃이 찾아진 것을 의미한다.

<표 1.> k-NN 추천 알고리즘을 사용하였을 때의 RMSE와 MAE 값

추천 알고리즘		RMSE		MAE	
알고리즘	유사도 함수	기존 알고리즘	제안된 알고리즘	기존 알고리즘	제안된 알고리즘
k-NN	Pearson	0.5080	**0.4981**	0.3180	**0.3065**
	Cosine	0.5215	**0.4997**	0.3257	**0.3071**

4.2 다른 추천 알고리즘과의 비교

k-NN의 변종 알고리즘들로 각각의 사용자의 평균 평점을 고려한 k-NN with Means, 각각의 사용자에 대해 z-score로 정규화한 k-NN with ZScore, Baseline 평점을 고려한 k-NN Baseline 알고리즘들과도 비교한 성능 값은 <표 2.>와 같다.

단순히 기존 추천알고리즘만 사용했을 때보다, 비편향 평점을 사용한 경우에 RMSE 및 MAE 보두 개선된 값을 보여주었다.

<표 2.> k-NN 변종 알고리즘들을 사용하였을 때의 RMSE와 MAE 값

추천 알고리즘		RMSE		MAE	
알고리즘	유사도 함수	기존 알고리즘	제안된 알고리즘	기존 알고리즘	제안된 알고리즘
k-NN Baseline	Pearson	1.0182	**0.9837**	0.7321	**0.6894**
	Cosine	1.0121	**0.9735**	0.6866	**0.6460**
k-NN with Means	Pearson	0.3697	**0.3640**	0.1057	**0.1042**
	Cosine	0.3905	**0.3653**	0.1303	**0.1196**
k-NN ZScore	Pearson	0.3474	**0.3436**	0.0962	**0.0958**
	Cosine	0.3930	**0.3651**	0.1325	**0.1217**

k-NN 알고리즘 외에 다른 알고리즘으로 CNN에서 얻어진 평점을 대입한 결과는 <표 3.>과 같다. Normal Predictor(트레이닝 데이터셋이 정규분포라고 가정하고 random 평점을 예측), Baseline(각 사용자 및 영화에 대한 baseline estimate를 예측), Slope One의 경우에는 성능이 개선되었다. 하지만 Co-Clustering 및 Matrix Factorization-base 알고리즘 종류로 SVD(Singular Value Decomposition), NMF(Non-negative Matrix Factorization)의 경우에는 비슷하거나 개선이 되지 않았다.

<표 3.> 다른 추천 알고리즘들을 사용하였을 때의 RMSE와 MAE 값

추천 알고리즘	RMSE		MAE	
	기존 알고리즘	제안된 알고리즘	기존 알고리즘	제안된 알고리즘
Normal Predictor	1.3097	**1.2453**	0.9145	**0.8575**
Baseline	0.5194	**0.4978**	0.3279	**0.3093**
Slope One	0.4032	**0.3825**	0.1282	**0.1188**
Co-Clustering	0.4195	0.4849	0.1778	0.2479
NMF	0.3346	0.3626	0.2406	0.2537
SVD	0.3382	**0.3364**	0.1658	**0.1619**
SVD++	0.2940	0.2968	0.1211	**0.1180**

Ⅴ. 결론

본 연구에서는 사용자에 따라 정량적인 기준이 다를 수 있는 평점만으로 추천 결과를 예측하는 기존의 협업 필터링을 개선하기 위해, 사용자의 의도나 감정이 좀 더 자세하게 표현되어 있는 댓글을 활용하여 협업 필터링을 하는 추천 알고리즘을 제안하였다.

사용자가 댓글에 표현하는 감정이나 느낌의 정도에 비해 평점은 주관적 기준에 좌우된다. 따라서 댓글과 평점으로 CNN을 학습시켜 감성분석을 하면 비슷한 댓글에 대해 비슷한 평점이 나오며, 이 비편향 평점을 기존의 추천 알고리즘에 접목시켜 예측 평점을 산출하였다. 기존의 추천 알고리즘 단독의 경우와, CNN과 추천 알고리즘을 결합한 모델의 결과를 비교한 결과, 여러 알고리즘에서 개선된 성능을 보였다.

본 연구에서는 약 13만건의 리뷰를 이용하고, 빅데이터로 이미 학습된 word2vec을 사용하여 CNN으로 학습하였지만, 댓글에 대한 의미를 완벽하게 학습하였다고 하기에는 한계를 지닌다. 이것은 유사도의 에러를 통해 잘못된 이웃을 선정하는 결과가 될 수 있다. 따라서 양질의 빅데이터 수집 및 CNN 외에도 LSTM 같은 여러 방법의 아키텍처에 대한 연구를 수행시킨다면 댓글의 감성을 더 정확하게 학습시킬 수 있을 것이다.

References

[1] Reimer, U., Hahn, U., (1997), A formal model of text summarization based on condensation operators of a terminological logic, *Intelligent Scalable Text Summarization*.

[2] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., (1994, October), GroupLens: an open architecture for collaborative filtering of netnews, *In Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175-186), ACM.

[3] Oh, B. H., Yang, J. H., Lee, H. J., (2014), A hybrid recommender system based on collaborative filtering with selective utilization of content-based predicted ratings, *J. KIISE: Softw. and Appl, 41*(4), 289-294.

[4] Kim, M. J., Park, D. S., Hong, M., Lee, H., (2015), Personalized movie recommendation system using context-aware collaborative filtering technique, *KIPS Transactions on Computer and Communication Systems, 4*(9), 289-296.

[5] 이오준, 백영태, (2014), 협업 필터링 추천 시스템을 위한 데이터 신뢰도 기반 가중치를 이용한 하이브리드 선호도 예측 기법, *한국컴퓨터정보학회논문지, 19*(5), 61-69.

[6] 김민정, 박두순, 홍민, 이화민, (2015), 상황기반과 협업 필터링 기법을 이용한 개인화 영화 추천 시스템, *정보처리학회논문지 컴퓨터 및 통신시스템, 제4권*, 제9호, 289-296.

[7] Lee, J. S., Park, S. D., (2007), Performance improvement of a movie recommendation system using genre-wise collaborative filtering, *Journal of*

Intelligence and Information Systems, 13(4), 65-78.

[8] Lee, S. H., Park, S. H., (2010), Accuracy improvement of a collaborative filtering recommender system, *Journal of the Korea Safety Management and Science, 12*(1), 127-136.

[9] Choeh, J. Y., Lee, S. K., Cho, Y. B., (2013), Applying Rating Scores Reliability of Customers to Enhance Prediction Accuracy in Recommender System, *The Journal of the Korea Contents Association, 13*(7), 379-385.

[10] 박찬수, 황태규, 홍정화, 김성권, (2015), Preference Difference Metric 을 이용한 아이템 분류방식의 추천알고리즘, *정보과학회 컴퓨팅의 실제 논문지, 21*(2), 121-125.

[11] Lee, H. C., Lee, S. J., Chung, Y. J., (2006), The effect of co-rating on the recommender system of user base, *Journal of the Korean Data and Information Science Society, 17*(3), 775-784.

[12] 박석인, 김택헌, 류영석, 양성봉, (2002), 추천 시스템의 예측 정확도 향상을 위한 전처리 방법, *한국정보과학회 학술발표논문집, 29*(1B), 247-249.

[13] 최용준, 김수철, 박찬수, 김성권, (2013), 노이즈 평점을 필터링하는 아이템기반 추천시스템, *한국정보과학회 학술발표논문집*, 291-293.

[14] 김현경, 김현진, 박상현, (2014. 6.), 평점 정규화를 이용하여 사용자 평가 성향을 반영한 영화 추천 방법, *한국정보과학회 한국컴퓨터종합학술대회 논문집*, 710-712.

[15] 류신, (2015), 오피니언 편향성 보정을 통한 영화 평점 시스템 신뢰성 향상 방안, *국민대학교 석사학위논문*.

[16] 김진현, 김정현, 정성욱, 강신재, (2017), SNS 게시글과 감성분류에 기반한

다단계 노래 추천 시스템, *예술인문사회융합멀티미디어논문지, 7*, 283-290.

[17] Hu, M., Liu, B. (2004, August), Mining and summarizing customer reviews, *In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 168-177), ACM.

[18] Theresa Wilson, Janyce Wiebe, Paul Hoffmann, (2005), Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis, *Preceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (HLT/EMNLP 2005), 347-354.

[19] 김유영, 송민, (2016), 영화 리뷰 감성분석을 위한 텍스트 마이닝 기반 감성 분류기 구축, *J Intell Inform Syst 2016 September 22*(3), 71-89.

[20] Kim, Y., (2014), Convolutional neural networks for sentence classification, *arXiv preprint arXiv:1408.5882*.

[21] Nal Kalchbrenner, Edward Grefenstette, Phil Blunsom, (2014), A Convolutional Neural Network for Modelling Sentences, *Preceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 655-665.

[22] Bei-Bei CUI, (2017), Design and Implementation of Movie Recommendation System Based Knn Collaborative Filtering Algorithm, *ITM Web of Conferences 12*.

Summary

Collaborative Filtering Recommendation System Correcting User Bias Based on Comments-Rating Relationship

When NN(Neural Network) is used in sentimental analysis, the positive/negative rating that the comment holds can be corresponded into a value of 1 to 5. The returned value can stand how similar the target value and the predicted value are. When the learning process is well done by big data, the target value and the predicted value will be same for the majority of the data. However, minority will have a difference, and it can be interpreted as a bias of the particular user. In this study, similar users are defined as users whose comments are similar. Users who have similar comments have similar ratings obtained by NN. Therefore, using the obatained, or unbiased rating at the recommendation system is a better way compared to using the rating that the users have given.

Figure 1. is a diagram of the recommendation system suggested by this study.

CNN is trained by users' comments, and the unbiased rating trains the recommendation system. Detailed procedure is shown in Figure 2.

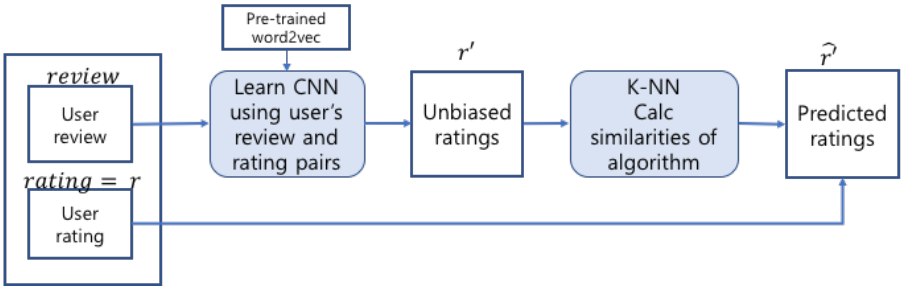


Figure 1. Diagram of the recommendation system

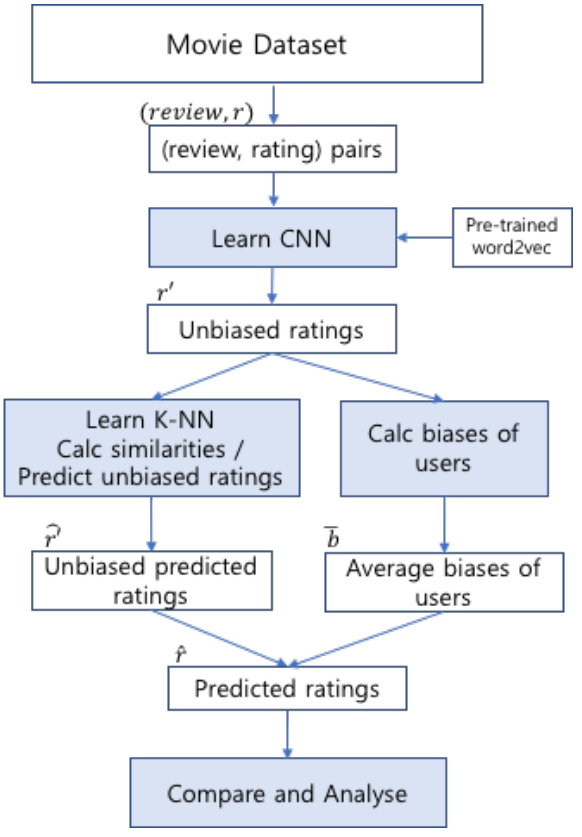


Figure 2. Procedure of the recommendation system

In this study, data collected by SNAP Project, Stanford University was used. The data contains information of comments and points about the movies from Amazon.

At CNN train step, CNN was trained by entire comment $review_A$ and rating r_A that user A has given. 3, 4, 5 were used as the filter size. Dropout value was 0.5, L2 and regularization lambda was 0.1. Batch size was 128, and 150 epoch was trained, resulting the accuracy up to 0.9247.

Then by the trained CNN, new rating, or unbiased rating r'_A was obtained. If A has a bias, let bias for an each comment is b_A. Then, r'_A can be thought as b_A added to r_A and can be written as (1).

$$r'_A \simeq r_A - b_A \tag{1}$$

Then, average bias of an user can be written as (2).

$$\overline{b_A} = \frac{\sum_{i}^{n_A}(r'_{A,i} - r_{A,i})}{n_A} \tag{2}$$

The suggested recommendation system was trained by the unbiased rating of all users. The system calculated the neighbors by the Pearson-correlation written as (3).

$$w'(A,B) = \frac{\sum_{i=1}^{q}(r'_{A,i} - \overline{r'_A})(r'_{B,i} - \overline{r'_B})}{\sqrt{\sum_{i=1}^{q}(r'_{A,i} - \overline{r'_A})^2 \sum_{i=1}^{q}(r'_{B,i} - \overline{r'_B})^2}} \tag{3}$$

Predicted unbiased rating can be obtained by the Pearson-correlation above. Predicted unbiased rating of movie i of user A can be written as (4).

$$\widehat{r'_{A,i}} = \overline{r'_A} + \frac{\sum_{j=1}^{k} w'(A,j)(r'_{j,i} - \overline{r'_j})}{\sum_{j=1}^{k} |w'(A,j)|} \quad (4)$$

Predicted biased rating was calculated as the predicted unbiased rating added to average bias of the user. It can be written as (5).

$$\widehat{r_{A,i}} \simeq \widehat{r'_{A,i}} + \overline{b_A} \quad (5)$$

Several different recommendation algorithms in surprise, a 3rd party library in Python, were used for the comparison.

The result when k-NN algorithm was used is shown in Table 1. When Pearson-correlation was used, RMSE was 0.5080 and MAE was 0.3180 at the original algorithm, but RMSE was 0.4981 and MAE was 0.3065 at the suggested algorithm. Similarly, when cosine was used, RMSE and MAE both had decreased. This means that when CNN is used at finding neighbors, more similar neighbors were found.

Table 1. RMSE and MAE values when k-NN algorithm was used

Recommendation system		RMSE		MAE	
Algorithm	Similarity function	original	suggested	original	suggested
k-NN	Pearson	0.5080	**0.4981**	0.3180	**0.3065**
	Cosine	0.5215	**0.4997**	0.3257	**0.3071**

RMSE and MAE were decreased for all other algorithms based on k-NN, such as

KNN with Means, KNN with ZScore, KNN baseline, shown in Table 2.

Table 2. RMSE and MAE values of other k-NN algorithms

Recommendations system		RMSE		MAE	
Algorithm	Similarity function	origianl	suggested	original	suggested
k-NN Baseline	Pearson	1.0182	**0.9837**	0.7321	**0.6894**
	Cosine	1.0121	**0.9735**	0.6866	**0.6460**
k-NN with Means	Pearson	0.3697	**0.3640**	0.1057	**0.1042**
	Cosine	0.3905	**0.3653**	0.1303	**0.1196**
k-NN ZScore	Pearson	0.3474	**0.3436**	0.0962	**0.0958**
	Cosine	0.3930	**0.3651**	0.1325	**0.1217**

Finally for other recommendation algorithms, suggested algorithm was more efficient for Normal Predictor, Baseline, and Slope One. However for Co-Clustering, SVD, and NMF, the efficiency was similar or lower. It is shown in Table 3.

Table 3. RMSE and MAE values when other recommendation algorithms were used

Recommendation algorithm	RMSE		MAE	
	original	suggested	original	suggested
Normal Predictor	1.3097	**1.2453**	0.9145	**0.8575**
Baseline	0.5194	**0.4978**	0.3279	**0.3093**
Slope One	0.4032	**0.3825**	0.1282	**0.1188**
Co-Clustering	0.4195	0.4849	0.1778	0.2479
NMF	0.3346	0.3626	0.2406	0.2537
SVD	0.3382	**0.3364**	0.1658	**0.1619**
SVD++	0.2940	0.2968	0.1211	**0.1180**

In this study, a recommendation system was suggested in order to improve the

collaborative filtering. Original collaborative filtering, which uses only the rating, doesn't consider the users' bias. Therefore, comments were used at the system because comments contain the emotion and purpose of the user. Comments and ratings were used to train CNN and the obatined rating, or the unbiased rating was collaborated the original recommendation algorithms. Through a comparison between the original algorithms and the suggested algorithm, the suggested algorithm performed better.