JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 30 No. 7, pp. 171-179, July 2025
https://doi.org/10.9708/jksci.2025.30.07.171

# Item-attribute-based Semantic Similarity for Data Sparsity in Collaborative Filtering

Soojung Lee*

*Professor, Dept. of Computer Education, Gyeongin National University of Education, Anyang, Korea

## [Abstract]

Neighbor-based collaborative filtering is a representative recommendation algorithm that predicts user preferences based on user or item similarity and it is known for its simplicity and interpretability. However, this approach faces the data sparsity problem, in which the accuracy of similarity computation deteriorates when user rating data is insufficient. To address this limitation, this study proposes novel symmetric and asymmetric item-based similarity measures that rely solely on item attribute information. The proposed measures do not require the number of co-rated items or the ratings distribution, but they compute semantic similarity based on common item attributes, which allows for robust performance even under sparse conditions. Experimental results using two public datasets demonstrate that the proposed method outperforms existing techniques in terms of average precision and coverage. The method is applicable even in systems with no explicit ratings provided, and if item attributes are static, similarity computation needs to be performed only once, offering strong system scalability advantages.

▸ **Key words**: Similarity Measure, Data Sparsity Problem, Item-based Collaborative Filtering, Recommender System

## [요 약]

이웃 기반 협력 필터링은 사용자 또는 항목 간의 유사도를 바탕으로 선호를 예측하는 대표적인 추천 알고리즘으로서, 구현이 간단하고 해석이 용이하다는 장점을 가진다. 그러나 이 방식은 사용자의 평가 데이터가 충분하지 않을 경우에 유사도 계산의 정확도가 저하되는 데이터 희소성 문제에 직면하게 된다. 본 연구에서는 이러한 한계를 극복하기 위해 항목의 속성 정보만을 활용하는 새로운 대칭 및 비대칭의 항목 기반 유사도 척도를 제안한다. 제안 방법은 사용자 간 공통 평가 항목 수 또는 평가치의 분포를 필요로 하지 않으며, 항목 간 공통 속성에 기반한 의미적 유사도를 산출하여 희소한 환경에서도 견고한 성능을 보인다. 두 가지 공개 데이터셋을 활용한 실험 결과, 평균 정밀도 및 커버리지 측면에서 기존 기법보다 우수한 성능을 확인하였다. 제안 척도는 사용자의 직접적인 평가치를 제공하지 않는 시스템에서도 적용 가능하며, 항목 속성 정보가 고정적일 경우에 유사도 계산을 1회만 수행해도 되므로 시스템 확장성이 우수하다는 잇점이 있다.

▸ **주제어**: 유사도 척도, 데이터 희소성 문제, 항목 기반 협력 필터링, 추천 시스템

• First Author: Soojung Lee, Corresponding Author: Soojung Lee
*Soojung Lee (sjlee@gin.ac.kr), Dept. of Computer Education, Gyeongin National University of Education
• Received: 2025. 06. 23, Revised: 2025. 07. 07, Accepted: 2025. 07. 16.


http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

## I. Introduction

개인화 추천 시스템은 디지털 환경에서 사용자 경험을 최적화하기 위한 핵심 기술로 자리 잡았다. 특히 넷플릭스, 아마존, 유튜브 등 다양한 온라인 서비스에서 사용자와 항목 간의 상호작용을 분석하여 개인의 취향에 맞는 콘텐츠를 제공하는 데 있어 협력 필터링(Collaborative Filtering, CF)은 가장 널리 사용되는 방법 중 하나이다. CF는 기본적으로 사용자-항목 평가치 행렬을 바탕으로 작동하며, 이 행렬에서 유사한 사용자 또는 유사한 항목을 찾아 추천을 생성한다. 이러한 접근은 사용자의 명시적 평가치뿐 아니라 암묵적 행동(예: 클릭, 구매, 시청 시간 등)을 활용할 수 있어 다양한 도메인에서 높은 실용성을 보여주고 있다[1][2].

하지만 CF는 데이터 희소성(data sparsity) 문제에 본질적으로 취약하다는 한계를 갖는다[3][4]. 실제 시스템에서는 대부분의 사용자들이 전체 항목 중 극히 일부에 대해서만 평가하거나 상호작용하기 때문에, 사용자-항목 행렬은 대부분 비어 있게 된다. 이로 인해 유사도 계산이 신뢰성을 잃게 되고, 결과적으로 추천의 품질이 저하되는 문제가 발생한다. 특히 신규 사용자 또는 신규 항목의 경우에는 거의 데이터가 존재하지 않아 추천이 어려운 콜드스타트 문제(cold-start problem)까지도 유발한다[5].

최근의 연구에서는 이러한 문제를 극복하기 위해 외부 도메인 지식이나 항목의 메타데이터를 유사도 계산에 통합하는 방식이 주목받고 있다. 예컨대, 콘텐츠 기반 정보(제목, 장르, 감독, 키워드 등)를 활용하거나, 사회적 관계 정보, 리뷰 내용, 지식 그래프 등을 통합하는 시도들이 나타나고 있다[6][7][8]. 또한 행렬 분해(matrix factorization), 딥러닝 기반 모델, 외부 지식 또는 콘텐츠 정보를 활용한 하이브리드 필터링(hybrid filtering), 클러스터링을 통한 최적화 등의 방식도 연구되었다[1][9][10][11][12]. 하지만 이러한 방법은 계산량이 크거나 복잡한 학습 구조를 필요로 하며, 설명 가능성이 낮다는 단점이 존재한다. 이에 따라 최근에는 협력 필터링의 구조는 유지하되, 유사도 계산을 개선하여 희소성을 보완하려는 시도가 주목받고 있다.

본 연구에서는 항목의 속성 정보를 활용한 새로운 유사도 척도를 제안하였다. 희소한 평가 데이터 환경에서 성능을 위하여 개발된, 사용자의 공통 평가 항목 수 또는 평가치 확률 분포 등을 활용하는 기존의 대표적인 척도인 Jaccard 계수나 Bhattacharyya 계수에 비하여[13][14][15], 제안 방법은 이러한 정보 대신에 두 항목 간의 공통적인 속성 정보만을 활용하여 유사도를 측정하는 항목 기반 CF 방식이다. 두 사용자의 공통 평가 항목 개수에 의존하지 않으므로 공통으로 평가한 항목이 전혀 없더라도 유사도 산출이 가능하며, 하나의 항목이 속성값을 여러개 갖는 경우에 산출 근거 데이터가 많아지므로 더욱 유리한 방법이다. 두 가지의 서로 다른 희소성 수준의 공개 데이터셋을 이용한 성능 실험 결과, 제안 방법은 평균 정밀도와 커버리지 측면에서 우수하였고 특히 더욱 희소한 환경에서 기존 방법들의 성능을 크게 능가하였다. 따라서 Jaccard 계수나 Bhattacharyya 계수와 마찬가지의 활용 형태, 즉, 독립적 또는 기존의 항목 기반 CF 알고리즘과의 통합 형태로서 유사도를 산출하여 향상된 CF 성능을 기대할 수 있다.

논문의 구성은 다음과 같다. 2절에서는 본 논문 주제와 관련된 기존 연구 결과를 소개한다. 3절에서 제안 방법을 소개하고 4절에서는 성능 실험 결과를 제시하며, 5절에서 논문의 결론을 맺는다.

## II. Related Works

전통적으로 유사도가 높은 이웃 기반의 CF 시스템에서는 추천 성능의 향상을 위해 사용자 또는 항목 간의 유사도 측정을 위한 다양한 척도가 개발되었다. 대표적인 척도는 코사인 유사도(Cosine Similarity), 피어슨 상관 계수(Pearson Correlation), 평균자승차이(Mean Squared Difference, MSD) 등이다. CF는 항목들에 대한 사용자의 과거 평점을 바탕으로 유사한 사용자 또는 항목을 찾아 이들의 선호 항목을 기반으로 추천 결과를 결정하기 때문에 시스템의 성능은 유사도 계산의 정확도에 크게 의존한다.

대개 유사도는 사용자-항목 평가치 행렬을 기반으로 계산되는데, 이 행렬은 실제로 매우 희소하여, 공통적으로 평가한 항목의 수가 적은 경우 유사도의 신뢰도가 저하되는 문제가 빈번히 발생한다. 데이터 희소성 문제의 해결을 위하여 다양한 기법들이 제안되었는데, [4]에서는 희소한 평점 행렬의 밀도를 증가시키기 위한 목적으로 사용자와 항목을 동시에 클러스터링하는 Bi-Separated Clustering과 각 클러스터의 평균값으로 결측값을 보완하는 Bi-Mean Imputation 기법을 제안하였다. 한편 [10]의 연구에서는 오토인코더, 딥 신경망, 하이브리드 모델 등을 활용한 딥러닝 기반 CF가 데이터 희소성 문제를 완화하는 데 효과적임을 보고하였다. [8]은 하이퍼그래프 구조를 활용한 Hypergraph Contrastive CF 모델을 제안하였으며

이 모델은 고차원의 사용자-항목 관계를 효과적으로 학습하여, 희소한 상호작용 데이터에서도 높은 추천 성능을 나타낸다고 하였다.

희소 데이터 환경을 위하여 공통의 평가 항목 개수를 기준으로 유사도를 측정하는 방법들이 개발되어 널리 활용되었는데, 대표적으로 Jaccard 계수는 사용자의 평가치를 활용하는 대신에 두 사용자가 평가한 공통항목의 상대적 개수를 기준으로 그들 간의 유사도를 측정한다. 이 계수는 희소한 평가 데이터 환경에서 매우 효율적인 방법으로 알려져 있다[13][14][16]. 또한 평가치 크기를 반영하는 다른 여러 유사도 척도와 통합하여 활용되는 등 CF 시스템에서의 활용도 및 성능 향상 기여도가 높다. 예를 들어, [16]에서는 Jaccard 계수와 여러 다른 유사도 척도 - 피어슨 상관도, 코사인 유사도 등 - 들을 각각 곱한 유사도 산출 방법을 제시하였고, 실험을 통하여 Jaccard 계수가 다른 수치 기반 척도들의 성능 향상에 효과적으로 기여할 수 있음을 증명하였다. [13]에서는 두 사용자가 평가한 공통항목들에 대한 평가치 뿐만 아니라 사용자가 부여한 모든 평가치들을 고려하여 관련된 유사한 이웃들을 분류하고 더욱 신속하게 추천을 생성하는 두 가지의 간단하면서도 효과적인 유사도 모델을 개발하였으며, 제안한 관련 Jaccard 유사도(relevant Jaccard similarity)는 다른 기존 유사도 모델보다 더 정확하고 효과적으로 추천을 생성한다고 하였다. 항목 간의 유사도 산출 방법으로서, [17]에서는 Ochiai라는 구조적 기반과 평가치 기반을 통합하는 유사도 산출 방법을 제안하였고 이는 모델 기반의 CF 방법보다 예측 성능면에서 우수함을 입증하였다.

이밖에도 사용자 평가치를 활용하는 대신에 평가치의 분포를 유사도 산출에 활용하려는 연구가 진행되어 왔다. [18]에서는 공통 평가치 항목의 제약에서 벗어나 새로운 유사도 측정 기법으로서 평가치의 확률 밀도 분포를 기반으로 항목 간의 관계를 식별하는 Kullback-Leibler divergence 기반의 항목 간 유사도 측정법을 제시하였으며, 이는 모든 평가치를 활용하기 때문에 희소성 문제에 대한 좋은 해결책이 될 수 있다고 하였다. [19]의 연구에서는 인기도와 다양성 간의 트레이드 오프 관계를 유지하고 단일 실행으로 여러 상충 관계 해결책을 도출하기 위해 다목적 최적화(multi-objective optimization) 기법을 사용하였다. 이 방법은 기존의 비선형 유사도 계산 모델에 Bhattacharyya 계수[15]를 통합하여 새로운 유사도 모델을 생성하였고 기존의 평점 평가 기법의 예측 정확도를 향상시켰다.

데이터 희소성을 해결하기 위한 또다른 전략으로서 평가치나 평가 개수 이외에 사용자의 평가 항목으로부터 문맥을 파악하여 유사도에 반영하는 의미적 유사도(semantic similarity) 관련 연구가 진행되었다. [20]에서는 평가치의 예측 정확도를 높이기 위하여 인접 이웃들의 평가치를 활용하는 대신에 항목 속성에 대한 사용자의 흥미 행렬을 구하고, 가장 유사한 이웃의 유사한 특징을 가진 평점 항목을 사용하였다. [21]은 항목 간의 유사성을 평가하고 궁극적으로 정확한 추천을 생성하기 위한 온톨로지 기반의 의미 유사도 측정법을 제안하였으며, 생성된 추천의 품질을 향상시키기 위해 새로운 의미 유사도 측정법과 표준 항목 기반 CF를 결합한 새로운 의미 강화 하이브리드 추천 방식을 제안하였다.

## III. Proposed Methodology

### 1. Description

본 연구는 희소 데이터 환경에서의 협력 필터링 시스템을 위한 항목 기반의 유사도 측정 방법을 제안한다. 제안 방법은 사용자의 항목 평가치 또는 개수를 활용하는 대신에 항목에 대한 의미적 속성을 기반으로 한다. 기본적인 아이디어는 두 항목이 공통으로 보유한 속성 개수가 많을수록 그들 간의 유사도는 증가하도록 한다.

각 항목은 의미적 속성 벡터 V로 표현하며 $V=(v_1,v_2, ..., v_F)$로 정의한다. 이 때 F는 벡터의 차원이며 $v_f$, f=1, ..., F는 각 차원에 해당하는 벡터 요소값을 말한다. 이는 기존의 많은 연구에서 사용자의 평가치들로 구성된 벡터를 사용한 것과 대비된다. 만약 각 항목이 영화를 나타내며, 속성 벡터가 영화의 주연 배우를 나타낸다면, 데이터셋이 보유한 모든 영화 배우 수가 F 값이며, $v_f$는 이들 중 한 명의 배우에 대응한다. 예를 들어, 영화 항목 x의 주연 배우가 단 한 명이며 벡터의 첫 번째 요소에 해당한다면 $V_x=(1, 0, ...., 0)$이다. 본 연구에서 활용한 영화 데이터셋인 MovieLens의 경우, 항목의 속성은 장르, 감독, 제작년도 중 하나로 선정할 수 있다. 장르일 경우에 데이터셋의 전체 장르수는 18개이므로 벡터 길이는 18이며, 영화 항목 Toy Story에 해당하는 장르는 3(Animation), 4(Children's), 5(Comedy)이므로 벡터 (0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)로 나타낸다.

제안 유사도 척도는 두 항목 벡터에서 공통적인 1의 요소값을 반영한다. 만약 공통 요소가 하나도 없다면 0의 유

사도를, 공통 요소 1의 위치와 개수가 일치한다면 유사도는 1이 된다. 본 연구에서는 두 가지의 유사도, 즉, SMJAC(SeMantic JACcard)과 SMJACA(SeMantic JACcard Asymmetric)를 제안한다. 전자는 두 벡터 간의 공통 요소 1의 상대적 개수로 정의하며 항목 i와 j에 대하여 SMJAC(i, j)=SMJAC(j, i)로서 대칭이다. 이 방법은 항목의 공통평가개수 대신에 속성의 공통 요소값 개수를 활용했을 때의 성능 개선 여부를 파악하기 위하여 개발한 것이다. 더 나아가서 후자 유사도인 SMJACA는 비대칭, 즉, SMJACA(i, j)과 SMJACA(j, i)는 반드시 동일하진 않은 값으로 산출될 수 있다. 이는 공통 요소 개수를 반영하는데 있어서 상대적 기준을 각 항목별로 다르게 설정하려는 취지이다. 예로써, $V_i$=(1, 0, 0, 0)이고 $V_j$=(1, 1, 1, 0)이라 하자. 이 때 두 벡터 간의 공통 요소 1의 값은 첫 번째 요소에 해당하는 값으로서 하나이며, 항목 i의 관점에서는 100% 일치하지만, 항목 j의 관점에서는 세 개의 1 요소값 중 한 개가 일치한 것이므로, 각 항목의 관점을 기준으로 하여 유사도는 서로 다르게 책정되어야 한다는 판단에 의거한다. 구체적으로 수치를 계산해 보면, SMJACA(i, j)=1.0이고 SMJACA(j, i)=0.333이다.

표 1은 제안 유사도와 이를 활용한 사용자의 미평가 항목에 대한 예측치 산출 알고리즘이다. 표 2에 알고리즘에서 사용한 기호 및 의미를 제시하였다. 1부터 10줄까지 모든 항목 쌍 간의 유사도, SMJAC과 SMJACA를 구하였다. 구체적으로 우선 두 항목이 동일한 경우에 이들 간의 유사도는 1로 정의한다(3줄). 그렇지 않은 경우, 항목 i와 j 각각의 벡터 요소값 1의 개수와(6줄, 7줄) 공통으로 보유한 요소값 1의 개수를 구한다(8줄). 이를 기반으로 9줄과 10줄의 공식과 같이 유사도를 산출한다. 유사도 산출이 완료되면 각 항목에 대하여 유사한 항목들의 집합인 인접 이웃(nearest neighbor) 집합이 결정된다. 현 사용자가 미평가한 항목에 대한 시스템의 예측 평가치는 각 인접 이웃과의 유사도를 가중치로 하여 14줄의 공식에 의하여 산출한다.

제안 방법은 기존의 많은 유사도 척도들과는 달리 사용자의 항목 평가치나 평가 개수에 의존하지 않으므로 데이터 희소성 문제 해결에 상당한 도움이 되며, 사용자의 직접적인 평가치를 제공하지 않는 시스템에서도 적용 가능하다. 또한 항목의 속성 및 속성값이 유동적이지 않는 경우에 모든 항목 간 유사도 산출은 단 1회만 실행하면 되므로, 사용자들의 추가 유입으로 인한 시스템 확장성 문제의 해결도 용이하다.

Table 1. Algorithm for the proposed similarity measure and rating prediction

```
1 for i=1 to I do
2   for j=1 to I do
3       if i=j then SMJAC(i, j)=SMJACA(i, j)=1
4       else
5         for f=1 to F do
6             if Vi(vf)=1 then ni++
7             if Vj(vf)=1 then nj++
8             if Vi(vf)=1 and Vj(vf)=1 then nij++
9         SMJAC(i, j) = nij / (ni + nj - nij)
10        SMJACA(i, j) = nij / ni

11 for u=1 to U do
12   for i=1 to I do
13       if u has not rated i then
14           r̂u,i = r̄i + Σ_{j∈NNi} sim(i,j)(ru,j − r̄j) / Σ_{j∈NNi} |sim(i,j)|
```

Table 2. Notations and their descriptions

| notation | description |
|---|---|
| $V_i$ | feature vector of item i |
| I | set of items |
| F | total number of vector elements |
| U | set of users |
| $r_{u,i}$ | rating value of user u for item i |
| $\bar{r}_i$ | mean rating value of item i |
| $\hat{r}_{u,i}$ | predicted rating of user u for item i |
| $NN_i$ | set of nearest neighbors of item i |
| sim(i, j) | similarity between items i and j |

## 2. Example of Similarity Calculation

표 3은 본 연구 주제와 관련된 유사도 측정 방법들을 적용하기 위한 예시 데이터이다. 항목의 평가 범위가 1~5인 정수값이고 두 항목 i와 j 간의 유사도를 구한다고 할 때, 표 3은 각 항목에 해당 평가치를 부여한 전체 사용자 수를 나타낸다.

Table 3. Number of users who rated each rating for the items

| item \ rating(r) | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| i | 4 | 6 | 6 | 3 | 1 |
| j | 5 | 4 | 8 | 2 | 1 |
| $p_{ir}$ | 0.2 | 0.3 | 0.3 | 0.15 | 0.05 |
| $p_{jr}$ | 0.25 | 0.2 | 0.4 | 0.1 | 0.05 |

Kullback-Leibler divergence(KL)와 Bhattacharyya 계수(BC)는 모두 두 분포 간의 차이를 측정하는데 아래와 같은 산출식을 사용한다.

$$BC(i,j)=\sum_{r=1}^{n}\sqrt{p_{ir}p_{jr}}$$

$$KL(i,j)=\sum_{r=1}^{n}p_{ir}\log\frac{p_{ir}}{p_{jr}}$$

따라서, 표의 데이터에 대한 $BC(i,j)=\sum_{r=1}^{5}\sqrt{p_{ir}p_{jr}}$ =0.98744 이고 $KL(i,j)=\sum_{r=1}^{5}p_{ir}\log\frac{p_{ir}}{p_{jr}}$ =0.07434이다.

## IV. Performance Experiments

### 1. Experimental Background

#### 1.1 Dataset

본 연구에서 제안한 유사도 척도의 성능 평가는 항목의 속성 정보를 보유한 데이터셋을 통하여 가능하다. 본 실험에서는 속성으로서 영화 장르를 선택하고, 많은 기존의 관련 연구에서 널리 활용하여 왔던 MovieLens (https://movielens.org) 데이터셋을 선정하였다. 이 데이터셋은 사용자들의 영화 항목에 대한 평가치와 함께 각 항목이 속한 장르 정보를 유지 관리하므로 각 평가치가 어떠한 장르에 속하는지 알 수 있다. 따라서 제안 방법의 성능 측정을 위하여 적합한 데이터셋이며, 각 항목은 1~6개의 장르에 속하고 전체 장르 수는 18개이다.

MovieLens 100K와 1M은 성능 실험에 널리 활용되어 왔는데[1], 후자는 6040명 사용자의 약 100만 건의 평가 데이터를 포함한다. 그러나 본 연구에서는 이 데이터 전부를 처리하기에는 용량 및 처리 속도 측면에서 제한적인 컴퓨팅 환경임을 감안하여 일부 사용자 데이터를 추출하여 실험을 진행하였다. 구체적으로, 데이터의 분포를 유지하고, 편향을 방지하기 위해 무작위 추출 방식을 적용하였는데, 1부터 6040의 사용자 ID 중에서 무작위로 ID를 2096번 반복 선택하여 선택된 ID의 사용자의 항목 평가 데이터를 모두 추출하였다. 실험에 사용한 두 데이터셋의 특징은 표 4에 제시하였다. MovieLens 1M은 원래 희소성 수준이 약 0.9581인데, 표 4와 같이 본 실험에서는 0.95677로서 큰 차이가 없으므로 일부 데이터를 추출 및 사용한 실험 시도는 유효할 것으로 판단된다. 또한 두 데이터셋의 희소성 수준의 차이로 인해, 실험 방법들의 성능 점검은 다양한 환경에서 더욱 세밀하게 이루어질 수 있다.

Table 4. Characteristics of datasets used for experiments

| | MovieLens 100K (ML-1) | MovieLens 1M (ML-2) |
|---|---|---|
| No. of users | 943 | 2096 |
| No. of items | 1682 | 3952 |
| total # of ratings | 100,000 | 358,158 |
| Rating range | 1~5, integer | 1~5, integer |
| Sparsity level | 0.93695 | 0.95677 |
| Min/max/avg #ratings per user | 20/737/106.05 | 20/1850/170.88 |
| Min/max/avg #ratings per item | 1/583/59.45 | 1/1291/90.63 |

#### 1.2 Performance Metrics

시스템이 산출한 예측 평가치가 얼마나 실제 평가치에 근접하는지를 측정하는 것은 추천 시스템의 성능 평가에 있어서 매우 중요하다. 본 실험에서는 이를 대표하는 척도로서 널리 활용되는 평균 절대 오차(Mean Absolute Error, MAE)를 도입하였는데, 테스트 데이터에 대하여 시스템이 산출한 예측 평가치와 실제치의 오차 평균을 의미한다. 테스트 데이터는 전체 데이터 중 20%로 설정하였고, 나머지 80%의 훈련용 데이터를 활용하여 항목 간의 유사도를 산출한 후에 성능 점검의 목적으로 활용하였다.

시스템은 사용자가 미평가한 항목에 대한 예측 평가치를 산출하고 이 값이 높은 항목 순위대로 추천 리스트를 제공한다. 따라서 사용자가 관심을 더 많이 가질만한 항목이 상위에 놓이는 것이 바람직하다. 이를 위해 성능 평가에 순서 개념을 도입한 것이 Mean Average Precision(MAP)이다[1]. Precision(정밀도)은 시스템이 추천한 모든 항목들 중에서 사용자가 실제로 선호하는 항목들의 비율을 의미한다. 각 사용자의 Average Precision(AP)은 다음과 같이 구하는데, rel(i)는 순위 i의 추천 항목을 사용자가 선호하면 1, 그렇지 않으면 0의 값을 가지며, m은 사용자의 총 선호 항목 개수이고, K는 시스템의 추천 항목 수를 나타낸다. MAP@K는 K개의 추천 항목 수에 대하여 각 사용자의 AP@K의 평균으로 산출한다.

$$AP@K=\frac{1}{m}\sum_{i=1}^{K}precision@i\cdot rel(i)$$

$$MAP@K=\frac{1}{|U|}\sum_{u\in U}AP_u@K$$

또다른 성능 평가 척도로서 커버리지(coverage)를 도입하였다. 이 척도는 시스템이 얼마나 다양한 항목들을 추천할 수 있는지를 측정하는 지표로서 커버리지가 클수록 다양한 항목이 추천되므로 사용자가 새로운 항목을 접할 수 있게 된다. 구체적으로 커버리지는 사용자가 미평가한 전체 항목들 중에서 시스템이 평가치를 부여한 항목들의 비

율로서 구한다.

### 1.3 Baseline Methods

성능 비교 대상으로서 평가 빈도 및 확률 분포에 근거한 기존의 대표적인 유사도 척도들과 평가치 기반의 전통적인 피어슨 상관의 유사도 척도를 포함하였다. 각 방법들의 범례 표기와 의미는 다음과 같다.

- COR: 피어슨 상관도
- JAC: Jaccard 계수
- KL: Kullback-Leibler Divergence
- BC: Bhattacharyya 계수
- SMJAC: 제안 방법인 항목 기반의 의미적 Jaccard 계수
- SMJACA: 제안 방법인 비대칭의 항목 기반의 의미적 Jaccard 계수

## 2. Results of Experiments

그림 1은 두 데이터셋을 활용하여 평균절대오차를 측정한 결과이다. 모든 방법들에 대하여 인접 이웃수가 증가함에 따라서 오차가 점차 감소하여 안정화됨을 알 수 있다. 다만 JAC의 경우에는 예외적으로 ML-1에서 인접 이웃수의 증가에 따라 오히려 약간의 성능 저하를 일으켰다. 두 데이터셋 공통적으로 COR는 가장 저조한 성능을, BC는 가장 우수한 성능을 보였는데, 실험 방법들 중에서 유일하게 평가치를 활용하는 COR는 희소한 데이터 환경에서 유사도 산출이 부정확할 수 있으므로 이와 같은 결과가 나타난 것으로 판단된다. 두 확률 분포의 동질성 여부를 파악하는 척도인 KL과 BC의 성능 격차는 ML-1 보다 ML-2에서 더욱 두드러졌는데, 이로써 BC는 데이터 희소성의 영향에 더욱 강함을 보였는데, 이는 [19]에서 KL이 확률분포의 연속성을 필요로 한다는 단점을 기술하고 대신에 BC를 활용한 것과 관련된 결과로 판단된다.

제안 방법의 성능은 그림 1에서 보듯이 ML-2에서 BC 다음으로 우수하였는데 이는 표 3에 제시한 사용자 및 항목당 평가 개수가 ML-1 보다 ML-2가 더욱 많으므로, 비록 후자 데이터셋의 희소성 수준이 높다고 할지라도 사용자의 미평가 항목에 대한 인접 이웃의 평가치를 구하기가 더 용이하기 때문인 것으로 판단된다. SMJAC과 SMJACA의 성능 차이는 미미하므로 비대칭 유사도 산출의 필요성은 적어도 MAE 관점에서는 낮은 것으로 파악된다.

그림 2는 추천 항목 수에 따른 MAP 성능 변화 결과이다. 두 데이터셋 모두에서 대체로 JAC의 성능이 가장 저조한 것을 확인할 수 있는데, 공통 평가 개수에 의거한 추천 결과이므로 이에 대하여는 사용자의 선호 순위가 높지 않은 것으로 판단된다. COR의 경우에는 ML-2에서 더욱 성능이 낮았는데 이는 희소 환경에서 COR 유사도 산출의 부정확성 때문일 것으로 판단된다. 항목에 대한 평가치 분포를 근거로 한 KL과 BC는 대등한 성능 결과를 보여 MAE 결과와는 다른 양상을 보였다. ML-1에서 SMJAC과 SMJACA는 대체로 가장 우수한 성능을 보인 방법들 중에 속하는 것으로 나타났고, ML-2에서는 다른 방법들보다 월등히 우수한 성능을 보였다. ML-2의 데이터 희소성이 높음에도 불구하고 항목 당 평가수가 많으므로 이러한 결과의 원인으로 작동한 것으로 보이며, 사용자가 선호하는 장르만을 기준으로 추천하여도 높은 만족도를 보임을 알 수 있다.

그림 3은 실험 방법들의 커버리지 결과이다. COR의 성능은 다른 성능 지표 결과에서와 마찬가지로 희소 수준의 영향을 매우 크게 받는 것으로 확인되었다. 특히 ML-2에서 JAC 보다도 낮은 가장 저조한 결과를 보였다. 이는 다른 성능 결과에서와 마찬가지로 희소 데이터를 이용할 때 COR 유사도의 정확성이 저하되기 때문이다. JAC 역시 희소 데이터 환경에서 유리한 유사도 척도라는 기존 연구 결과와는 달리 커버리지 측면에서는 매우 낮은 성능 결과를 나타냈는데 이는 항목 집합이 커지면 공통 평가 항목의 개수가 상대적으로 더 적어지기 때문인 것으로 판단된다. 확률 분포 기반의 KL과 BC는 이들보다는 우수하였으며 특히 BC는 MAE 결과에서와 마찬가지로 KL을 넘어서는 결과를 나타냄으로써 [19]의 주장을 뒷받침하였다. ML-1에서 BC와 KL과 마찬가지로 제안 방법들도 대등하게 우수한 성능 결과를 나타냈는데, 특히 SMJAC은 인접 이웃수가 증가함에 따라 그 성능이 가장 우수하였다. 주목할만한 점은 ML-2에서 제안 방법들의 성능이 월등히 우수하였는데, 특히 SMJACA는 다른 모든 방법들을 크게 능가하는 성능을 보였다. 이는 표 3에 제시한 바와 같이 사용자 및 항목 당 평가개수가 ML-1 보다 크므로 비대칭적 유사도 산출의 효과가 작용하였기 때문으로 판단된다.

## 3. Comparison of Performance Complexity

본 절에서는 메모리와 시간 복잡도 측면에서 실험 방법들의 성능을 살펴 보고자 한다. 모든 방법들의 평가치 예측은 동일한 절차를 따르므로, 유사도 산출을 위한 성능만을 비교한다. 표 5에서와 같이 COR, JAC, KL, BC는 사용자와 항목의 평가 매트릭스가 기준이므로 그 크기만큼의 메모리가 필요하다. 반면에 제안 방법은 실험에서 각 항목의 장르 속성값을 필요로 하므로 G를 장르 집합이라고 할 때 표에 제시한 바와 같은 메모리가 사용된다.

유사도 산출을 위한 시간 성능에 있어서 COR와 JAC은 두 사용자 간의 모든 항목 평가치를 참조해야 하며, KL과 BC는 각 항목의 사용자 평가치 분포를 계산하는데 필요한 O(|U|×|I|) 시간과 모든 두 항목 간의 유사도 산출을 위한 O(|I|×|I|×n)의 시간이 소요된다(n은 부여 가능한 평가치 종류 수). 제안 방법의 성능은 항목 기반이므로 표 1의 알고리즘을 토대로 하여 표 5에 제시한 바와 같이 산출된다.

Table 5. Memory and time complexity of the methods

| | COR | JAC | KL | BC | SMJAC | SMJACA |
|---|---|---|---|---|---|---|
| memory | O(\|U\|×\|I\|) | | | | O(\|G\|×\|I\|) | |
| time | O(\|U\|×\|U\|×\|I\|) | | O(\|U\|×\|I\|+\|I\|×\|I\|×n) | | O(\|I\|×\|I\|×F) | |

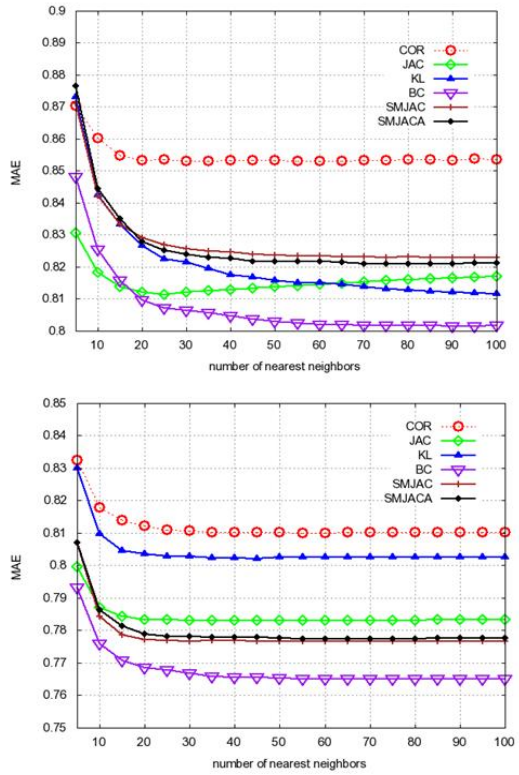


Fig. 1. MAE with varying number of nearest neighbors (up: ML-1, down: ML-2)

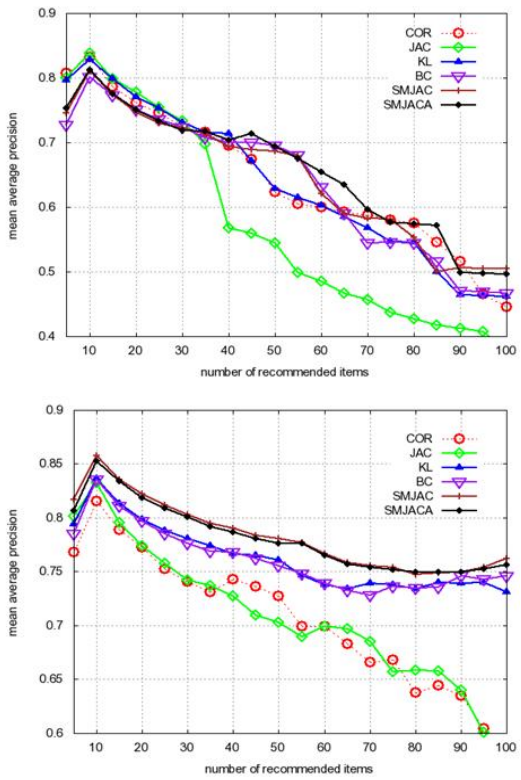


Fig. 2. MAP with varying number of recommended items (up: ML-1, down: ML-2)

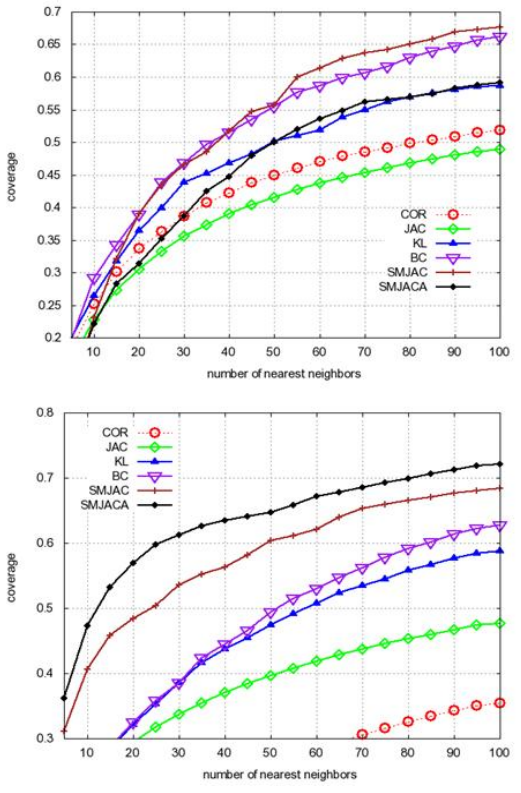


Fig. 3. Coverage with varying number of nearest neighbors (up: ML-1, down: ML-2)

## V. Conclusions

이웃 기반의 협력 필터링 시스템에서는 현 사용자를 위한 추천 항목들을 생성하기 위하여, 유사한 사용자 또는 항목들에 의존하므로 유사도 계산의 정확도를 저하시키는 데이터 희소성 문제는 시스템 성능에 크게 영향을 미친다. 이러한 문제의 해결을 위하여 다양한 연구 결과가 발표되었으나, 본 연구에서는 계산량이 크거나 복잡한 학습 구조에 의지하지 않으면서도 효율적인 항목 기반의 유사도 척도를 제안한다. 이와 같은 성격을 가진 기존의 대표적인 척도인 Jaccard 계수나 Bhattacharyya 계수와는 달리, 본 제안 방법은 항목의 속성 정보를 활용하는 의미적 유사도에 속하므로 데이터 희소성에 견디는 힘이 더욱 강하다. 서로 다른 희소성 수준의 데이터셋을 이용한 성능 실험 결과, 제안 방법은 평균 정밀도와 커버리지 측면에서 우수하였고 특히 더욱 희소하면서 항목별 평가개수가 많은 환경에서 기존 방법들의 성능을 크게 능가하였다. 단, 평균절대오차 측면에서는 Bhattacharyya 계수가 가장 우수한 성능을 보였는데, 실제적으로 예측치의 정확도 보다는 사용자의 만족도가 우선시되는 실제 환경을 고려할 때 제안 방법의 실효성이 더욱 높다고 볼 수 있다.

제안 방법은 기존의 많은 유사도 척도들과는 달리 사용자의 공통 항목 평가치나 평가 개수에 의존하지 않으므로 데이터 희소성 문제 해결에 더욱 큰 도움이 되며, 사용자의 직접적인 평가치를 제공하지 않는 시스템에서도 적용 가능하다. 또한 항목의 속성 및 속성값이 유동적이지 않는 경우에 모든 항목 간 유사도 산출은 단 1회만 실행하면 되므로, 사용자들의 추가 유입으로 인한 시스템 확장성 문제에도 도움이 된다. 본 실험에서는 영화 장르를 속성으로 선택하였으나, 다른 속성에 대해서도 적용 가능하므로 유연성이 높다는 잇점이 있다.

## REFERENCES

[1] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, “Evaluating Collaborative Filtering Recommender Algorithms: A Survey,” IEEE Access, Vol. 6, pp. 74003-74024, 2018. DOI: 10.1109/ACCESS.2018.2883742

[2] F. Fkih, "Similarity measures for Collaborative Filtering-based Recommender Systems: Review and Experimental Comparison," Journal of King Saud University - Computer and Information Sciences, Vol. 34, No. 9, pp. 7645-7669, 2022. DOI: 10.1016/j.jksuci.2021.09.014

[3] A. Althbiti, R. Alshamrani, T. Alghamdi, S. Lee, and X. Ma, "Addressing Data Sparsity in Collaborative Filtering Based Recommender Systems Using Clustering and Artificial Neural Network," IEEE 11th Annual Computing and Communication Workshop and Conference, pp. 218-227, 2021. DOI: 10.1109/CCWC51732.2021.9376008

[4] A. Roko, A. Almu, A. Mohammed, and I. Saidu, “An Enhanced Data Sparsity Reduction Method for Effective Collaborative Filtering Recommendations,” International Journal of Education and Management Engineering, Vol. 10, No. 1, pp. 27-42, 2020. DOI: 10.5815/ijeme.2020.01.04.

[5] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A.H. Gandomi, “Resolving Data Sparsity and Cold Start Problem in Collaborative Filtering Recommender System using Linked Open Data,” Expert Systems with Applications, Vol. 149, 2020. DOI: 10.1016/j.eswa.2020.113248

[6] M. Mataoui, H. Bahloul, and A. Ziane, “Leveraging Knowledge Graphs for Paper Recommendation Systems,” Intelligent Systems and Pattern Recognition, Communications in Computer and Information Science, Vol. 2303, 2025. DOI: 10.1007/978-3-031-82150-9_18

[7] Z. Shokrzadeh, M.-R. Feizi-Derakhshi, M.-A. Balafar, and J.B. Mohasefi, “Knowledge Graph-based Recommendation System Enhanced by Neural Collaborative Filtering and Knowledge Graph Embedding,” Ain Shams Engineering Journal, Vol. 5, No. 1, 2024. DOI: 10.1016/j.asej.2023.102263

[8] L. Xia, C. Huang, Y. Xu, J. Zhao, D. Yin, and J. X. Huang, “Hypergraph Contrastive Collaborative Filtering,” The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 70-79. 2022. DOI: 10.1145/3477495.3532058

[9] J. Latrech, Z. Kodia, and N.B. Azzouna, “CoD-MaF: Toward a Context-Driven Collaborative Filtering using Contextual Biased Matrix Factorization,” International Journal of Data Science and Analytics, 2025. DOI: 10.1007/s41060-025-00747-6

[10] A. Torkashvand, S.M. Jameii, and A. Reza, “Deep Learning-based Collaborative Filtering Recommender Systems: A Comprehensive and Systematic Review,” Neural Computing and Applications, Vol. 35, pp. 1-25, 2023. DOI: 10.1007/s00521-023-08958-3

[11] H. Xia, Y. Luo, and Y. Liu, “Attention Neural Collaboration Filtering based on GRU for Recommender Systems,” Complex & Intelligent Systems, Vol. 7, pp. 1367-1379, 2021. DOI: 10.1007/s40747-021-00274-4

[12] S. Lee, “Fuzzy Clustering with Optimization for Collaborative Filtering-based Recommender Systems,” Journal of Ambient Intelligence and Humanized Computing, Vol. 13, No. 9, pp. 1-18, 2022. DOI: 10.1007/s12652-021-03552-8

[13] S. Bag, S.K. Kumar, and M.K. Tiwari, “An Efficient Recommendation Generation using Relevant Jaccard

Similarity," Information Sciences, Vol. 483, pp. 53-64, 2019. DOI: 10.1016/j.ins.2019.01.023

[14] M. Ayub, M.A. Ghazanfar, T. Khan, and A. Saleem, "An Effective Model for Jaccard Coefficient to Increase the Performance of Collaborative Filtering," Arabian Journal for Science and Engineering, Vol. 45, pp. 9997-10017, 2020. DOI: 10.1007/s13369-020-04568-6

[15] R. Pan, C. Ge, L. Zhang, W. Zhao, and X. Shao, "A New Similarity Model based on Collaborative Filtering for New User Cold Start Recommendation," IEICE Transsactions on Information and Systems, Vol. 6, pp. 1388-1394, 2020. DOI: 10.1587/transinf.2019EDP7258

[16] H.I. Abdalla, Y.A. Amer, L. Nguyen, et al., "Numerical Similarity Measures Versus Jaccard for Collaborative Filtering," International Conference on Advanced Intelligent Systems and Informatics, 2023. DOI: 10.1007/978-3-031-43247-7_20

[17] D. Wang, Y. Yih, and M. Ventresca, "Improving Neighbor-based Collaborative Filtering by Using a Hybrid Similarity Measurement," Expert Systems with Applications, Vol. 160, 2020. DOI: 10.1016/j.eswa.2020.113651

[18] J. Deng, Y. Wang, J. Guo, Y. Deng, J. Gao, and Y. Park, "A Similarity Measure based on Kullback–Leibler Divergence for Collaborative Filtering in Sparse Data," Journal of Information Science. Vol. 45, 2018. DOI: 10.1177/0165551518808188

[19] A. Jain, P.K. Singh, and J. Dhar, "Multi-objective Item Evaluation for Diverse as well as Novel Item Recommendations," Expert Systems with Applications, Vol. 139, 2020. DOI: 10.1016/j.eswa.2019.112857.

[20] R.K. Singh, P.K. Singh, J.P. Singh, A.K. Singh, and S. Dhanasekaran, "Utilizing Alike Neighbor Influenced Similarity Metric for Efficient Prediction in Collaborative Filter-Approach-Based Recommendation System," Applied Sciences, Vol. 12, No. 22, 2022. DOI: 10.3390/app122211686

[21] M. Al-Hassan, B. Abu-Salih, E. Alshdaifat, et al. "An Improved Fusion-Based Semantic Similarity Measure for Effective Collaborative Filtering Recommendations," International Journal of Computational Intelligence Systems, Vol. 17, No. 45, 2024. DOI: 10.1007/s44196-024-00429-4

## Authors

Soojung Lee received the B.S. degree in Mathematics Education from Ewha Woman's University, Korea in 1985. She received M.S. and Ph.D. degrees in Computer Science from Texas A&M University in 1990 and 1994, respectively. Dr. Lee joined the faculty of the Department of Computer Education at Gyeongin National University of Education, Gyunggi-do, Korea, in 1998, as a professor. She is interested in recommender systems, information filtering, data mining techniques, and computer education.