Collaborative_Filtering

HA SEUNG HYUN

Collaborative Filtering

넷플릭스

Collaborative Filtering

넷플릭스

Collaborative Filtering

넷플릭스

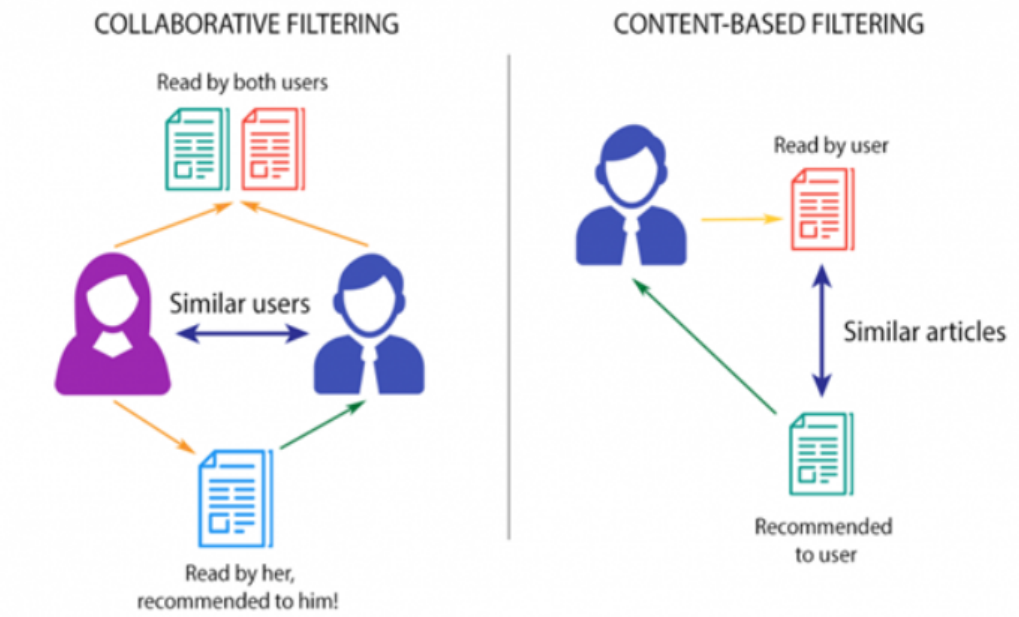
사용자 취향에 맞는 영화 포스터 제공

Collaborative Filtering

[테크월드=김지윤 기자] 넷플릭스를 이용하는 많은 사람들은 끊임없이 추천되는 자신의 취향을 저격하는 영상들에 몇 시간 넘게 푹 빠져버린 경험이 있을 것이다. 넷플릭스는 어떻게 각각의 취향을 파악해 콘텐츠를 추천해줄 수 있을까?

그것을 바로 알고리즘 덕분!

독일의 과학 저널리스트 크리스토프 드뢰서는 책 <알고리즘이 당신에게 이것을 추천합니다>을 통해 넷플렉스의 추천 서비스가 모두 '알고리즘' 덕분이라고 말한다. 그는 오늘날 대부분의 추천 시스템이 '협업 필터링(collaborative filtering)' 알고리즘과 '내용 기반 필터링(content-based filtering)' 알고리즘을 조합한 형태라고 한다.



출처: NETHRU

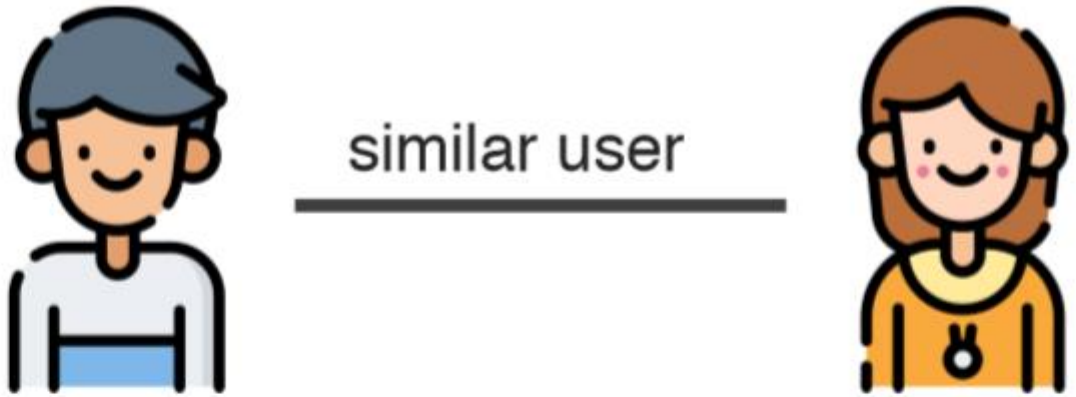
Collaborative Filtering

협업 필터링은 다른 상품(item)이나 사용자(user) 정보를 이용하여 평점(rating)이 없는 데이터의 평점을 예측한다.
예측 평점이 높은 상위 아이템을 추천함으로써 작동하는 대표적인 추천 시스템 알고리즘이다.

계란
빵
우유
커피
치즈

빵
커피
계란
치즈
딸기잼

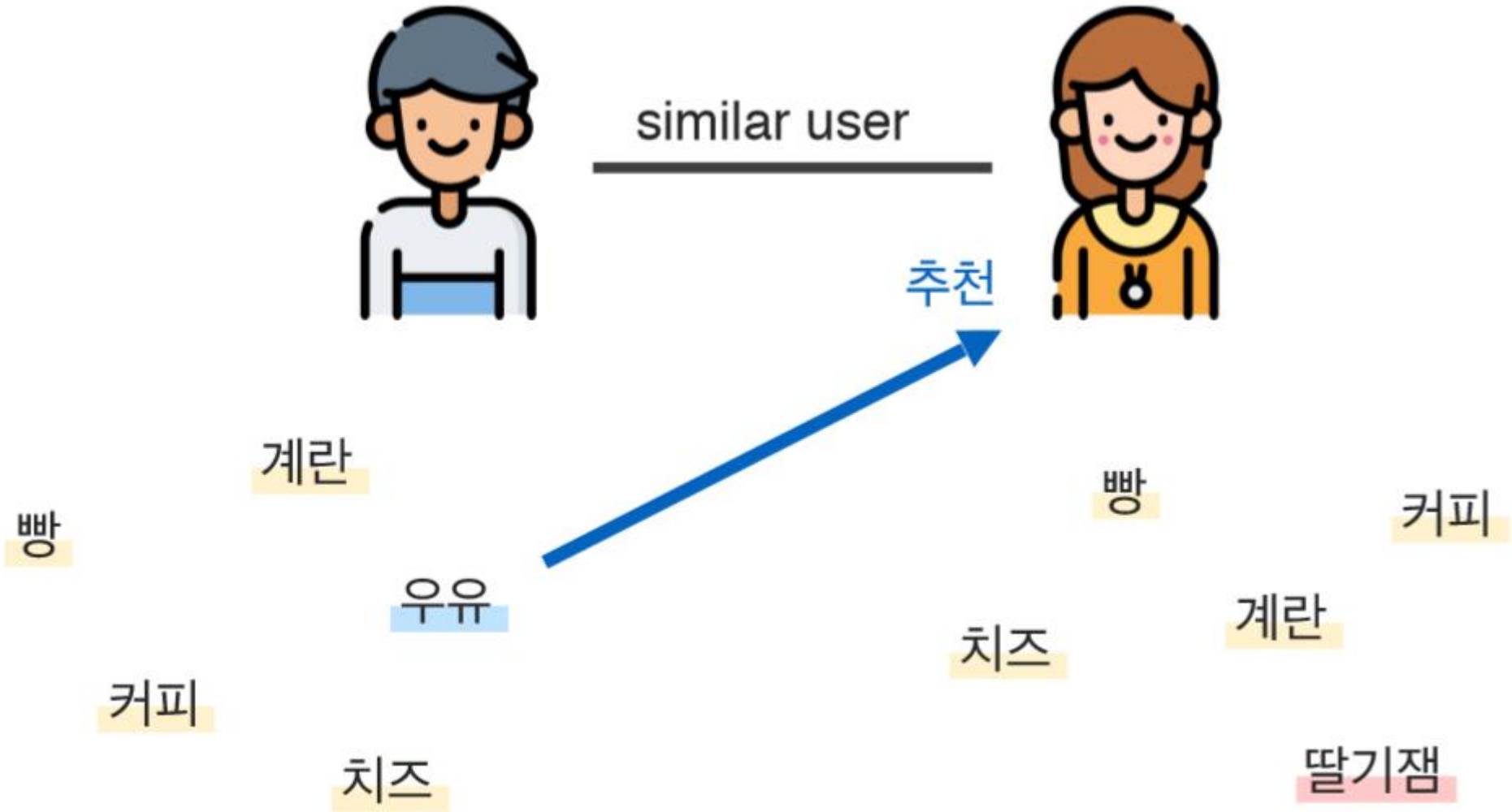
Collaborative Filtering



계란
빵
우유
커피
치즈

빵
커피
계란
치즈
딸기잼

Collaborative Filtering



Collaborative Filtering

추천시스템(Recommender System)에서 널리 사용되는 협업 필터링(이하 Collaborative Filtering)의 원리를 알아보고 이를 구현해 보도록 합니다. 추천 시스템은 사용자(이하 사용자)가 특정 물건이나 서비스(이하 상품)에 대한 선호 여부나 선호도를 예측하는 시스템을 의미합니다. 추천 시스템은 아마존과 같은 이커머스부터 페이스북과 같은 SNS, 유튜브, 넷플릭스 등과 같은 동영상 플랫폼까지 다양한 분야에서 두루 활용되고 있습니다. 대기업 한정 데이터 확보 가능

Collaborative Filtering에는 사용자에게 상품을 추천 받는 방법이 크게 두 가지가 있습니다.

1. 사용자가 선호하는 상품과 유사한 다른 상품 을 추천(상품 기반)하거나 기업에서는 item 기반을 많이 사용
2. 사용자와 유사한 다른 사용자가 선호하는 상품을 추천(사용자 기반)합니다.

사용자 기반 기법이 먼저 등장한 전통적인 알고리즘이고 상품 기반 방식은 이후 아마존(Amazon)이 제안한 기법입니다. 상품 기반 기법이 더 많은 기업들에서 사용되고 있다고 합니다.

엄청난 대기업이라 사용자 기반이 가능

사용자 기반 방식이 갖는 문제는 우선 **1. 계산 복잡성 문제**와 **2. 희소성 문제**가 대표적입니다. 아마존과 같이 거대 이커머스 회사들은 수백만 명의 사용자와 수백만 개의 상품을 관리해야 하는데 사용자 기반 방식을 사용하는 경우 사용자가 추가될 때마다 나머지 모든 사용자와의 유사도를 연산해야 한다는 문제점이 있습니다. 상품 기반 방식을 사용하는 경우에 미리 구해 놓은 상품 간 유사도를 활용할 수 있기 때문에 이러한 문제점이 어느 정도 해결됩니다! 물론 상품 기반 방식도 상품과 사용자가 계속 추가되므로 일정 기간마다 새롭게 유사도를 구해야 하지만 사용자 기반 방식보다는 훨씬 계산 복잡성이 작습니다. 그리고 계산 복잡성 문제가 해결되는 대신 이 거대한 행렬을 저장할 공간이 따로 확보되어야 한다는 점을 굳이 단점으로 뽑을 수 있습니다. 데이터 희소성 문제는 협업 필터링 알고리즘의 본질적인 취약한 점이지만 사용자가 많은 상품을 평가한 경우는 보통 없어서 이런 경우 사용자간의 유사도를 연산하는 것 자체가 어렵기 때문에 보통 사용자 기반 방식이 더 취약합니다.

Collaborative Filtering

예를 들어 이커머스 서비스 추천 시스템을 구현해본다고 가정했을 때, 크게 두 가지 정보를 활용해볼 수 있습니다.

1. 우선 사용자가 상품을 구매한 이후 남긴 평점 정보를 활용할 수 있습니다.
2. 혹은 사용자가 상품 판매 페이지에 머무른 시간 혹은 해당 상품을 클릭했는지 등의 정보를 활용해볼 수도 있습니다.

앞서 말한 경우와 같이 사용자가 상품에 내린 직접적인 평가 데이터를 **명시적 정보 (이하 explicit ratings),** 사용자 행동을 통해 추론한 상품에 대한 간접적인 평가 데이터를 **암시적 정보 (이하 implicit ratings)**라고 말합니다.

explicit ratings는 사용자로부터 얻을 수 있는 가장 정확한 평점입니다.
하지만, 사용자가 평가를 내릴 때 충분한 시간을 할애하는 것은 아니기에 평점 간의 척도가 정확하지 않을 수 있고 평점 수가 충분하지 않다는 한계가 있습니다.

이에 비해 **implicit ratings**는 평점을 쉽게 많이 수집할 수 있다는 장점이 있지만, 해당 정보를 무조건 사용자가 상품에 내린 긍정적 평가라고 결론 내릴 수는 없다는 단점이 있습니다.

Collaborative Filtering

상품 / 사용자 기반 기법은 전반적으로 다음과 같은 흐름으로 동작합니다.

1. 우선 사용자 uu가 내릴 상품 ii에 대한 평점(rating)을 추정하고자 합니다. 상품 ii / 사용자 uu와 나머지 모든 상품 / 사용자의 유사도를 연산합니다.

2.유사도가 높은 k개 상품 / 사용자를 선택합니다. 이를 이웃이라고 부르겠습니다.

3.상품 기반 혹은 사용자 기반 기법에 따라 아래 단계를 수행하며 평점을 예측합니다.
- 상품 기반 : 이웃 상품에 내린 사용자 uu의 평점(rating)을 상품 ii와의 유사도에 따라 가중 평균을 구합니다.
- 사용자 기반 : 이웃 사용자가 상품 ii에 내린 평점(rating)을 사용자 uu와의 유사도에 따라 가중 평균을 구합니다.

4.아직 평점(rating)이 없는 항목에 대해 모든 평점(rating)을 예측합니다. 평점(rating) 예측 값 상위 n개 상품을 추천합니다.

이러한 알고리즘을 잘 이해하기 위해서 파이썬 프로그래밍 언어로 직접 구현해서 동작 원리를 다시 익혀 볼 수 있도록 합니다.

Collaborative Filtering

사용자 별로 상품에 대한 평점 정보를 담고 있는 행렬 형태의 데이터를 활용한다.

item I i, j 등으로 표기

user U
u, v 등으로 표기

	노인과 바다	백설공주	신데렐라	어린 왕자	콩쥐팥쥐	흥부전
민지		5	4	1	5	3
현우	3		2	3	1	2
민수	3	4	4	3	4	
지민	4	1	1	5	2	3
지연	5		3	4	3	3

rating r_{ui}
사용자 u가 상품 i에 남긴 평점

U_{ij} : 상품 i와 상품 j에 대한 평점 정보가 모두 담겨져 있는 사용자 집합

I_{uv} : 사용자 u와 사용자 v가 평점을 남긴 모든 정보가 있는 상품 집합

Collaborative Filtering

크게 user가 선호하는 item과 유사한 다른 item을 추천하는 item-based 기법과 user와 유사한 다른 user가 선호하는 item을 추천하는 user-based 기법이 있다.

$\hat{r}_{민지,\ 노인과\ 바다}$

	노인과 바다	백설공주	신데렐라	어린 왕자	콩쥐팥쥐	흥부전
민지	?	5	4	1	5	3
현우	3		2	3	1	2
민수	3	4	4	3	4	
지민	4	1	1	5	2	3
지연	5		3	4	3	3

Collaborative Filtering

User Based Filtering

user-based 예시

$\hat{r}_{민지,\ 노인과\ 바다}$

	노인과 바다	백설공주	신데렐라	어린 왕자	콩쥐팥쥐	흥부전	
민지		5	4	1	5	3	
현우	3		2	3	1	2	0.7261
민수	3	4	4	3	4		0.9547
지민	4	1	1	5	2	3	0.5985
지연	5		3	4	3	3	0.8541

"민지"를 제외한 나머지 사용자(neighbor user)가 남긴 상품(노인과 바다)에 대한 평점을 유사도에 따라 가중 평균(weighted mean)을 구한다.

$$\hat{r}_{민지,\ 노인과\ 바다} = \frac{0.7261 \times 3 + 0.9547 \times 3 + 0.5985 \times 4 + 0.8541 \times 5}{0.7261 + 0.9547 + 0.5985 + 0.8541} = 3.74$$

Collaborative Filtering

Item Based Filtering

item-based 기법도 이와 유사하게
유사도를 사용자(user)가 아닌 상품(item) 별로 계산하여 예측해 볼 수 있다.

item-based 예시

$\hat{r}_{\text{민지, 노인과 바다}}$

$$\hat{r}_{\text{민지, 노인과 바다}} = \frac{0.7761 \times 5 + 0.8784 \times 4 + 0.9830 \times 1 + 0.9032 \times 5 + 0.9949 \times 3}{0.7761 + 0.8784 + 0.9830 + 0.9032 + 0.9949} = 3.50$$

	노인과 바다	백설공주	신데렐라	어린 왕자	콩쥐팥쥐	흥부전
민지		5	4	1	5	3
현우	3		2	3	1	2
민수	3	4	4	3	4	
지민	4	1	1	5	2	3
지연	5		3	4	3	3
		0.7761	0.8784	0.9830	0.9032	0.9949

Collaborative Filtering

가령 "민지"라는 사용자가 '노인과 바다'라는 책을 읽었을 경우, 몇 점의 평점을 남길 것인지 알고 싶다면?
1) 민지와 유사한 성향을 가진 사용자를 찾은 뒤
2) 이 사용자들이 남긴 '노인과 바다'의 평점의 평균(가중평균)을 구하면 된다.

	노인과 바다	백설공주	신데렐라	어린 왕자	콩쥐팥쥐	흥부전	
민지		5	4	1	5	3	
현우	3		2	3	1	2	0.7261
민수	3	4	4	3	4		0.9547
지민	4	1	1	5	2	3	0.5985
지연	5		3	4	3	3	0.8541

Collaborative Filtering

가령 “민지”라는 사용자가 “노인과 바다”라는 책을 읽었을 경우, 평점을 몇 점을 남길 것인지 알고 싶다면?

1) “민지”와 유사한 성향을 가진 사용자를 찾은 뒤, 2) 이 사용자들이 남긴 노인과 바다의 평점의 평균(가중 평균)을 구하면 된다

$\hat{r}_{\text{민지, 노인과 바다}}$

	노인과 바다	백설공주	신데렐라	어린 왕자	콩쥐팥쥐	흥부전	
민지		5	4	1	5	3	
현우	3		2	3	1	2	0.7261
민수	3	4	4	3	4		0.9547
지민	4	1	1	5	2	3	0.5985
지연	5		3	4	3	3	0.8541

01 사용자 “민지”와 나머지 사용자간의 유사도를 계산한다.
(유사도를 계산하는 방법은 차후 설명)

Collaborative Filtering

가령 "민지"라는 사용자가 "노인과 바다"라는 책을 읽었을 경우, 평점을 몇 점을 남길 것인지 알고 싶다면?

1) "민지"와 유사한 성향을 가진 사용자를 찾은 뒤, 2) 이 사용자들이 남긴 노인과 바다의 평점의 평균(가중 평균)을 구하면 된다

$\hat{r}_{\text{민지, 노인과 바다}}$

	노인과 바다	백설공주	신데렐라	어린 왕자	콩쥐팥쥐	흥부전	
민지		5	4	1	5	3	
현우	3		2	3	1	2	0.7261
민수	3	4	4	3	4		0.9547
지민	4	1	1	5	2	3	0.5985
지연	5		3	4	3	3	0.8541

02 "민지"를 제외한 나머지 사용자(neighbor user)가 남긴 상품(노인과 바다)에 대한 평점을 유사도에 따라 가중 평균(weighted mean)을 구한다.

$$\hat{r}_{\text{민지, 노인과 바다}} = \frac{0.7261 \times 3 + 0.9547 \times 3 + 0.5985 \times 4 + 0.8541 \times 5}{0.7261 + 0.9547 + 0.5985 + 0.8541} = 3.74$$

유사도(similarity) 구하기

유사도(similarity)를 구하는 방법은 여러가지가 있으며, 각 방법마다의 장단점이 있다
여기서는 가장 일반적으로 사용하는 코사인 유사도(cosine similarity)를 사용할 것

1. 유클리디안 거리 (Euclidean Distance)

$$\sqrt{\sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2}$$

2. 코사인 유사도 (Cosine Similarity)

$$\frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}}$$

3. 피어슨 상관계수 (Pearson Correlation Coefficient)

$$\frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u) \bullet (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

Collaborative Filtering

Q) 유사도(similarity)를 어떻게 구하지?

유사도(similarity)를 구하는 방법은 여러가지가 있으며, 각 방법마다의 장단점이 있다
여기서는 가장 일반적으로 사용하는 코사인 유사도(cosine similarity)를 사용할 것

2. 코사인 유사도 (Cosine Similarity)

$$\frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}}$$

Cosine Similarity

코사인 유사도는 두 벡터의 코사인 각 크기를 통하여 두 벡터의 유사한 정도를 의미한다.
벡터의 크기와 무관하게 두 벡터가 가리키는 방향이 얼마나 유사한지 파악하기 위해 사용한다.

Example
사용자1(이하 철수)와 사용자2(이하 영희)가 있을 때

$$\frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}}$$

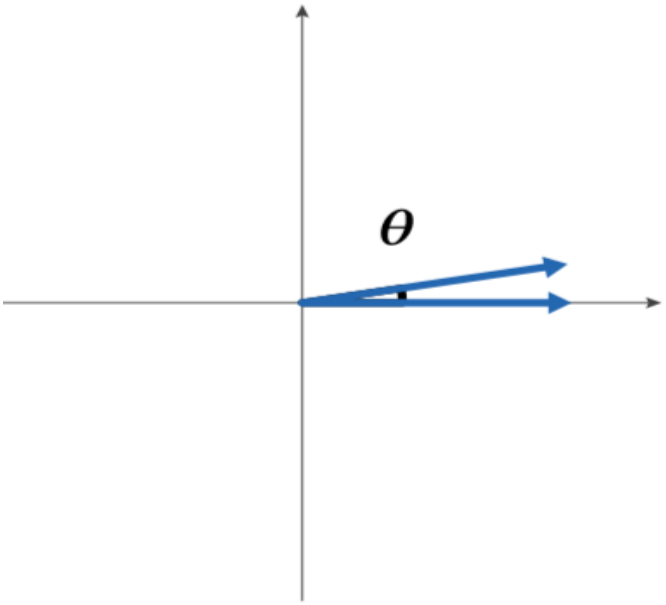
철수는
- 노인과바다에 3점을
- 신데렐라에 4점을
- 흥부전에 3점을

영희는
- 노인과바다에 3점을
- 신데렐라에 2점을
- 흥부전에 4점을 줬다면, 두 사용자의 코사인 유사도는

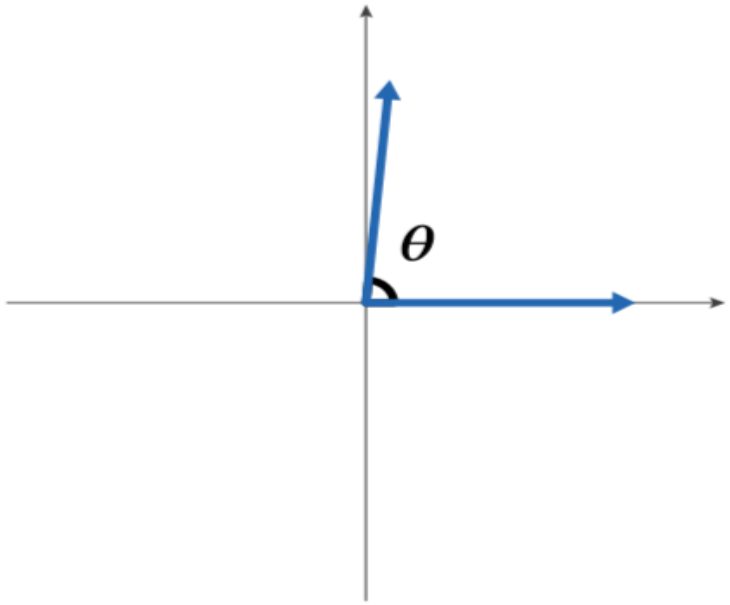
$$\frac{(3x3 + 4x2 + 3x4)}{\sqrt{(3x3 + 4x4 + 3x3)^2 (3x3 + 2x2 + 4x4)^2}} = 0.92$$

Cosine Similarity

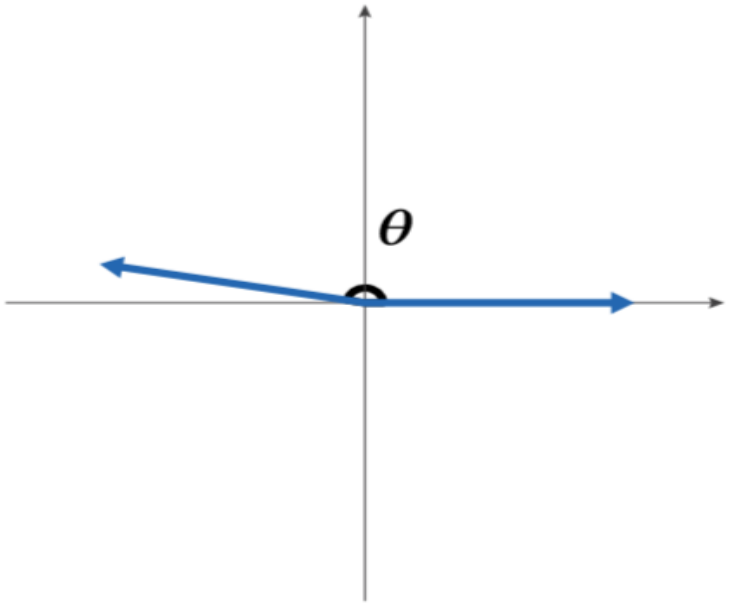
두 벡터의 방향이 완전히 같다면 +1, 완전히 다르다면 -1에 가까운 값을 갖는다.
두 벡터가 아무런 관련이 없으면, 즉 독립이면 0의 값을 갖는다.



$cos\theta \approx 1$



$cos\theta \approx 0$



$cos\theta \approx -1$

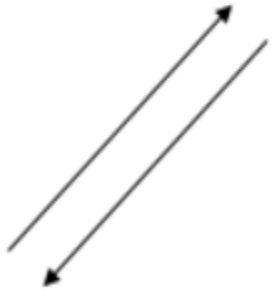
두 사용자가 유사하다

Accuracy를 구하는 것이 아니라서 이 방식이 가능

두 사용자가 유사하지 않다

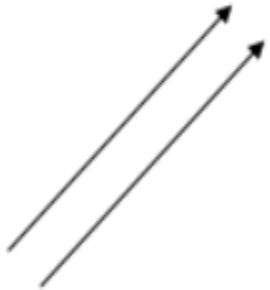
Cosine Similarity

코사인 유사도는 두 벡터 간의 코사인 각도를 이용하여 구할 수 있는 두 벡터의 유사도를 의미합니다. 두 벡터의 방향이 완전히 동일한 경우에는 1의 값을 가지며, 90°의 각을 이루면 0, 180°로 반대의 방향을 가지면 -1의 값을 갖게 됩니다. 즉, 결국 코사인 유사도는 -1 이상 1 이하의 값을 가지며 값이 1에 가까울수록 유사도가 높다고 판단할 수 있습니다. 이를 직관적으로 이해하면 두 벡터가 가리키는 방향이 얼마나 유사한가를 의미합니다.



코사인 유사도 : -1

코사인 유사도 : 0



코사인 유사도 : 1