- Collaboraive Filtering(CF) – 협업필터링

- 가장 유명한 추천 방식
- 과거에 비슷한 취미를 가진 이용자들은 미래에도 비슷한 취미를 가질 것이라는 가정을 전제

- Pure CF Approaches

- Input : 사용자, 아이템의 등급 matrix 사용
- Output : 사용자가 특정 항목을 좋아할지 싫어할지에 대한 예측값, 상위 N개 추천 항목

- User-based nearest-neighbor collaborative filtering

The basic technique
엘리스(활동하고 있는 이용자), 아이템 I (엘리스가 보지 못한)
- 과거 엘리스와 같은 물건을 좋아하고 아이템 i에 평점을 준 사용자 집단 찾기
- 엘리스가 아이템 i를 좋아하는지 예측하기 위해서 사용자 집단의 평균 평점을 이용하기
- 엘리스가 보지 못한 아이템에 대해서 위에 같이 똑같은 작업을 하고 최고평점 아이템을 추천해 주기

Basic assumption and idea
- 사용자의 과거의 취향이 미래에도 비슷한 취향을 가질 것이다.
- 사용자의 선호는 시간이 지나도 일정하게 유지될 것이다.

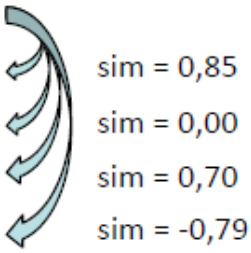
	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

- Alice의 Item5 점수 확인하기
- 점수를 예측하기 전 3가지 의문
  : 유사성은 어떻게 측정할 것 인가?
    얼마나 많은 이웃들을 고려해야 되는가?
    이웃들의 평점으로부터 어떻게 예측을 할 것 인가?

- **A popular similarity measure in user-based CF: Pearson correlation**
  - a, b : users
  - $r_{a,p}$: rating of user a for item p
  - P : set of items, rated both by a and b
  - Possible similarity values between −1 and 1

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1



- **A common prediction function:**

$$pred(a,p) = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \overline{r_b})}{\sum_{b \in N} sim(a,b)}$$

- Alice가 평점을 매기지 않은 item i가 이웃한 집단의 평점보다 높은지 않은지 계산하기
- 평점들의 차이를 결합한다. (유사성을 이용하여 가중치를 부여한다.)
- 사용자들의 평균 평점에 이웃집단의 bias를 추가하거나 삭제하여 예측하는데 이용하기

- Metrics 향상하기 / 예측함수

평점에 대한 가치는 모두 동등하지 않다.
- 일반적으로 선호하는 아이템의 Agreement은 그렇지 않은 아이템들의 Agreement보다 유익하지 않다.
- 이를 해결하기 위해 variance가 높은 아이템들에게 가중치를 부여한다.

*동일 등급 항목 수의 값
- 유의한 가중치를 부여한다. Co-rated의 수가 적을 때 가중치를 선형적으로 줄여나간다.

*case 증폭(amplification)
- 비슷한 값을 가진 neighbors(유사성이 1에 가까운)에 대하 좀더 가중치를 주어라

*Neighborhood selection
- 유사성 임계점(threshold) or neighbors의 수를 고정

- 메모리 기반과 모델 기반의 접근법

*사용자 기반 협업 필터링 (메모리기반 접근법)
- the rating matrix는 바로 neighbors를 찾고 예측을 하는데 사용된다.
- 대부분의 실제 세상만큼 크기가 확장되지 않는다.
- 대형 전자상거래사이트는 수천만 명의 고객과 수백 개의 items가 있다.

*아이템 기반 협업 필터링 (모델기반 접근법)
- 오프라인 사전처리 또는 모델학습 단계에 기반한다.
- 학습된 모델만이 예측을 할 수 있다.
- 모델들의 업데이트와 재-학습을 정기적으로 해야 한다.
- 크고 다양한 기술들이 사용된다.
- 모델 구축과 업데이트가 계산적으로 비싸다.

- 아이템 기반 협업 필터링
  - 아이템들간의 유사성을 이용해서 예측을 만들어낸다.

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

-item5와 유사한 item을 찾아서 유사성을 측정한다.
-Alice의 평점을 예측한다.

- Cosine 유사도 측정

-Item-to-Item filtering에서 더 좋은 결과
-Rating들은 n차원의 vector로 표현된다.
-유사성은 vector들 사이의 angle에 의해 계산된다.

$$sim(\vec{a},\vec{b}) = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}| * |\vec{b}|}$$

-조정된 cosine 유사도 (평균 사용자 등급을 고려하여 원래 등급을 변환)

$$sim(\vec{a},\vec{b}) = \frac{\sum_{u \in U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_u})^2}\sqrt{\sum_{u \in U}(r_{u,b} - \overline{r_u})^2}}$$

- 예측하기

$$pred(u,p) = \frac{\sum_{i \in ratedItem(u)} sim(i,p) * r_{u,i}}{\sum_{i \in ratedItem(u)} sim(i,p)}$$

-Neighborhood size는 일반적으로 특정 크기로 제한된다.
-모든 Neighbors는 예측에 이용되지 않는다.
-MoveiLens dataset의 분석에서 실제 상황에서 20개 ~ 50개의 neighbors의수가 가장 합리적이다.

- 아이템 기반 필터링을 위한 pre-processing

-item-base filtering은 그 자체로 확장성 문제를 해결하지 못한다.
-pre-processing approach by Amazon.com(in 2003)
모두 쌍별로 유사성을 미리 계산한다.
Run-time에 사용되는 The neighborhood는 item에 대해서 평점을 준 사용자들을 대상으로 하기에 때문에 일반적으로 작다.
Item의 유사성은 user의 유사성보다 더 안정적이어야 한다.

- Explicit ratings

-사용자 본인 얼마나 item에 호감이 있는지를 수치로 feedback주는것
-가장 정확하고 일반적으로 이용하는 rating (1 to 5, 1 to 7)
-Research topics
규모의 최적화 세분화 : 영화산업에서는 10-point scale이 사용된다.
(이산형에서 정확한 손실이 없다, 사용자의 선호를 좀더 잘 포착할 수 있다)

-Main problems
사용자들은 항상 많은 items에 대해서 평점을 주지 못한다.
사용자들을 어떻게 더 많은 item에 대해서 평점을 줄 지가 문제이다.

- Implicit ratings

-사용자가 직접적인 점수를 주는 대신 사용자의 간접적인 정보만을 제공한 것들
-추천 시스템이 포함한 인터넷 상점이나 어플리케이션에서 수집된다.

-고객들이 물건을 살 때 추천시스템은 이러한 행동들을 긍정적인 rate로 해석한다. (Clicks, page views, time spent on some page, demo downloads)
-Implicit rating은 끊임 없이 수집이 가능하고 사용자 측에서 추가적인 노력이 필요하지 않다.

-Main problem
1. 사용자의 행동들을 정확하게 해석하지 못한다.
2. 조금만 클릭하거나 영상을 듣더라고 구매하기 때문에 matrix의 거의 대부분은 negative observation이 positive obsercation의 수를 압도한다. 이러한 점을 고려하지 않으면 model이 ovefitting이 된다.
3. 또한 물건을 구매하였더라도, 이 물건에 대해서 반드시 긍정적으로 생각할 것이라 기대할 수는 없을 것이다.