**Week 3**

**2.4 Clustering**

Data의 크기가 매우 커질 경우, 시간복잡도 문제 발생.

이를 해결하기 위해 offline phase에 clustering phase를 추가한다.

Clustering phase의 기본은 K-means에 기본점을 두고 있다.

임의로 k개의 대표를 설정 후, 점별로 가장 가까운 대표를 cluster에 일단 두고

각 cluster별로 평균점을 다시 구해 이를 반복하여 clustering 시키는 방법 사용.

Top-k를 구할 때, 같은 cluster내에서만 선별하게 된다

이를 사용할 경우, 정확도는 조금 내려가지만 효율성이 극대화. 매우 practical하다

**질문사항, 거리 계산시 서로 포인트만 있는 점으로 L1-measure나, L2-measure를 사용하는데**

**애초에 Nan-Euclidean distance는 쓸 수 없을까?**

**2.5 Dimension Reduction**

Sparse한 행렬에 이를 적용하여 좀 더 dense한 low-dim matrix를 얻게 된다.

Latent factor model이라고도 불리기도 한다.

**M \* N 행렬을 축소하여, M \* D 행렬로 축소하여, 이를 바탕으로 Sim을 게산하는 아이디어를 쓴다.** 이를 위해서는 먼저 rating matrix의 빈 칸을 채워야 한다. (평균으로 채우던가, svd로 예측)

But, 이 문제에서 **Missing val을 채우면서 bias가 발생**하게 된다.

이에 따라, 상관관계가 뒤틀려 상관관계가 역전되는 경우도 존재. (책의 예제 존재)

이를 해결하는 방법으로

- Maximum Likelihood of the Cov. Matrix

- Direct Matrix Factorization (직접적으로 rating matrix에 바로 SVD 적용)

**\* 질문) 앞에서 말하는 Matrix factorization이랑 bias에서 말하는 Direct의 차이는?**

**2.6 Regression Modeling View**

지난주에 봤던, user-based의 rating 예측식은 사실상, top-k user들에 가중치를 부여, 즉 선형 결합의 방식으로 해석가능하며, 이에 대해 선형 회귀를 푸는 문제로 변형 가능하다.

다시 말해, Objective funtion의 MSE를 계산하여 이를 최소화 시키는 문제로 해결 가능하다.

(이에 대한 closed form은 4장에서 배우게 된다)

**\* SLIM Model이란?**

- 기존 Linear Model과 다르게, 거의 모든 nonzero coeff에 사용 가능.

- implicit feedback dataset를 위해 design 됨. 혹은 positive-preferences only에 사용

- coeff in SLIM은 non-negative

- predicted val -> ranking the items을 위함.

**2.7 Graph Model**

Sparsity 문제를 해결하는 대표적인 방법으로 Graph Model을 들 수 있다.

**User-item의 경우**

기본적인 model은 user / item 끼리 연결하는 bipartite graph로 표현한다.

여기에서 random-walk나 katz-measure을 적용하여 간접적 연결성을 measure화 할 수 있다.

**User-User의 경우**

실제로 user-user간 거리 2로 연결되는 상황이 아니라 4, 6, 8…등 간접적으로 건너건너 연결되는 것을 계산할 수 있게 될 것이다.(indirective connectivity)

User-user간 최단 경로를 계산하며, user v가 u를 예측할 수 있는 조건들이 있다.

- horting (교집합이 특정 값 이상인가)

- predictability (horting 만족시, rating에 특정한 선형함수를 적용하여 교집합 원소 개수로 나눈 값이 특정 값 이상인지 검토),

위에서 본 f를 활용하여 shortest path를 따라가며 Linear transform들을 순차적으로 합성하여 rating을 예측한다.

**Item-item의 경우**

Correlation graph를 고려하게 된다. Weight는 하나의 교집합 이상이 있을 경우 존재. 양방향으로 network를 만들게 된다.  
(1) Ui, Uj 교집합 원소의 개수로 가중치 초기화 후 normalized -> 여기서 symmetric 깨짐  
(2) 이것들이 random-walk probabilities가 된다.

**2.8 ~ Summary**

Summary

Collaborative filtering은 분류/회귀가 일반화 된 문제.

User-based에서는 (1) target user의 이웃 결정(sim 활용)

(2) (1)을 바탕으로 rate 예측.

Item-based도 동일하게 하나 diverse한 추천은 어려움.

속도 향상을 위해 clustering활용.

Nbd-based Method를 linear model로 볼 수 있음.

Weight는 sim val값을 활용하여 선택 가능.

혹은 linear regression models를 활용하여 weight 추정 가능.

Data sparsity를 해결하기 위해 dimension reduction이나 graph-based 활용

(graph-based에서는 random walk나 shortest-path를 활용)