

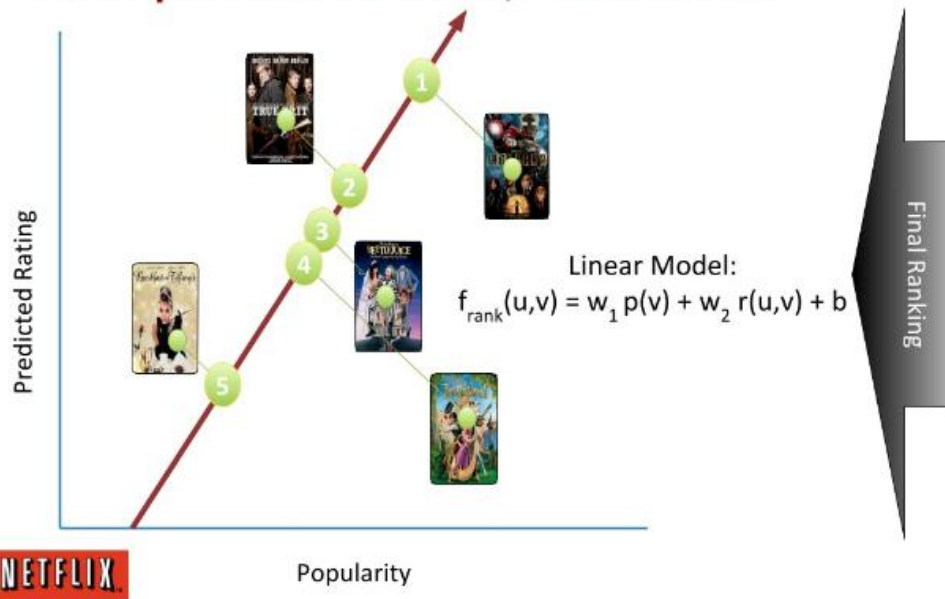
Novel approaches to Recommendation : beyond "traditional" CF and CB

<https://www.youtube.com/watch?v=mRToFXINBpQ&feature=youtu.be>

Learning to Ranking

- 추천시스템을 통해 추천을 해줄 때는 선호도(Popularity)가 가장 기본
- 추가적으로 Ratings prediction을 사용하기도한다.
- RMSE를 사용하지 않고 Ranking으로 상품을 추천해보자

Example: Two features, linear model



- training data로 ranking model 만드는 것이 최종 목표이다.
- item 순서의 결과는 숫자형으로 나타날 것이며
- 표준 지도학습모델을 다루는 것과 유사하다.

Quality of ranking을 측정하는 metrics

- Normalized Discounted Cumulative Gain
- Mean Reciprocal Rank(MRR)
- Fraction of Concordant Pairs (FCP)

Approaches

1.Pointwise : 가장 일반적으로 사용하는 지도학습 분류모형 ex) Logistic Regression, SVM, GBDT

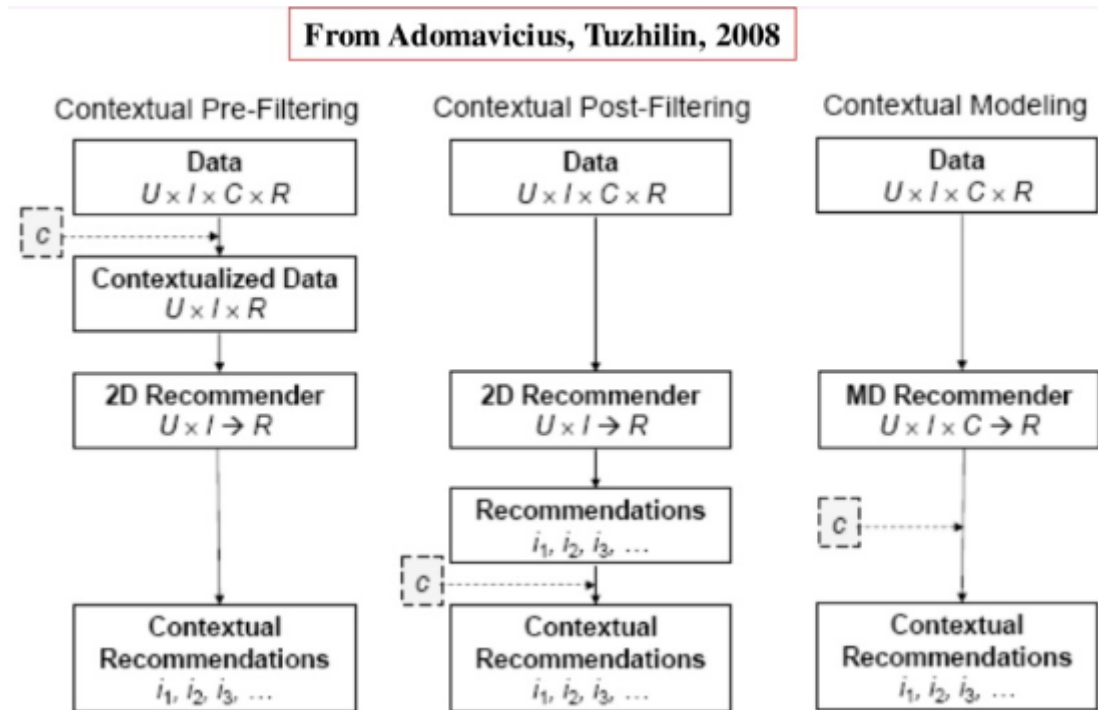
2.Pairwise : loss함수는 pair-wise preferences에서 정의되었다. 순위의 역전횟수를 최소화하는 것이 목표이다. Ex) RankSVM, RankBoost, RankNet, Frank

3.Listwise : 어떤 metrics이든 상관없이, 전체 category item을 optimize하는 것이다.

Measure하는 방법(NDCG, MRR, FCP)들의 차별점이 없기에 주어진 행렬을 그냥 optimize하는 것은 어렵다. Indirect Loss function방법사용

*Indirect Loss function- RankCosine, ListNet등의 방법이 있다. Indirect 방법으로 optimizing을 하여도 그 성능은 완전히 보장하지 못한다.

Context-aware Recommendations



Pre-Filtering Techniques

-input data에 context데이터를 포함시켜 모델링을 한다.

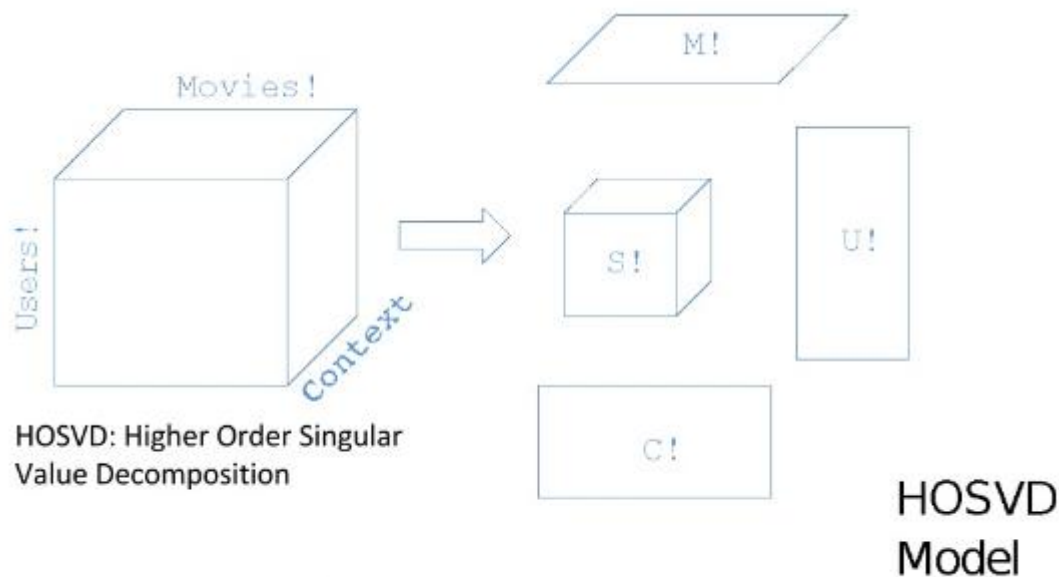
Post-Filtering Techniques

-모델링을 한 뒤에 context데이터를 포함시켜 추천을 실시

Contextual modeling

-모델학습 단계에서 contextual 정보를 바로 사용한다. Multi-dimensional recommendation models 기존의 two-dimension model에서 N-dimension model로 변화 (user, movie, time, company, place)

Tensor Factorization



$$U \in \mathbb{R}^{n \times d_U}, M \in \mathbb{R}^{m \times d_M} \text{ and } C \in \mathbb{R}^{e \times d_C}$$

$$S \in \mathbb{R}^{d_U \times d_M \times d_C}$$

1. Tensor 큐브를 분해한다.

$$F_{ijk} = S \xrightarrow{U} U_{ik} \xrightarrow{M} M_{jk} \xrightarrow{C} C_{k\kappa}$$

$$\Omega[F] = \lambda_M \|M\|_F^2 + \lambda_U \|U\|_F^2 + \lambda_C \|C\|_F^2$$

$$R[U, M, C, S] := L(F, Y) + \Omega[U, M, C] + \Omega[S]$$

$$\Omega[S] := \lambda_S \|S\|_F^2$$

2. Loss function 식을 구하고 이를 정규화 시켜준다.

$$l(f, y) = \frac{1}{2}(f - y)^2 \quad l(f, y) = |f - y|$$

3. simple squared error 혹은 absolute error 값을 구한다.

4. partial gradients를 반복적으로 사용하여 parameter matrix와 tensor를 update 시킨다.

Ex) context 데이터를 포함하지 않은 matrix와 tensor (context 데이터를 포함한) factorization을 비교해보았다. Context를 포함한 matrix에서 낮은 error를 보였다.

Factorization Machine

-정규화된 matrix(tensor) factorization 방법에서 선형회귀 방법을 결합시켰다.

-다항 회귀모델에서 사용되며, feature들간에는 interaction이 있다.

$$f(\mathbf{x}) = b + \sum_{i=1}^d w_i x_i + \sum_{i=1}^d \sum_{j=i+1}^d x_i x_j \mathbf{v}_i^T \mathbf{v}_j \qquad \mathcal{O}(d^2)$$

$$= b + \sum_{i=1}^d w_i x_i + \frac{1}{2} \sum_{f=1}^k \left(\left(\sum_{i=1}^d x_i v_{i,f} \right)^2 - \sum_{i=1}^d x_i^2 v_{i,f}^2 \right) \quad \mathcal{O}(kd)$$