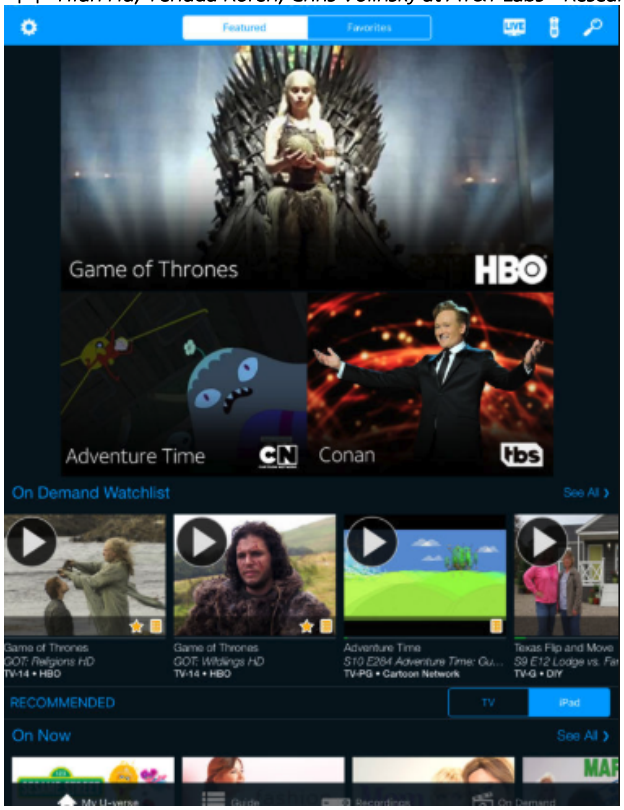
[추천시스템] Implicit Feedback

Info.

- 본 문서는 데이터의 유형중 하나인 Implicit feedback을 이용해 추천시스템을 구축하는 이론과 방법을 정리한 문서임
- 참고논문은 아래 2개임
  - Collaborative Filtering for Implicit Feedback Datasets
  - Logistic Matrix Factorization for implicit feedback data

작성완료

1. 목적
   a. Explicit feedback 및 Implicit feedback을 구분한다.
   b. 각 데이터에 따른 MF 모델 적용 방식을 이해한다.
2. The first paper
   a. 제목: Collaborative Filtering for Implicit Feedback Datasets
   b. 저자: *Yifan Hu, Yehuda Koren, Chris Volinsky at AT&T Labs - Research (based on TV Shows)*
   c. 

3. Introduction
    a. Explicit feedback?
        i. 유저가 아이템에 대해 명시적으로 선호도를 입력한 값
        ii. 예) 넷플릭스의 영화 평점 (1~5)
        iii. 단, 현실적으로 대부분의 아이템/유저에 대한 데이터 수집이 어려움
            1. 일부 데이터로만 학습할 경우 overfitting 가능성
            2. 따라서 Implicit Feedback 활용 필요
    b. Implicit Feedback?
        i. 유저가 아이템에 대해 암묵적으로 남긴 기록/데이터
        ii. 예) 클릭 및 구매기록, 검색 패턴, 체류시간 등
    c. Implicit Feedback's Characteristics
        i. *"No negative feedback"*
            1. Explicit 기반 모델에서는 명시적으로 수집된 데이터에 한정해서 모델링
            2. likes or dislikes가 명확히 구분되지만 많은 missing values 발생하여 편향된 결과가 발생할수도 있음 → missing value 고려 필요
            3. 그러나 implicit feedback에서는 missing values를 dislike 로 판단하기 어려움 (잠재적으로 선호할 아이템이나 존재를 몰라서 negative로 수집된 상태)
        ii. *"Implicit feedback is inherently noisy"*
            1. Implicit feedback으로 선호 아이템을 추정하는 것은 항상 오류를 내포
            2. 예) 구매기록을 이용할 경우 → 본인이 아닌 타인을 위한 선물로 구매한 가능성
        iii. *"The numerical value of explicit feedback indicates preference, whereas the numerical value of implicit feedback indicates confidence"*
            1. Implicit Feedback은 Explicit feedback처럼 preference를 나타내지 않지만, 유용함 (특히 일회성이 아니 반복적 행동 특성일 경우)
        iv. "Evaluation of implicit-feedback recommender requires appropriate measures"
            1. 평가 지표를 개발할때 다음과 같은 부분을 고려해야 함
            2. item availability, 다른 아이템과의 상호작용, 반복적 피드백 등
4. Previous Work
    a. Neighborhood models
        i. Similarity: Pearson correlation

$$\hat{r}_{ui} = \frac{\sum_{j \in S^k(i;u)} s_{ij} r_{uj}}{\sum_{j \in S^k(i;u)} s_{ij}}$$

        ii.
        iii. implicit feedback 데이터로 similarity 를 명확히 계산하기 모호한 부분이 있음
        iv. 행동 데이터에 대한 scale이 다른 점도 고려사항
    b. Latent Factor models
        i. SVD

$$\min_{x_\star, y_\star} \sum_{r_{u,i} \text{ is known}} (r_{ui} - x_u^T y_i)^2 + \lambda(\|x_u\|^2 + \|y_i\|^2)$$

        ii.
        iii. 본 논문의 Implicit Feedback model을 개발할때 차용한 식
5. The researcher's Model
    a. pui: Binary variable

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases}$$

i.
ii. 만약 Pui가 0일 경우 부정적인 피드백과 더불어, unaware of the existence of the item, limited availability 등의 가능성을 내포
iii. 1일 경우도 반드시 선호하는 것이 아닌 다양한 여러 가능성을 내포하는 것으로 고려

b. cui: confidence level

$$c_{ui} = 1 + \alpha r_{ui} \qquad \text{또는} \qquad c_{ui} = 1 + \alpha \log(1 + r_{ui}/\epsilon).$$

i.
ii. if more evidence is observed, cui increases
iii. a는 40일 경우 보통 좋은 결과를 보임 (일반적인 케이스) → 상수이므로 최적화 과정에서 바뀌지 않음

c. cost function

$$\min_{x_\star, y_\star} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

i.
ii. inner products: $x^T y$ → Matrix Factorization과 동일

d. Solver (ALS)
i. cui 부분이 계산량 증가시키므로 대안안이 제시됨

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u)$$

ii.

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i)$$

iii.

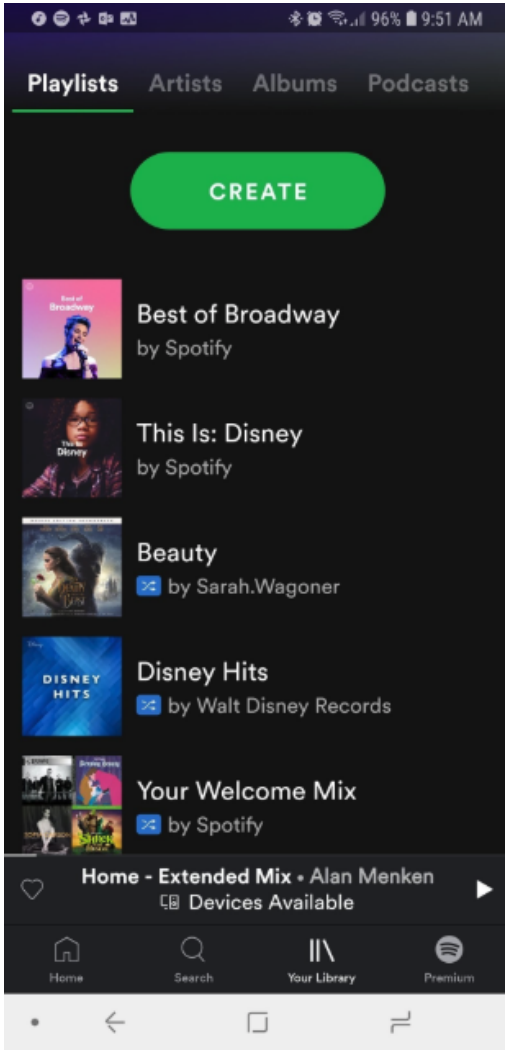
e. ⭐ Main Properties
i. raw observation을 Pui, Cui 으로 변환 → 데이터에 대한 본질적 접근 & 정확성 향상
ii. m x n 사이즈가 매우 큼 → 선형대수를 이용한 해(solution) 도출법 활용

6. Experimental study & Discussion
a. 생략

---

7. The second paper

a. 제목: Logistic Matrix Factorization for implicit feedback data
b. 저자: Christopher Johnson (at Spotify)

c. 

8. Introduction
    a. *"The goal of recommender systems is to analyze data associated with users and products in order to provide users with personalized recommendations"*
    b. 개인화 추천모델 종류
        i. Baseline model
        ii. Collaborative Filtering
            1. Neighborhood Models (User-based CF, Item-based CF)
            2. Latent Factor Models(Matrix Factorization, SVD)
        i. Content-Based Recommendation
        ii. Content-based
        iii. Association Rules
    c. 최근에는 Implicit feedback 이용한 추천모델 개발이 활발히 진행중 (click, page views, purchases 등)
        i. 직접 유저에게 물어볼 필요가 없으며 비교적 데이터가 많음 (explicit model 대비 missing values 적고 수집이 용이)

d. 이 논문에서는 MF에 기반한 새로운 implicit feedback 데이터 활용법을 제안함

9. Problem Setup
   a. Matrix's Space 부분 → impute with zero
   b. 0의 의미는 아직 모른다는 의미일뿐 Negative의 의미는 아님
   c. 수집된 Implicit feedback 데이터에 가중치 활용 가능
      i. 자동스트림 < 클릭후 스트림
      ii. 과거 데이터 < 최근 데이터
   d. Our Goal → *Rui = 0 (unknown) 인 아이템에 대해 Top K의 recommended items을 찾는 것*
10. Related Work
   a. Previous: Neighborhood, Matrix Factorization with explicit feedback datasets..
   b. Implicit feedback을 이용하는 여러 모델이 개발
      i. Collaborative Filtering for Implicit Feedback Datasets, One-Class Collaborative Filtering
      ii. Hybrid approaches: context or temporal information
      iii. Probabilistic approaches: Poisson distributions, Gaussian 등 분포를 활용해 유저 반응을 추론
      iv. 본 논문도 Logistic function을 이용한 Probabilistic approaches 중 하나임
11. Logistic MF (sigmoid)
   a. 로지스틱 회귀

$$p(l_{ui} \mid x_u, y_i, \beta_i, \beta_j) = \frac{\exp(x_i y_i^T + \beta_u + \beta_i)}{1 + \exp(x_u y_i^T + \beta_u + \beta_i)}$$

   b.
   c. Logistic MF는 RMSE를 최소화하는 최적화(with GD, ALS)대신, 확률적 접근을 시도
   d. Confidence level 이용

$$c_{ui} = 1 + \alpha r_{ui} \qquad c_{ui} = 1 + \alpha \log(1 + r_{ui}/\epsilon).$$

      i. 또는
      ii. Increasing α places more weight on the non-zero entries while decreasing α places more weight on the zero entries
   e. Likelihood given that u, i are independent (회귀계수 추정)
      i. Note (Bernoulli trial's likelihood)

$$L = \prod_i p^{y_i}(1-p)^{1-y_i} \qquad p(X = 56|\theta = 0.5) = \binom{100}{56} 0.5^{56} 0.5^{44} \approx 0.0389$$

         1. →
      ii. Logistic Regression's likelihood
         1. Set basiclikelihood function, where ar_{ui}: confidence

$$\mathcal{L}(R \mid X, Y, \beta) = \prod_{u,i} p(l_{ui} \mid x_u, y_i, \beta_u, \beta_i)^{\alpha r_{ui}} (1 - p(l_{ui} \mid x_u, y_i, \beta_u, \beta_i))$$

            a.
         2. Take the log of posterior

$$\log p(X, Y, \beta \mid R) = \sum_{u,i} \alpha r_{ui}(x_u y_i^T + \beta_u + \beta_i) - (1 + \alpha r_{ui}) \log(1 + \exp(x_u y_i^T + \beta_u + \beta_i)) - \frac{\lambda}{2}\|x_u\|^2 - \frac{\lambda}{2}\|y_i\|^2$$

         3.
      iii. Set Objective

$$\arg\max X, Y, \beta \log p(X, Y, \beta \mid R)$$

         1.

12. 모델 테스트
    a. Surprise 라이브러리에 위 논문의 로직이 구현된 함수가 없음 → SVD++ 이용
    b. Code

```
## SVD++
# split the dataset
trainset, testset = train_test_split(data, test_size=.25)

# grid search
algo = SVDpp()

algo.fit(trainset)
test_pred = algo.test(testset)

cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=5, verbose=True)

print(accuracy.rmse(test_pred, verbose=True))
print(accuracy.mae(test_pred, verbose=True))
```

```
Evaluating RMSE, MAE of algorithm SVDpp on 5 split(s).
                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
MAE (testset)     1.5771  1.5921  1.5858  1.5840  1.6010  1.5880  0.0081
RMSE (testset)    2.9879  3.1149  3.0444  3.0717  3.2179  3.0874  0.0772
Fit time          184.07  180.57  179.41  179.29  179.77  180.62  1.78
Test time         4.28    4.18    4.43    4.32    4.21    4.28    0.09
RMSE: 3.2136
3.2135722264969813
MAE:  1.5919
1.5919205728175867
```

    c.
13. Next Step
    a. 현재의 user x item 매트릭스가 아닌 profiling 혹은 factor를 별도로 정의해서 매트리스를 구성
    b. 세미나 주제 선정후 공유
        i. MF(Matrix Factorization) 확장 모델 조사
        ii. Learning to Rank