OCCF

Collaborative Filtering for Implicit Feedback Datasets

이수연

2022.01.09

Contents

Introduction

Recommender system

track user behavior → model user preference
→ personalized recommendations

strategy

content based filtering

collaborative filtering

movie profile
(genre, actors, popularity)

user profile
(previously watched movie)

user
(past behavior)

like-minded users
(similar history)

→ gathering external information might not be available or easy to collect

→ no explicit profile, domain free, generally more accurate. But cold start problem

Introduction

Recommender system

input type

0 or 1

→ **one-class** collaborative filtering(OCCF)

Explicit feedback

- star rating
- thumbs-up/down

→ convenient, but not always available

Implicit feedback

- purchase history
- browsing history
- search pattern
- mouse movement

→ with the user's approval, abundant

Introduction

Implicit feedback

the unique characteristics of implicit feedback
different with explicit feedback

1. **No negative feedback**
   did not watch a certain show ← disliking it? not knowing it? or not available to watch it?
   0: where most negative feedback is expected to be found, not missing data

2. **Inherently noisy**
   1: does not necessarily indicate a positive view… a gift or disappointed

3. **Confidence**
   The numerical value of implicit feedback indicates confidence, not preference.
   A recurring event is more likely to reflect the user opinion.

4. **Needs appropriate measure**
   Instead of RMSE, new performance measure to take into account availability, competition, repeat feedback…

Preliminaries

indexing letters

- user *u, v*
- item *i, j*

input data: **observation $\mathbf{r_{ui}}$**

- In this TV recommender case, it indicates how many times *u* fully watched show *i*.

  ex) 100분 중 70분 시청 → 0.7, 두 번 시청 → 2
- If no action was observed r_{ui} is set to zero.

Previous work

- Collaborative Filtering
  - Neighborhood
    - user-oriented
    - item-oriented
  - Latent factor

estimate unknown ratings...

user-oriented: based on recorded ratings of **like minded users**

item-oriented: using known ratings made by the same user on **similar items**

→ better scalability & improved accuracy,
more amenable to explaining the reasoning behind predictions.

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

→ no distinction between preference and confidence

Previous work

Latent Factor Collaborative Filtering

latent factor in the observation matrix
→ matrix factorization → recommendation

Explicit feedback

observation r_{ui}

→ Singular Value Decomposition

(accuracy & scalability)

user-factors vector $x_u \in \mathbb{R}^f$
item-factors vector $y_i \in \mathbb{R}^f$

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

prediction $\hat{r}_{ui} = x_u^T y_i$

→ implicit feedback: **observation ≠ preference**
model formulation & optimization technique에서 modification 필요

Our model

Latent Factor Collaborative Filtering

Implicit feedback

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

controlling the rate of increase of confidence, experimentally α=40

minimal confidence

confidence in observing p_{ui}, $c_{ui} = \mathbf{1} + \boldsymbol{\alpha} r_{ui}$

prediction $\hat{p}_{ui} = x_u^T y_i$

necessary for regularization
λ is data-dependent and determined by cross validation.

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

$\mathrm{m} \cdot \mathrm{n}$ term $(x_u^T y_i)$ → when either x_u or y_i are fixed, the cost function becomes quadratic
→ **Alternating-Least-Squares**(ALS) optimization process

Our model

Alternating-Least-Squares(ALS)

x, y 중 하나를 고정(상수 취급) → quadratic form (convex)

$$\frac{\partial L(x_u)}{\partial x_u} = 0 \rightarrow \; x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u)$$

$$\frac{\partial L(y_i)}{\partial y_i} = 0 \rightarrow \; y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i)$$

⇒ 10번 정도 paired recomputation 후,
$\hat{p}_{ui} = x_u^T y_i$가 가장 큰 K개의 items 추천

$C_{ii}^u = c_{ui}$
diagonal

C^u Confidence matrix for user u

$C_{uu}^i = c_{ui}$
diagonal

C^i Confidence matrix for item i

$Y^T C^u Y$: $O(f^2 n)$ *for each m users*

$= Y^T Y + Y^T(C^u - I)Y$: $O(f^2 n_u)$, where $n_u \ll n$

$(Y^T C^u Y + \lambda I)^{-1}$: $O(f^3)$

$\rightarrow$ $O(f^2 \mathcal{N} + f^3 m)$, $\mathcal{N} = \sum_u n_u$,

linear in the size of the input

Explaining recommendation

"Good recommendation should be accompanied with an explanation."

However, for latent factor model all past user actions are abstracted via the user factors.

$$\begin{aligned}\hat{p}_{ui} &= y_i^T \boldsymbol{x_u} \\ &= y_i^T (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u) \\ &= y_i^T \underbrace{\boldsymbol{W^u}}_{\text{weighting matrix}} Y^T C^u p(u) \\ &= \textstyle\sum_{j:r_{uj}>0} \boldsymbol{s_{ij}^u} c_{uj}\end{aligned}$$

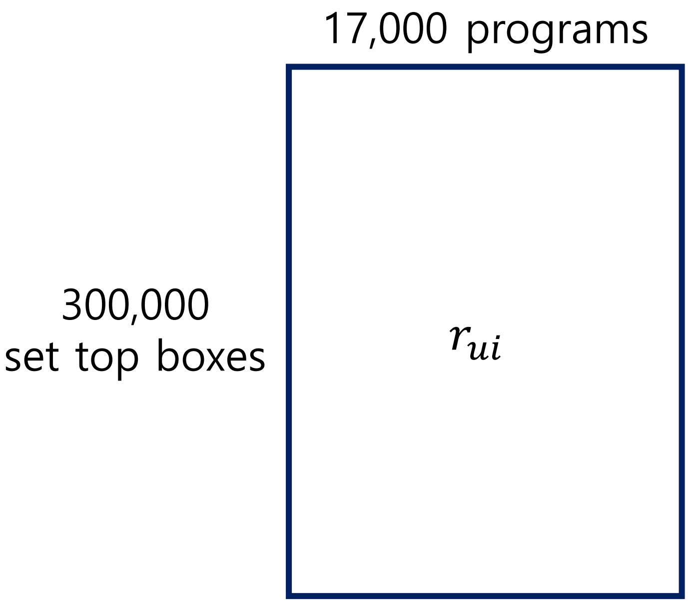
Alternating-Least-Squares: $x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u)$

→ the similarity to target item i from **u's viewpoint** – $\boldsymbol{s_{ij}^u}$ & the significance of the relation to user u – $\boldsymbol{c_{uj}}$으로 각각의 item들에 대한 contribution 파악 가능

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

Experimental study

Data description



data from a digital television service

training data 4주 + test data 1주

- toggle to zero all entries with $r_{ui}^{t} < 0.5$
- “easy” prediction 제외
- momentum effect

subsequent show: down-weighting $\frac{e^{-(at-b)}}{1+e^{-(at-b)}}$

Experimental study

Evaluation methodology

not watching a program can stem from multiple different reasons

+ unable to track user reactions to our recommendations

→ precision based metrics are not appropriate

percentile-ranking: general and independent of the number of programs

$$\overline{rank} = \frac{\sum_{u,i} r_{ui}^t rank_{ui}}{\sum_{u,i} r_{ui}^t}$$

The lower, the better. (적어도 random prediction일 때의 50%보다는 낮아야 함)

unloved program은 제외 → 안 본 program에 대해서는 $r_{ui}^t = 0$으로, 계산에 포함 안 됨

Experimental study

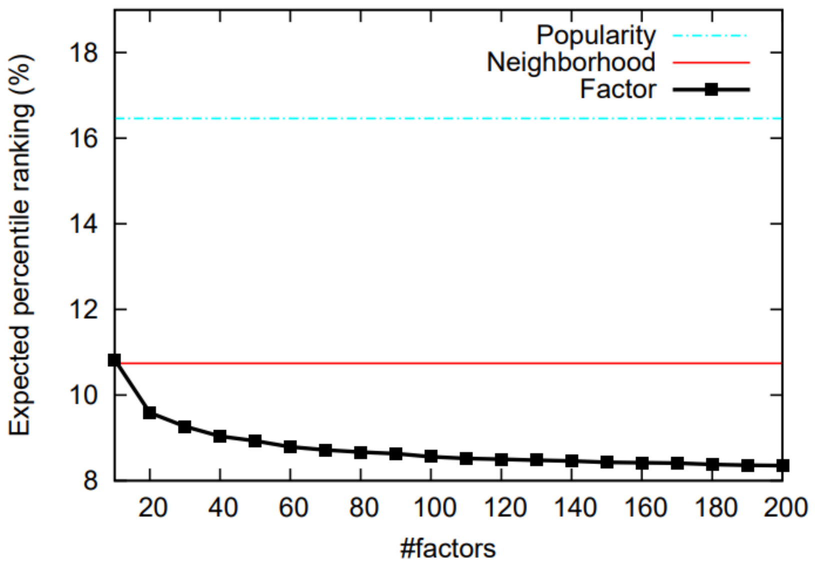
Evaluation results

Competing models

1. popularity ranking(baseline model): sorting all shows based their popularity

2. neighborhood based model: takes all items as "neighbors" and uses cosine similarity

predicted preference $\hat{p}_{ui} = \sum_j s_{ij} r_{uj}$ (모든 user에 대해 똑같은 similarity 값을 가짐)



16.46%

10.74%

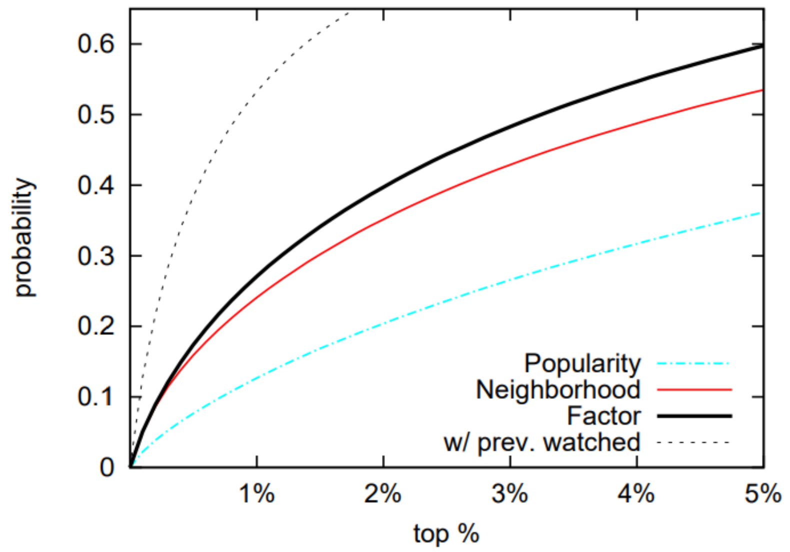
8.35%

Experimental study

Evaluation Results

Cumulative function: rank를 이용하여 quality of recommendation을 평가하는 또다른 방법
"top rank에 얼마나 많은 watched program이 포함되어 있는가"

the dotted line: previously watched program을 제거하지 않으면 probability는 매우 높아진다.



Discussion

- collaborative filtering on datasets with **implicit feedback**

- observation → two paired magnitudes: **preferences** and **confidence** levels

- latent factor algorithm taking **all user-item preferences** as an input
  → scalability issues… by exploiting the algebraic structure of the model, leading to an algorithm that scales **linearly with the input size**

- **explaining** the recommendations

- extension of the model - adding **a dynamic time variable** addressing the tendency of a user to watch TV on certain times

- the purpose of a recommender system: to point users to items that they might not have otherwise purchased or consumed
  → in depth **user study and surveying**

감사합니다