Comparison of online and offline evaluation metrics in Recommender Systems

Petr Kasalický

Department of Applied Mathematics
Faculty of Information Technology
Czech Technical University in Prague

September 30, 2021

Outline

1. Recommender System - overview
   - Definition
   - Source data
2. Offline evaluation
   - Rating prediction-based evaluation
   - Ranking-based evaluation
3. Online evalution
   - A/B test
   - Click-through rate

Outline

1. Recommender System - overview
   - Definition
   - Source data
2. Offline evaluation
   - Rating prediction-based evaluation
   - Ranking-based evaluation
3. Online evalution
   - A/B test
   - Click-through rate

Recommender System - Definition

- *"The goal is to generate meaningful recommendations to users for items that might interest them"* [1]
- Requirements are more and more complex
    - Evaluation methods/metrics need to adapt

Source data

- Attributes of items and users
  - Content-based recommendation
- Interactions (feedback) between users and items
  - Explicit vs implicit
  - Interaction attributes
    - Timestamp
    - Type of interaction (click, purchase, rating)

$$F = \{f_1, \dots, f_p\}$$

$$f_j \in (U \times I \times \mathbb{Z}_t \times \mathbb{R}_v) \text{ for } \forall j \in \{1, \dots, p\}$$

  - Missing-Not-A-Random (MNAR) problem
  - Cold-start problem

Interaction (rating) matrix

$$F_{u,i} = \{(u_j, i_j, t_j, v_j) \mid (u_j, i_j, t_j, v_j) \in F : u_j = u \wedge i_j = i\}$$

$$r_{u,i} = \zeta(F_{u,i}), \zeta : \{0,1\}^{|U|\times|I|\times\mathbb{Z}_t\times\mathbb{R}_v} \to \mathbb{R}_?$$

	item_247	item_837	item_196	item_161	item_919	item_594	item_632		item_138
user_5748	1				-1		0	• • •	1
user_3816	-0.5					0.5		• • •	
user_6491				-1				• • •	
user_8039		0.5	-0.5					• • •	0.25
user_2970					0.75			• • •	
user_6176	1						0.25	• • •	
user_1015			-1					• • •	0.5
			⋮				⋮	⋮	
user_7719		-1			0.5			• • •	-1

user's interaction vector

item's interaction vector

Outline

1. Recommender System - overview
   - Definition
   - Source data
2. Offline evaluation
   - Rating prediction-based evaluation
   - Ranking-based evaluation
3. Online evalution
   - A/B test
   - Click-through rate

Offline evaluation

- Based on already collected data
- Fixed set of interactions split to training and test subset

$$F = F^{train} \cup F^{test} \text{ for } F^{train} \cap F^{test} = \emptyset$$

Rating prediction-based evaluation

- The task is to predict missing ratings in the interaction table

$$\tau : \mathbb{R}_?^{|U|\times|I|} \rightarrow \mathbb{R}^{|U|\times|I|}$$

- Predicted ratings:

$$T = \{(\hat{r}_{u,i}, r_{u,i}) \mid i \in I, u \in U : r_{u,i} \neq ? \wedge r_{u,i} \in R^{test}\}$$

- MSE:

$$E^{MSE}(T) = \frac{\sum\limits_{\hat{r}_{u,i}, r_{u,i} \in T} (\hat{r}_{u,i} - r_{u,i})^2}{\mid T \mid}$$

- Similarly RMSE, MAE, NMAE etc.
- used until about 2010 when several experiments showed that minimising RSME may not improve the quality of recommendation algorithms in practical applications

Ranking-based evaluation

- Classification of relevant and not-relevant items for each user:

	classified as relevant	classified as not-relevant
truly relevant	true-relevant (TR)	false-not-relevant (FN)
truly not-relevant	false-relevant (FR)	true-not-relevant (TN)

- and measuring:

$$precision = \frac{TR}{TR + FR} \qquad \text{and} \qquad recall = \frac{TR}{TR + FN}$$

$$\text{F-score}_\gamma = (1 + \gamma^2)\frac{precision \times recall}{(\gamma^2 \times precision) + recall}$$

- Which items are truly relevant for a particular user?

Top-K recommendation

- Only K items can be labeled as relevant for one recommendation
- Most common scenario in practical applications
- Recall and other metrics are typically measured on subset of users:

$$U = U^{train} \cup U^{test} \text{ for } U^{train} \cap U^{test} = \emptyset$$

$$F^{train} = \bigcup_{\substack{u \in U^{train} \\ i \in I}} F_{u,i}$$

$$F^{test} = \bigcup_{\substack{u \in U^{test} \\ i \in I}} F_{u,i}$$

Recall measured using leave-one-out validation

- One item is hidden and it should be recommended based on the other items interacted by the user
- Does not follow the user's behavior over time, disadvantages *remainder* model

$$recall@K_{LOO} = \frac{\sum_{u \in U^{test}} \sum_{i \in RI_u^{test}} \left| \{i\} \cap Top(K, RI_u^{test} \setminus \{i\}) \right|}{\sum_{u \in U^{test}} \left| RI_u^{test} \right|}$$

- Protection against bot and long tailedness: user-normalization:

$$recall@K_{LOO}^{UN} = \sum_{u \in U^{test}} \frac{\sum_{i \in RI_u^{test}} \left| \{i\} \cap Top(K, RI_u^{test} \setminus \{i\}) \right|}{\left| RI_u^{test} \right|}$$

Recall measured using leave-last-one-out validation

- Computationally more expensive than LLOO
- Aggregation of interactions in sliding time window is recommended

$$recall@K_{LLOO} = \frac{\sum\limits_{\substack{u \in U^{test} \\ (i_1, t_1) \in F_u^{test}}} \left| \{i_1\} \cap Top(K, \{i_2 \mid (i_2, t_2) \in F_u : t_2 < t_1\}) \right|}{\sum\limits_{u \in U^{test}} \left| RI_u^{test} \right|}$$

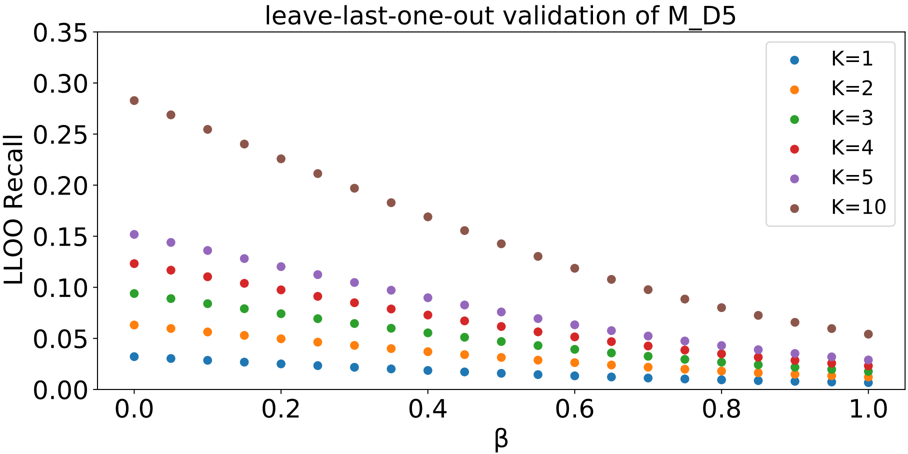
$$F_u = \{(i_j, t_j) \mid (u_j, i_j, t_j, v_j) \in F : u_j = u\}$$

Popularity-stratified LOO recall [2]

$$recall@K^{\beta,UN}_{LOO,PS} = \sum_{u \in U^{test}} w^{\beta}(u) \frac{\sum_{i \in RI_u^{test}} \left| \{i\} \cap Top(K, RI_u^{test} \setminus \{i\}) \right| \; p(i)^{-\beta}}{\sum_{i \in RI_u^{test}} p(i)^{-\beta}}$$

$$p(i) = \frac{\sum_{u \in U^{train}} \left| F_{u,i} \right|}{\sum_{j \in I} \sum_{u \in U^{train}} \left| F_{u,j} \right|}$$

- $\beta \in [0, 1]$ determines how much popular items should be penalised


leave-last-one-out validation of M_D5
LLOO Recall
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.0
0.2
0.4
0.6
0.8
1.0
β
K=1
K=2
K=3
K=4
K=5
K=10

Outline

1 Recommender System - overview
- Definition
- Source data

2 Offline evaluation
- Rating prediction-based evaluation
- Ranking-based evaluation

3 Online evalution
- A/B test
- Click-through rate

Online evalution

- Implicit vs explicit
- Typically expensive to measure (fake customers, engineers effort)
- Only few models can be measured
- Unrepeatable
- Examples: CTR, Conversion rate, Customer lifetime value

A/B test

- Common way to compare different versions of the system
- Users are divided into groups
- Each user group is presented with a different recommendation algorithm
- Necessary to be aware of robots, scrapers, and other nonhuman users

Click-through rate

$$CTR = \frac{\text{number of accepted recommendations}}{\text{number of recommendations}}$$

- An example of a short-term reward
- Significantly influenced by other factors

Implicit Click-through rate (iCTR)

- Hyperparameter d - size of the time window

$$REC : Z_t \times U \times \{0,1\}^I$$

$$iCTR(d) = \frac{\sum_{(t,u,I') \in REC} sgn(\mid I' \cap F_u(t,d) \mid)}{\mid REC \mid}$$

$$F_u(t,d) = \{i_j \mid (i_j, t_j) \in F_u : (t_j >= t) \wedge (t + d >= t_j)\}$$

References

1 Melville P., Sindhwani V. (2011) Recommender Systems. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA.

2 Harald Steck. 2011. Item popularity and recommendation accuracy. In Proceedings of the fifth ACM conference on Recommender systems (RecSys '11). Association for Computing Machinery, New York, NY, USA, 125–132.