IT492: Recommendation Systems

Lecture - 15

Re-ranking Approaches

Arpit Rana

28th Mar 2022

Recommender System Architecture (RSA)

Recommender systems typically proceed through (at least) three steps:

- Candidate generation
- Scoring
- Top-*N* recommendation

RSA: Candidate Generation

Since there are so many items, we choose a smaller subset of candidate items depending on the context.

Examples:

- Candidates for user u might be her un-rated items, i.e. items i for which $r_{ui} = null$. This has two potential problems. What are they?
- For "linear TV", candidates might be programmes that are being broadcast this evening.
- For movie-going, candidates might be films that are being screened at the user's multiplex this week.
- For online news, candidates will be recent stories.
- For on-the-go travel (e.g. restaurants, hotels and points-of-interest), candidates must be nearby and open.

RSA: Scoring

Each candidate item is scored, e.g. for how relevant it is to this user, allowing the candidates to be ranked in order of decreasing score.

There are many ways of doing this.

- ***Collaborative methods*** recommend items that either users with similar tastes liked in the past or that, according to the other users, are similar to items that are liked by the active user.
- ***Content-based methods*** recommend items which, according to the item descriptions, are similar to items that are liked by the active user.
- ***Hybrid methods*** combine collaborative and content-based methods.

RSA: Top-N recommendation

The last step is to select the N candidates whose scores are highest and recommend these to the user.

Additional criteria to take into account at this stage -

- ***Business rules:*** e.g., there may be some items the business is trying to push (e.g. think about sponsored content).
- ***Ensemble/Hybrid recommendations***: combine scores of more than one recommender model
- ***Re-ranking:*** to ensure the degree of *diversity* or some notion of *fairness*.

Re-ranking Recommendations

Recommendation as a Ranked List

- In general, a ranking function is learnt from the labeled dataset to optimize the global performance, which produces a ranking score for each individual item.

This may be sub-optimal !!!

- The scoring function applies to each item individually -
  - It does not explicitly consider the mutual influence between items,
  - Also, the di□erences of users' preferences or intents.

Re-ranking Recommendations

Re-ranking task:

- **Input**: A set of recommendations (***RS***) for user ***u***, each item in the set with relevance score
- **Output**: A ranked list (***RL***) containing all items in ***RS*** that attempts to satisfy some notion of interest (e.g. diversity)

Re-ranking Recommendations

Recommendation Diversification:

- Recommendation diversification aims to determine an optimal recommendation set of size N items, denoted here by RL^*
- Commonly formulated as a linear combination of
  - the relevance of the items in the recommendation set, and
  - the diversity of that set,

the trade-off between the two being controlled by a parameter λ (0 ≤ λ ≤ 1).

$$RL^* = \underset{RL,|RL|=N}{\arg\max}\ (1-\lambda)s(RL) + \lambda\,\mathrm{div}(RL)$$

$$s(RL) = \sum_{i \in RL} s(u,i)$$

Re-ranking Recommendations

Recommendation Diversification:

- Recommendation diversification aims to determine an optimal recommendation set of size $\boldsymbol{N}$ items, denoted here by $\boldsymbol{RL}^*$
- Commonly formulated as a linear combination of
  - the relevance of the items in the recommendation set, and
  - the diversity of that set,

the trade-off between the two being controlled by a parameter λ (0 ≤ λ ≤ 1).

$$RL^* = \underset{RL, |RL|=N}{\arg\max}\,(1-\lambda)s(RL) + \lambda\,\mathrm{div}(RL)$$

$$s(RL) = \textstyle\sum_{i \in RL} s(u,i)$$

Finding the optimal recommendation set $\boldsymbol{RL}^*$ is NP-hard.

Re-ranking Recommendations

Greedy Re-ranking

$RL \leftarrow [\,]$
while $|RS| > 0$ **do**
    $i^* \leftarrow \arg\max_{i \in RS} f_{obj}(i, RL)$
    delete i^* from RS
    append i^* to the end of RL
return RL

$$f_{obj}(i, RL) = (1 - \lambda) s(u, i) + \lambda \operatorname{div}(i, RL)$$

In Maximal Marginal Relevance (MMR), ***`div(i,RL)`*** is the maximum of the distances between ***`i`*** and the items already selected

$$\operatorname{div}(i, RL) = \max_{j \in RL} \operatorname{dist}(i, j)$$

Re-ranking Recommendations

Greedy Re-ranking

1: $RL \leftarrow [\,]$
2: **while** $|RS| > 0$ **do**
3: $\quad i^* \leftarrow \arg\max_{i \in RS} f_{obj}(i, RL)$
4: $\quad$ delete i^* from RS
5: $\quad$ append i^* to the end of RL
6: **return** RL

$$f_{obj}(i, RL) = (1 - \lambda) s(u, i) + \lambda \operatorname{div}(i, RL)$$

In Maximal Marginal Relevance (MMR), ***div(i,RL)*** is the maximum of the distances between ***i*** and the items already selected

$$\operatorname{div}(i, RL) = \max_{j \in RL} \operatorname{dist}(i, j)$$

Does it ensures users' tastes or interests?

Next lecture -

Explaining Recommendations