# Lecture 19-22

- Re-ranking Techniques

IT492: Recommendation Systems (AY 2023/24) — **Dr. Arpit Rana**

# Recommendation 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 $\boldsymbol{u}$ might be her un-rated items, i.e. items $\boldsymbol{i}$ for which $\boldsymbol{r_{ui}}$**=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 Recommendations

**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 differences of users' preferences or intents.

# Re-ranking Recommendations: The Task

Given a recommendation algorithm whose scoring function $s : U \; x \; I \rightarrow R$ determines the ranking of a recommendation, we propose applying a re-scoring by means of a linear combination between this scoring function and the novelty (diversity) of the items:

$$s_{div}(i) = (1 - \lambda)\; s(u,i) + \lambda\; div(i)$$

Here, λ is the parameter that controls the trade-off between the original (relevance-oriented) score and the diversity. The scores **s(u,i)** and **div(i)** need to be normalized.

# Re-ranking Recommendations: The 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, explanation)

$$s_{div}(i \mid RL) = (1 - \lambda)\, s(u,i) + \lambda\, div(i \mid RL)$$

# Re-ranking Recommendations for 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).

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

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

Finding the optimal recommendation set $RL^*$ is NP-hard.

# Greedy Re-ranking for Recommendations Diversification

$RL \leftarrow [\,]$
**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** $RL$

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

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

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

Does it ensures users' tastes or interests?

- Intent-Aware Diversification for IR to RecSys

# Intent-Aware Diversification

We study the problem of answering ambiguous web queries under the following assumptions –

- there exists a taxonomy of information,
- user intents are modeled at the topical level of this taxonomy.
- that both queries and documents may belong to more than one category of this taxonomy, and
- usage statistics have been collected on the distribution of user intents over the categories.

## Intent-Aware Diversification

**Preliminaries**

- C(q): the set of categories to which a query q belongs,
  C(d): and that for a document d, and
  C(d)∩ C(q) may be empty.

- P(c|q): the probability of a given query belonging to given categories (assuming this distribution is known), and that our knowledge is complete, i.e.,

$$\sum_{c \in C(q)} P(c|q) = 1.$$

# Intent-Aware Diversification

**Preliminaries**

- $V(d \mid q, c)$ denote the quality value (in the range of [0, 1]) of a document d for query q when the intended category is c, meant to capture the relevance of the document.

- **Independence assumption**

  Suppose two documents $d_1$ and $d_2$ are the results for a query q belonging to category c, the probability that the user will find none of the documents useful equals:

$$(1 - V(d_1 \mid q, c))\,(1 - V(d_2 \mid q, c)).$$

# Intent-Aware Diversification

**Basic Idea**

- A simple scheme that proportionately allocates the number of results to show for each category according to the percentage of users interested in that category may perform poorly.

- The missing link in this simple scheme is that the attractiveness of adding another document decreases when we have already shown a high quality document from the same category.

# Intent-Aware Diversification: Problem Formulation

Given query q, a set of documents D, a probability distribution of categories for the query P(c|q), the quality values of the documents V (d | q, c), and an integer k. Find a set of documents S ⊆ D with |S| = k that maximizes –

$$\underbrace{P(S|q)}_{\substack{\text{the probability that the} \\ \underline{\text{set of documents S}} \\ \text{(order is not considered)} \\ \text{satisfies the “average”} \\ \text{user who issues query q.}}} = \overbrace{\sum_c P(c|q)}^{\substack{\text{summing up over} \\ \text{all categories,} \\ \text{weighted by P(c | q),}}} \underbrace{(1 - \overbrace{\prod_{d \in S} (1 - V(d|q,c))}^{\substack{\text{probability that the} \\ \text{set of documents} \\ \text{will all fail to satisfy}}})}_{\substack{\text{the probability that} \\ \text{some document will} \\ \text{satisfy category c.}}}$$

This is an
NP-hard problem.

# Intent-Aware Diversification: A Greedy Solver

**A Few More Terms**

- R(q): the top k documents selected by some classical ranking algorithm for the target query.
- U(c|q, S): the conditional probability that the query q belongs to category c, given that all documents in set S fail to satisfy the user.

Our algorithm will reorder R(q) to maximize the objective P(S|q).

# Intent-Aware Diversification: A Greedy Solver

- The algorithm selects output documents one at a time.
- At every step, it chooses the document that has the highest marginal utility,

$$g(d|q, c, S) = U(c|q, S) \cdot V(d|q, c)$$

- Marginal utility can be interpreted as the probability that the selected document satisfies the user given that all documents that come before it fail to do so.

# Intent-Aware Diversification: A Greedy Solver

**Input** $k, q, C(q), R(q), C(d), P(c|q), V(d|q,c)$
**Output** set of documents $S$

$S = \emptyset$
$\forall c, U(c|q,S) = P(c|q)$
**while** $|S| < k$ **do**
  **for** $d \in R(q)$ **do**
    $g(d|q,c,S) \leftarrow \sum_{c \in C(d)} U(c|q,S) V(d|q,c)$
  **end for**
  $d^* \leftarrow argmax \ g(d|q,c,S)$ [ties broken arbitrarily]
  $S \leftarrow S \cup \{d^*\}$
  $\forall c \in C(d^*), U(c|q,S) = (1 - V(d^*|q,c)) U(c|q, S \setminus \{d^*\})$
  $R(q) \leftarrow R(q) \setminus \{d^*\}$
**end while**
**return** S

# Intent-Aware Diversification: Submodularity

**P(S|q) is a submodular function.**

Given a finite ground set N, a set function f: $2^N \to R$ is submodular if and only if for all sets S, T $\subseteq$ N such that S $\subseteq$ T, and d $\in$ N \ T,

$$f(S + d) - f(S) \geq f(T + d) - f(T).$$

Intuition: the marginal benefit of adding a document to a larger collection is less than that of adding it to a smaller collection.

# Intent-Aware Diversification: Submodularity

- For a submodular set function f, let S* be the optimal set of k elements that maximizes f.

- Let S’ be the k-element set constructed by greedily selecting element one at a time that gives the largest marginal increase to f. Then,

$$f(S') \geq (1 - 1/e) f(S^*).$$

Since our objective, P(S|q), is submodular, and our algorithm is greedy in the same sense, therefore, it is a (1 − 1/e)-approximation algorithm for diversifying top-k.

# Intent-Aware Diversification: Problem Formulation

Given a query q for which a list of documents R is retrieved, the Intent-Aware framework considers a set of subtopics s — possible interpretations or facets of the query — and calculates the marginal relevance of the result list R for each of them:

$$\text{M-IA}(R_q) = \sum_{s} p(s \mid q)\, m(R_q \mid s)$$

where m(R | s) is the relevance of the result list R with respect to subtopic s.

# Intent-Aware Diversification: Problem Formulation

A usual choice for the relevance metric m is the Expected Reciprocal Rank (ERR) of Chapelle et al. (2009), which is its intent-aware variant ERR-IA

$$
\begin{aligned}
\text{ERR-IA}(R_q) &= \sum_{s} p(s \mid q) \sum_{k=1}^{|R_q|} \frac{1}{k} p(\mathrm{rel} \mid d_k, s) \prod_{j=1}^{k-1} (1 - p(\mathrm{rel} \mid d_j, s)) \\
&= \sum_{d \in R_q} \frac{1}{k_d} \sum_{s} p(s \mid q)\, p(\mathrm{rel} \mid d, s) \prod_{d' \,:\, k_{d'} < k_d} (1 - p(\mathrm{rel} \mid d', s))
\end{aligned}
$$

where $d_k$ is the document ranked at position k in $R_q$ and p(rel | $d_k$, s) is the probability of relevance of document $d_k$ with respect to subtopic s.

# Intent-Aware Diversification: Problem Formulation

The adaptation of the intent-aware version of the Expected Reciprocal Rank (ERR-IA) for recommendation tasks can be expressed as:

$$\mathrm{ERR-IA}(R) = \sum_{a \in \mathcal{A}} p(a|u) \sum_{i \in R} \frac{1}{k_i} p(rel|i,u,a) \prod_{j\,:\,k_j < k_i} (1 - p(rel|j,u,a))$$

where $k_i$ is the position of item i in the recommendation list R and p(rel | i, u, a) is the probability that the user u finds the recommended item i relevant when interested in aspect a.

## Next Lecture

- Re-ranking Techniques: Learning to Rank