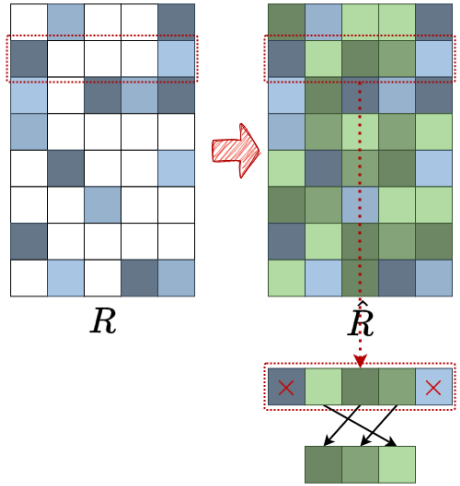
Personalized Machine Learning

Evaluation of Recommenders

Rodrigo Alves

November 5, 2024.

Basic Recommendation Framework



Basic Recommendation Framework

How to **evaluate** the basic recommender framework? **Ideas**?

There are **three main** paradigms:

- Offline analytics
- Case study
- Online experiment

Offline Analytics

- The main objectives of offline analytics are to assess recommendation algorithms.
  - Used to filter out unsuitable algorithms and retain potential candidates.
- The dataset of users' behaviors should be **collected** in advance.
- For instance, the dataset should include the choices or ratings made by users on the items.
  - These datasets can be used to **simulate the interactions** between users and the recommender systems.

Offline Analytics

- Offline analytics in recommender systems is also a common practice in ML.
- The testing method typically involves cross-validation.
- The main advantage of offline analytics is that it **does not** require interaction from real users.
  - It is cost-effective and allows for quick testing and evaluation of different recommendation algorithms.
- However, there are also disadvantages,
  - such as being limited in evaluating **serendipity** or **novelty**;
  - It might **not reflect changes** in taste distribution.

User Study

- To conduct a user study, it is necessary to **recruit testers** who will perform tasks using the recommender systems.
- The study must consider the distribution of testers, including demographic factors.
- When testers perform these tasks, the researcher must **observe** and **record** their behaviors, and gather information about the tasks, such as:
  - Which tasks were completed?
  - How much time was spent on each task?
  - The accuracy of the task results.
- Often, testers are also required to answer qualitative questions, including:
  - Their preferences regarding the user interface.
  - Their perception of the complexity of the tasks.
- **Qualitative data** are often crucial for explaining the quantitative results.

User Study

- Observe that these results **cannot** be obtained in offline analytics
- User study is an important method for evaluating recommender systems.
- However, user study also has some weaknesses.
  - First, the cost of user study is very high.
  - A large number of testers must be recruited.
  - Finally, testers must finish a large number of interacting tasks.
- **Trade-off:** cost $\times$ guaranteed in statistical significance $\times$ quality of data

Online Experiment

- An online experiment involves conducting large-scale testing on a **deployed** recommender system.
- It allows for the evaluation and comparison of different recommender systems based on real tasks performed by actual users.
- Online experiments provide the most realistic testing results among the three evaluation methods.
- These experiments can assess the overall performance of recommender systems, including long-term business profit and user retention.
- Instead of relying on cross-validation, the balance of these metrics is considered when selecting the parameters for recommendation algorithms.

Online Experiment

- In many real-world recommendation scenarios, system designers aim to influence user behavior using recommender systems.
- Therefore, when users interact with recommender systems employing different algorithms, designers seek to evaluate the impact of these algorithms on user behavior through online experiments.

Example: if a user clicks only **one item** when the system based on **algorithm A** recommends five items, while the user clicks **four items** when the system based on **algorithm B** also recommends five items, we consider the recommender system based on algorithm B to be *more effective* than the one based on algorithm A.

Online Experiment

- The researcher must ensure **random sampling of users** to maintain a balanced distribution across different recommender systems.
- Additionally, other influencing factors should remain consistent, such as maintaining a uniform user interface (UI) across all recommenders.
- Risks associated with online evaluation:
  - Recommenders **may suggest unrelated** items during the online experiment.
  - Ethical concerns may arise.
- Online experiments are often conducted as the final step among the three types of evaluation methods.

Best Practices in Recommender Evaluation

First, **offline analytics** are used to evaluate and compare various recommender algorithms.

Subsequently, in the **user study** phase, different users' interactions with the systems are recorded.

Finally, the most suitable recommender system is selected through an **online experiment**.

Machine Learning Perspective

- Various machine learning algorithms have been employed for the prediction of user ratings.
- Consequently, the assessment of recommender systems naturally extends to the domain of **prediction accuracy** in machine learning.
- Prediction accuracy, fundamentally, concerns the precision of predictions.
- When used in the context of evaluating recommender systems, this metric primarily assesses the system's ability to anticipate user behavior.

Machine Learning Perspective

- To calculate prediction accuracy, an offline dataset is required, containing user scores, such as ratings for items.
- This dataset is typically divided into a training set and a testing set.
  - It's essential to use the same dataset for evaluating different algorithms.
- A model for predicting user ratings is trained on the training set, and then predictions are made on the testing set.
- The error is the difference between the predicted rating and the actual rating.

Machine Learning Perspective

Machine Learning Metrics

Let R_Ω be a partially observed matrix where we observe the entries that are contained in the set Ω. Let Ω_{train} and Ω_{test} be the entries of the training and test sets, respectively, where $\Omega = \Omega_{\text{train}} \cup \Omega_{\text{test}}$. The main metrics computed on the test set from the machine learning perspective are:

- **Mean Absolute Error**

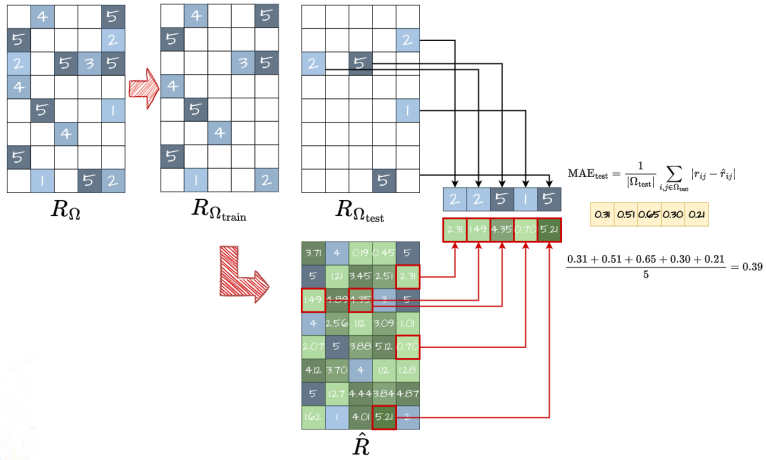
$$\text{MAE}_{\text{test}} = \frac{1}{|\Omega_{\text{test}}|} \sum_{i,j \in \Omega_{\text{test}}} |r_{ij} - \hat{r}_{ij}|$$

- **Root-Mean-Squared Error**

$$\text{RMSE}_{\text{test}} = \sqrt{\frac{1}{|\Omega_{\text{test}}|} \sum_{i,j \in \Omega_{\text{test}}} (r_{ij} - \hat{r}_{ij})^2}$$

where $|\Omega_{\text{test}}|$ is the size of the test set.

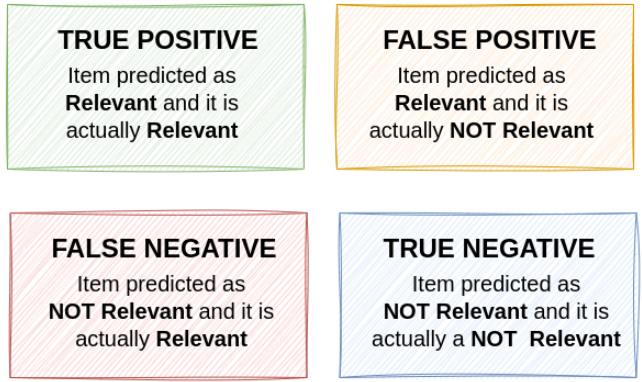
Machine Learning Perspective



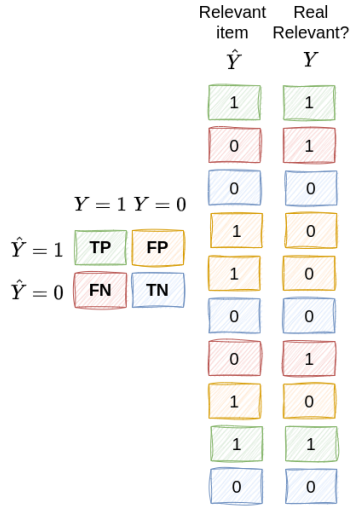
Information Retrieval Perspective

- Recommender systems are considered a specialized subset of **information retrieval** systems.
- Frequently, users are not concerned with fine numerical accuracy;
  - e.g., know whether an item rating is 3.4 or 3.5 is not that relevant.
  - Moreover, implicit feedback datasets do not encompass explicit ratings.
- Their primary interest often lies in selecting the **best available** item.
- Consequently, it is essential to explore metrics related to decision support and ranking, focusing on the **order of recommended items** that align with users' preferences.

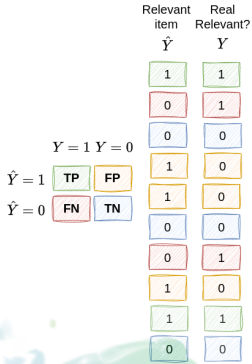
Confusion Matrix



Confusion Matrix



Confusion Matrix



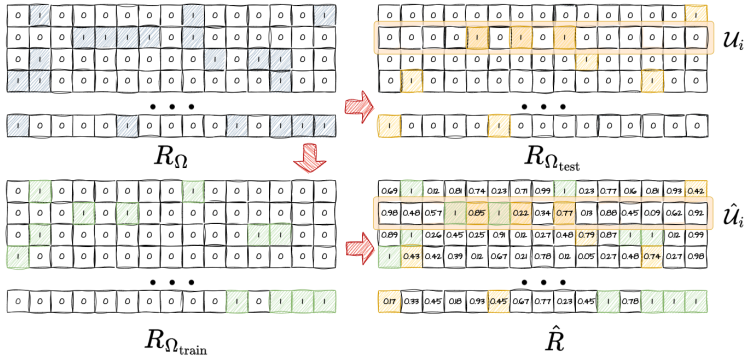
- TP $= 2$, FP $= 3$, FN $= 2$, and TN $= 3$ / $N = 10$
- $N_+ =$ # of $(Y = 1) = TP + FN = 4$ / $N_- =$ # of $(Y = 0) = TN + FP = 6$
- $\hat{N}_+ =$ # of $(\hat{Y} = 1) = TP + FP = 5$ / $\hat{N}_- =$ # of $(\hat{Y} = 0) = TN + FN = 5$
- **Recall (REC) or True Positive Rate(TPR):** REC $=$ TPR $=$ TP$/N_+$
- **False alarm or False Positive Rate(FPR):** FPR $=$ FP$/N_-$
- **Selectivity or True Negative Rate(TNR):** TNR $=$ TN$/N_-$
- **Miss rate or False Negative Rate(FNR):** FNR $=$ FN$/N_+$
- **Accuracy (ACC):** ACC $=$ (TP+TN)$/N$
- **Precision (PR):** PR $=$ TP$/\hat{N}_+$
- **F1-Score (F_1):** $F_1 = \frac{2}{1/PR+1/REC}$

Exercise: compute the rates for our example!

Information Retrieval Perspective

- In offline analytics, it is often challenging to discern relevant from non-relevant items for a user.
  - It's worth noting that, even in online experiments, computing such metrics is not always feasible (or meaningful).
- To address this challenge, we use approximations.
- The key metrics in the confusion matrix used in Recommender Systems are **Recall** and **Precision**. The **F1-score** is also occasionally computed.
- We often restrict it to the top-k recommendation.

Information Retrieval Perspective



Precision and Recall in Recommenders

Precision@K and Recall@K

- **Precision@K** measures the accuracy of the top K recommendations:

$$\text{Precision@K} = \frac{\text{TP in the Top K}}{K}$$

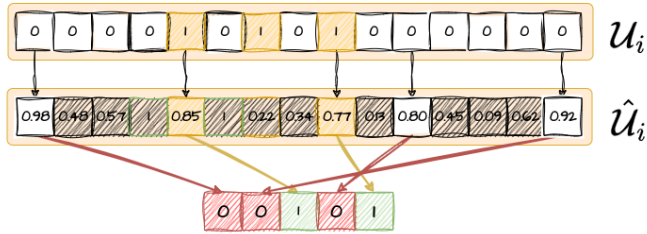
- **Recall@K** measures the ability to capture relevant items within the top K recommendations:

$$\text{Recall@K} = \frac{\text{TP in the Top K}}{\text{Total Number of Relevant Items in the Dataset}}$$

We normally output the **average** of Recall or Precision of the users in the test set.

Recall@k

$K = 5$



$$\text{Recall}_{u_i} = \frac{\text{TP at K}}{\text{Total of Relevants}} = \frac{2}{3} = 0.66$$

Normalized Discounted Cumulative Gain

- Observe that the order does not affect precision and recall.
- NDCG is a widely used evaluation metric in information retrieval and recommendation systems.
- It assesses the **quality of ranked lists** or **recommendations** by considering both relevance and ranking position.
- NDCG@k values range between 0 and 1, with 1 indicating a perfect ranking.
- It's a valuable metric for assessing recommendation system performance and the quality of search engine results.

Normalized Discounted Cumulative Gain

NDCG Formula

$$\text{NDCG@}k = \frac{\text{DCG@}k}{\text{IDCG@}k}$$

- DCG (Discounted Cumulative Gain) measures the cumulative gain of the top k items, with decreasing contributions for items lower in the list.

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)}.$$

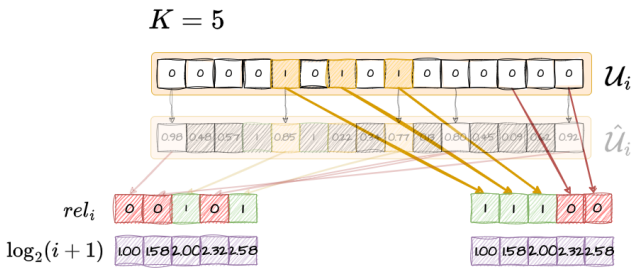
- IDCG (Ideal DCG) represents the ideal cumulative gain if all the top K items were perfectly relevant.

$$\text{IDCG@}k = \sum_{i=1}^{k} \frac{rel'_i}{\log_2(i+1)},$$

where rel'_i represents the relevance of the ideal (perfect) ranking at position i.

NDCG@k

$K = 5$



$$\text{DCG@}k = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)} = \frac{0}{1} + \frac{0}{1.58} + \frac{1}{2} + \frac{0}{2.32} + \frac{1}{2.58} = 0.88$$

$$\text{IDCG@}k = \sum_{i=1}^{k} \frac{rel'_i}{\log_2(i+1)} = \frac{1}{1} + \frac{1}{1.58} + \frac{1}{2} + \frac{0}{2.32} + \frac{0}{2.58} = 2.13$$

$$\text{NDCG@}k = \frac{\text{DCG@}k}{\text{IDCG@}k} = \frac{0.88}{2.13} = 0.41$$

Coverage

- Item coverage represents **the proportion** of recommended items out of all available items.
- It indicates the system's ability to encompass a broad range of recommended items.
- A robust recommendation system should aim for both **high prediction accuracy** and **comprehensive coverage**.

Coverage Formula

Let $\mathcal{I}$ denote the set of all items, and $|A|$ represent the size of set A. Suppose $\mathcal{I}^{(i)}$ represents the set of recommended items for user i. The coverage C of the recommender system is defined as:

$$C = \frac{|\bigcup_i \mathcal{I}^{(i)}|}{|\mathcal{I}|}$$

Perspective of Human Interaction

- Can we consider a system perfect when **NDCG** $=$ **Recall** $= C = 1$?
- Not necessarily. RSs operate in dynamic environments.

Imagine a user who just purchased a new smartphone from an online store. While recommending highly relevant smartphone models could be accurate, the user may not be interested at that moment. Recommending accessories for the newly acquired smartphone might be more timely and relevant to the user's needs.

- **Diversity**, **Trust**, **Serendipity**, and **Novelty** are metrics used to evaluate recommender systems from the perspective of human interaction.

Diversity

- Diversity in recommender systems is the **opposite of similarity**.
- Recommending similar items in all situations may not make practical sense(e.g., the smartphone example from previous slide).
- When designing a recommender system, it's essential to consider not only prediction accuracy but also the diversity of recommended products to meet the varying needs of users.
- Moreover, diversity can help mitigate ethical concerns, such as the formation of filter bubbles.

Trust

- Trust in recommender systems refers to the level of trust users have in the system's recommendations.
- When a user is presented with recommended items they are familiar with and appreciate, they are more likely to trust the system.
- Trust can be established through user surveys, assessing users' trust in the recommendations.
- **Transparent** and **reasonable** explanations enhance user trust in the recommender system.

Novelty

- Novelty in a recommender system entails recommending items that users are not familiar with.
- Filtering items that users have already bought or rated is a straightforward way to promote novelty, but it may not be foolproof.
- Another approach is to consider the popularity of recommended items, with less popular items contributing to a sense of novelty.
- Novelty can be assessed through user surveys to ensure recommendations are introducing users to new and unexplored items.

Serendipity

- Serendipity in a recommender system embodies the quality of surprise and unpredictability in recommendations.
- For instance, while recommending early films of a favorite actor can introduce novelty, true serendipity involves delighting users with recommendations they wouldn't have expected.

In summary, **novelty** focuses on introducing **new and unfamiliar content**, while **serendipity** adds the extra layer of providing **pleasant surprises**.

- Evaluating serendipity often includes measuring the similarity between recommended items and a user's previous preferences and assessing user satisfaction with the recommendations.

Perspective of Software Engineering

- Various points of software engineering are involved in the dynamics of recommender systems.
- For instance, you may develop an accurate and diverse recommendation algorithm, but it could be too computationally expensive for real-time execution.
- Recommender systems are expected to be highly available, often functioning continuously.
- Evaluation dimensions from this perspective include **real-time performance**, **robustness**, and **scalability**.

Real-Time

- Real-time systems are essential for items like news and new arrivals.
- Recommending new arrivals promptly can attract web traffic and opportunities.
- Real-time recommender systems have two key aspects:
  - Real-time recommendations for newly added items.
  - Real-time recommendations based on user behavior (e.g., purchases or clicks).
- The Recommender can thus be evaluated based on the **response time** or the **time** it takes to **accurately recommend** a new item.

Robustness

- Robustness in a recommender system refers to its stability when faced with false information aimed at influencing recommendation results.
- Many recommender systems rely on user behavior data (e.g., purchases or clicks), making them vulnerable to attacks on the system.
- Attacks on recommender systems involve false changes in user behavior that can impact recommendation outcomes.

For instance, a hotel manager may register fake users to increase positive ratings and reviews for their hotel, while also giving negative feedback to competitors. Robustness, in this context, is the system's **ability to resist** such cheating behaviors.

- Detecting abnormal user behaviors can be achieved, for instance, by using of attack models or anomaly detection techniques.

Robustness

- Robustness also refers to the system's stability against **extreme events**, such as handling a massive influx of user requests within a short timeframe.
- This type of robustness is closely related to the infrastructure and architecture of the recommender system, including hardware scale and reliability, database software, etc.
- Recommendation algorithms designed for parallel computing tend to exhibit higher robustness in such scenarios.

Scalability and Resource Management

- Scalability is a critical consideration in the design of recommending systems.
- Resource consumption should be carefully monitored during system operation.
- When a recommending system is newly launched, its database, including the number of items and users, is typically small and may contain simulated data.

For example, **Algorithm A** may outperform **Algorithm B** in prediction accuracy when the database is small. However, as the database expands, **Algorithm A**'s prediction accuracy approaches that of **Algorithm B**, but its execution speed may become significantly slower.

- This complexity makes it challenging to definitively determine which algorithm is superior.

Business Perspective

- Business success is a fundamental goal of recommender systems, and evaluation plays a **pivotal role** in achieving it.
- Key business-oriented evaluation metrics include:
  - **Conversion Rate:** Measures the percentage of users who acted on recommendations, such as making a purchase or subscribing.
  - **Revenue Generation:** Tracks the direct impact of recommendations on sales and revenue.
  - **Customer Lifetime Value (CLV):** Assesses the long-term value of users acquired through recommendations.
  - **Retention Rate:** Measures how well recommendations retain and engage users over time.
- Beyond metrics, the alignment of recommendations with business objectives, user engagement, and customer satisfaction are critical aspects of evaluation.

Obrigado :) - Faculty of Information Technology