Recommender system using active learning and recursive algorithm in collaborative filtering

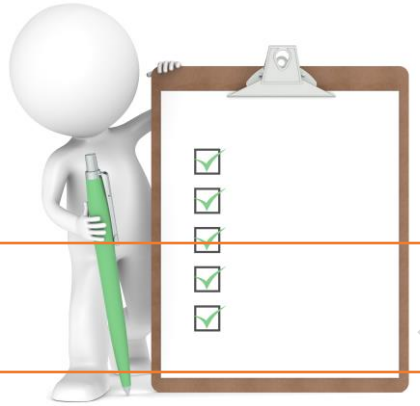
12 May 2022

Haithem BEN DRISSI

M2 MODO

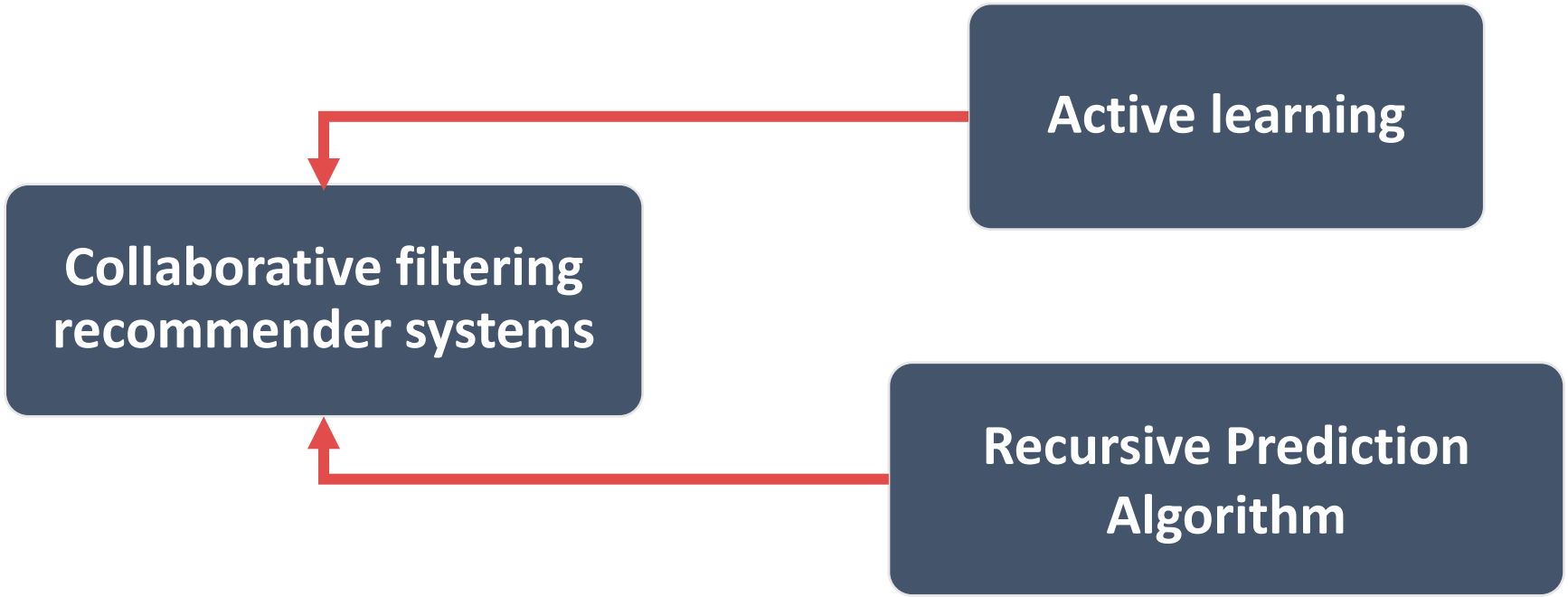
2021 - 2022

Outline



1. Introduction

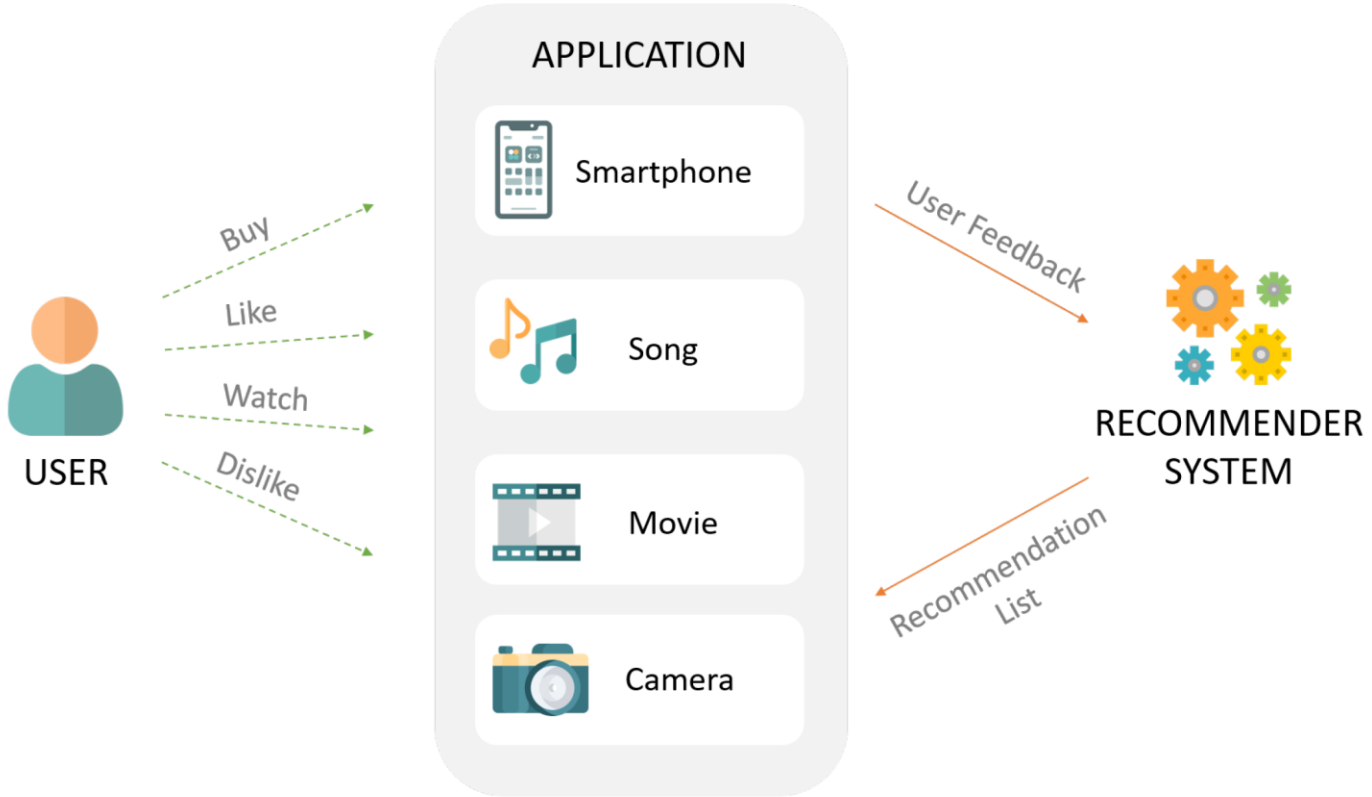
❖ **Survey :**



❖ **Implementation of a two-layer recommender system**

2. Collaborative filtring

Collaborative filtering (CF) recommender systems uses item ratings provided by a collection of users. It recommends items that the target user has not considered but may like.



2. Collaborative filtring

Collaborative filtering systems compute recommendations by exploiting relationships and similarities between users and items.

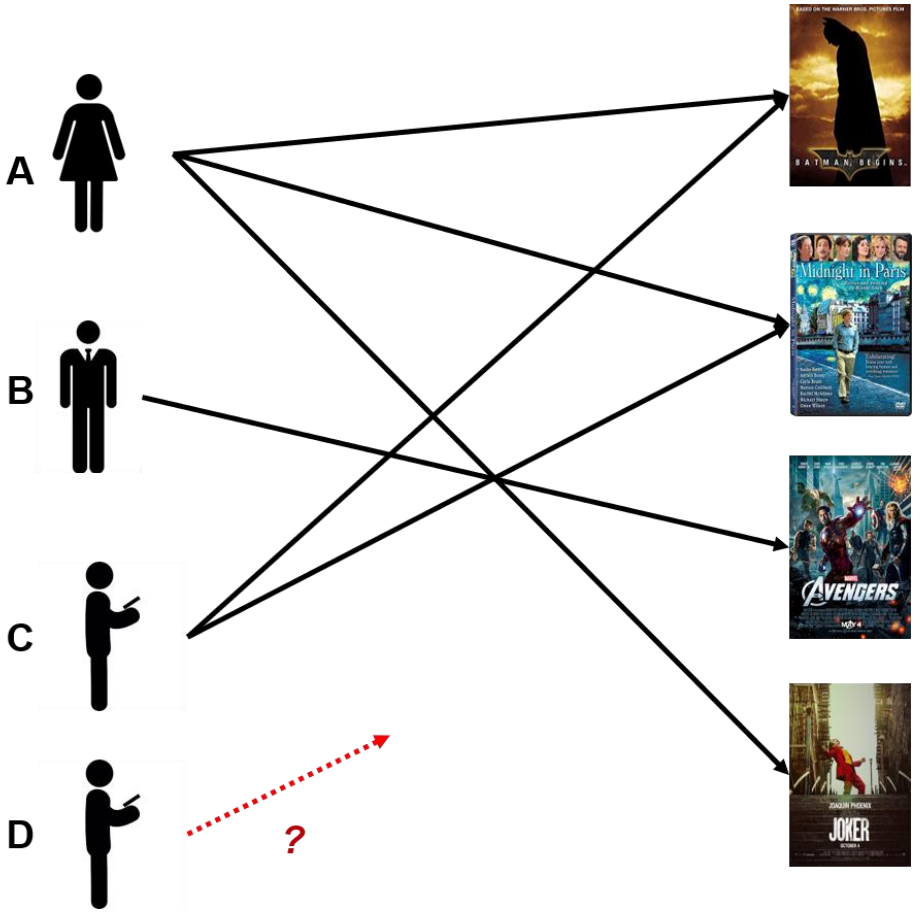
2.1. Neighborhood-based model

- **Baseline strategy (BS):** Selects the top K nearest-neighbors who have rated a given item.
- **Baseline strategy with overlap threshold (BS+):** Top K nearest-neighbors who have rated the given item and having rated at least φ items in common with the active user.
- **Similarity strategy (SS):** Top K nearest neighbors purely according to their similarity with the active user.
- **Combination strategy (CS):** Combines (BS) and (SS).
- **Combination strategy with overlap threshold (CS+):** Combines (BS+) and (SS)

2. Collaborative filtring

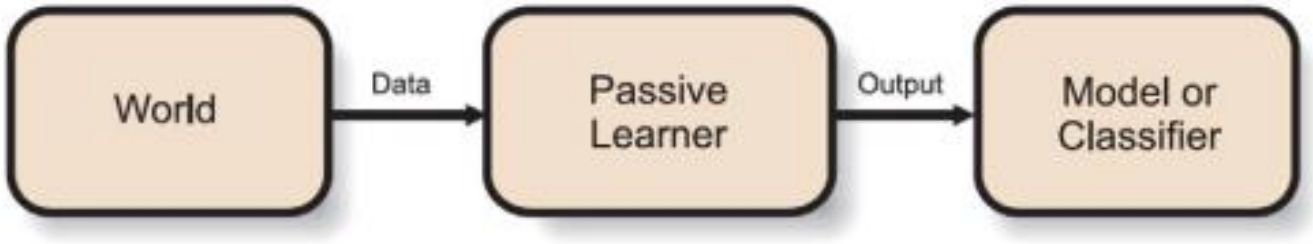
2.2. Cold start problem in collaborative filtering

A major problem of collaborative filtering is the **cold start**, i.e., when the system has not yet acquired enough information (ratings), it can not generate reliable recommendations for a new user.



3. Active Learning

- Acquiring the training data in this way is expensive, time consuming, and may not be useful to improve the system performance.



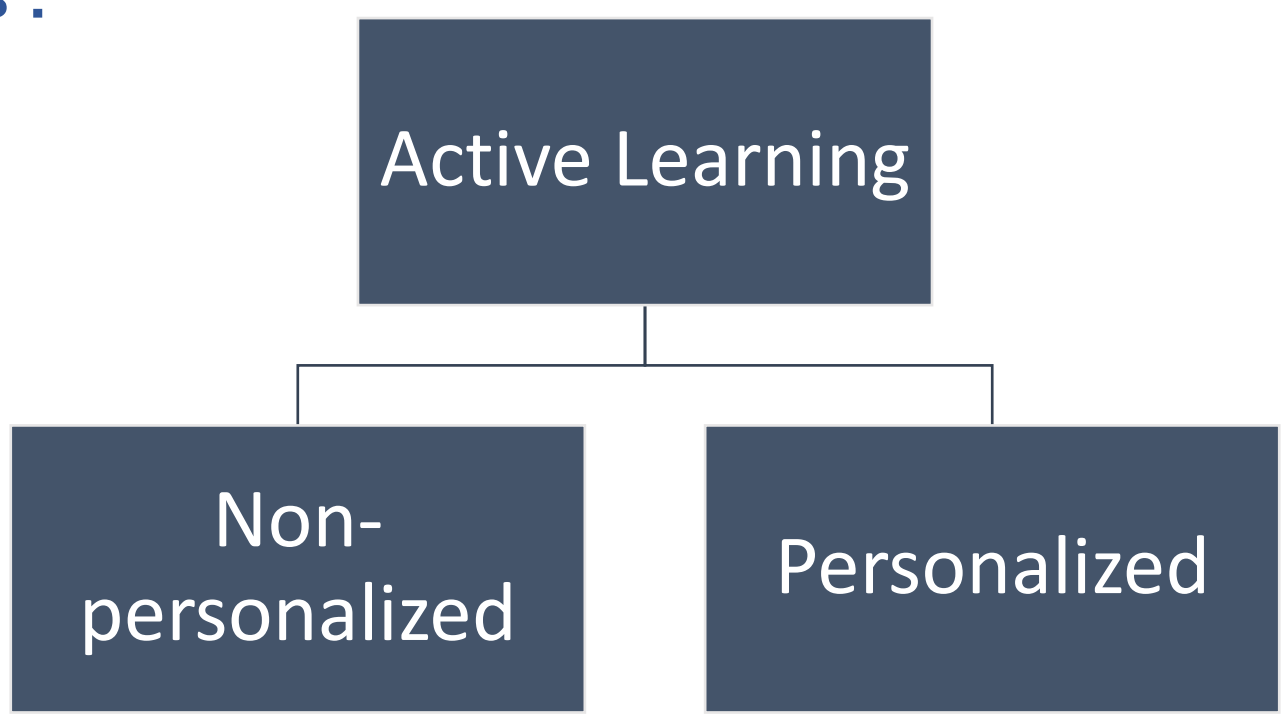
General schema for a passive learner.

- Thanks to the **interaction with the user**, the number of data points used for training a model based on **active learning** is lower and has shown promising results in dealing with the cold start problem.

General schema for an active learner.

3. Active Learning

❖ **Strategies :**



Don't consider past user ratings and instead ask everyone to review the identical items.

3. Active Learning

❖ Examples of non-personalized active learning

Variance

This strategy favors the items that have been rated diversely by the users.

Representative-based

Tries to identify a set of items that best represent the entire catalogue.

Random-Popularity

Select randomly a list of movies and extend it with popular movies to improve the probability that the user can rate them.

3. Active Learning

❖ Examples of personalized active learning

Item-Item

selects the items with the highest similarity to the user's previously rated items.

Binary Prediction

Tries to identify the items more likely experienced by the user using a prediction model.

4. Recursive Prediction Algorithm for collaborative filtering

- The goal of the recursive prediction algorithm is to include nearest neighbors who haven't rated the given item in the prediction process.

- When the process requires a rating value that doesn't exists in the dataset, we can estimate it recursively on the fly, and then use it in the recommendation process.

❖ Prediction Computation

$$r_{x,i} = \bar{r}_x + \frac{\sum_{y \in U_x} (r_{y,i} - \bar{r}_y) sim(x,y)}{\sum_{y \in U_x} |sim(x,y)|}$$

$$with: \qquad sim(u,v) = \bar{r}_u + \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

5. Implementation of the recommender system

1. Preprocessing data
   Output: Structured data ready to use

2. Active Learning implementation (Variance Strategy)
   Output: Movies suggested to the active user for rating

3. Selecting Neighbors process based on similarity with active user
   Output: K-nearest neighbor to the active user

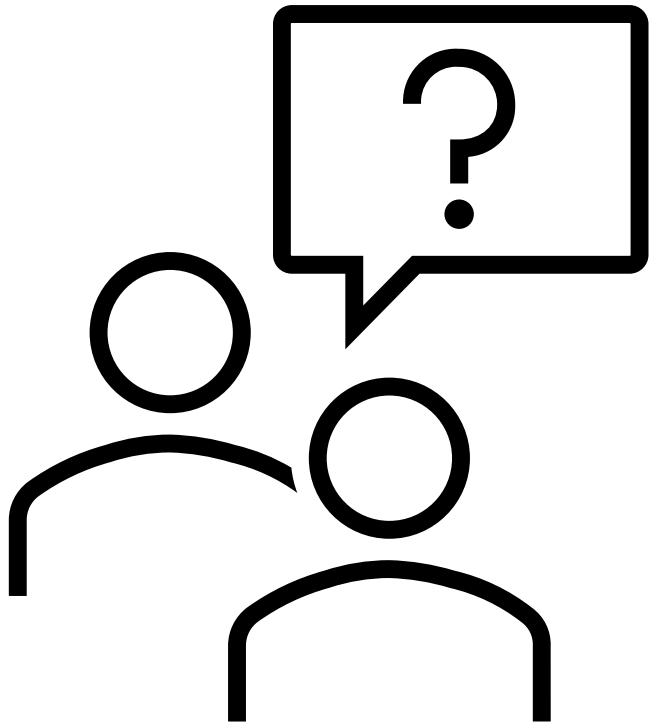
4. Recursive Prediction Algorithm implementation
   Output: Prediction of missing rating of neighbors

5. Recommendation system based on active learning strategy and Recursive Prediction Algorithm
   Output: Recommendation for the active user.

5. Implementation of the recommender system

Conclusion and future work

- ✓ We present the state of art relative to collaborative filtering, active learning and recursive algorithm used in recommender systems.
- ✓ We introduce and implement an approach based on two layers (active learning and recursive algorithm) to get more accurate recommendations of movies for new users while the system did not acquire enough data on them.
- ✓ Future works can conduct analysis of this approach using different active learning strategies and recursive algorithm parameters.



Thank you for your attention