Personalized Machine Learning

New trends in PML

Rodrigo Alves

November 20,2025

Matrix Factorization

- Traditional matrix factorization methods rely on user-item interactions.
- Matrix factorization decomposes the user-item interaction matrix into latent factors using the following optimization objective:

$$\operatorname{argmin}_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Omega} (r_{i,j} - u_i^T v_j)^2 + \lambda\Big(\sum_x ||u_x||^2 + \sum_y ||v_y||^2\Big).$$

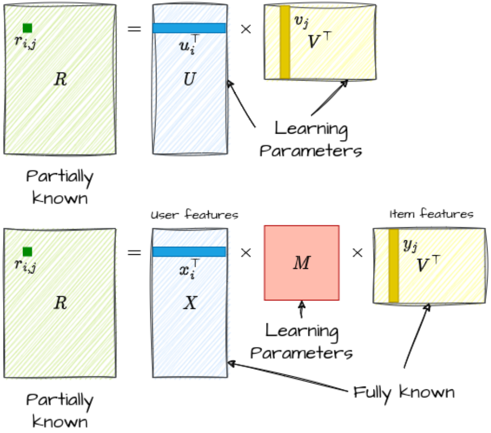
- The goal is to minimize the difference between observed ratings $r_{i,j}$ and the predicted ratings $\hat{r}_{i,j} = u_i^\top v_j$ for user i and item j.
- MF is a very popular, efficient and accurate method.

Limitations: traditional matrix factorization has challenges in adapting to **new users or items**, especially in dynamic environments.

Inductive Matrix Factorization

- Inductive Matrix Factorization extends these methods to handle new users and items not present during training.
- Incorporates side information about users and items (e.g., user demographics, item characteristics).
- Enables the model to generalize to new users and items by leveraging auxiliary data during training.
- Applications: particularly beneficial in **dynamic environments** and emerging markets.
- Because it uses additional information, IMC also can enhance the model's accuracy for **warm-start** users.

MF *vs* IMF

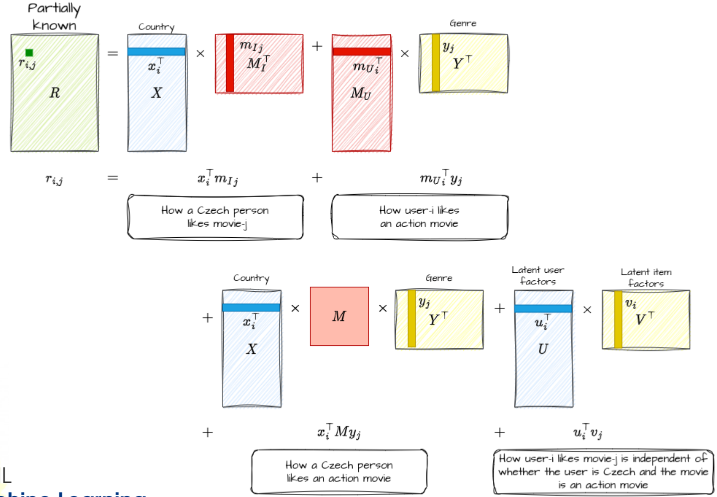

$r_{i,j}$
R
=
$u_i^\top$
U
×
v_j
$V^\top$
Learning Parameters
Partially known
User features
Item features
$r_{i,j}$
R
=
$x_i^\top$
X
×
M
×
y_j
$V^\top$
Learning Parameters
Fully known
Partially known

Hybrid MF and IMF Model

- MF and IMF are not disconnected but complementary.
- In machine learning in general, features are often an abstraction of reality.
- For example, in real estate price prediction, the price p_i model would be the area a_i of the property times a learning parameter ω_i.
- If you don't have any information, it will be safely better than predictic the average since bigger properties are generally more expensive.
- We know that not only the area, but also many other variables might influence the price of the flat. However, no matter how comprehensive the set of variables is, it would still be incomplete to fully predict.

Hybrid **MF** and **IMF** model: **IMF** is responsible for extracting the **contribution of side information** to the predictor, while **MF** works in the **side-information-free** part.

MF + IMF



$$r_{i,j} = x_i^\top m_{Ij} + m_{U_i}^\top y_j$$

How a Czech person likes movie-j

How user-i likes an action movie

$$+ \; x_i^\top M y_j + u_i^\top v_j$$

How a Czech person likes an action movie

How user-i likes movie-j is independent of whether the user is Czech and the movie is an action movie

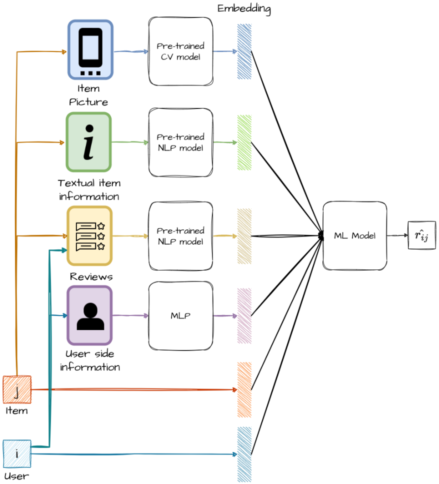
Multimodal Recommender Systems

- Traditional recommender systems rely on user-item interactions.
- IMF, for example goes beyond and also deal with side-information.
- Multimodal recommender systems extend this paradigm by incorporating diverse types of information.
- Modalities include text, images, audio, and more, providing a richer understanding of user preferences.

Why Multimodal?

- **Enhanced user understanding:** combining multiple modalities offers a more comprehensive view of user preferences.
- **Addressing cold start:** useful for new users or items with limited interaction history.
- **Improved recommendation accuracy:** leveraging diverse data sources helps capture nuanced user interests.
- **Real-world applications:** multimodal systems excel in domains like fashion, art, and multimedia content.

Multimodal Recommender Systems



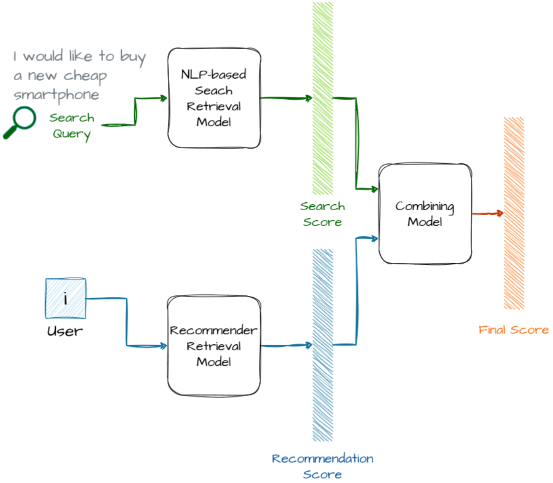
Personalized Search

- Traditional search engines provide **one-size-fits-all** results.
- Personalized search **tailors** search results by gathering information about user preferences, behavior, and history.
- It enhances the user experience by delivering more relevant and targeted information.
- It adapts the search algorithm to continuously improve and adapt to changing user preferences.

Personalized Search

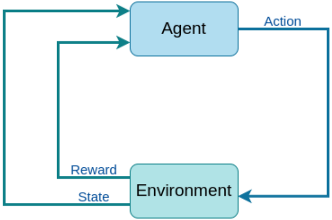
- Benefits:
  - ✓ **Improved relevance:** users receive results tailored to their interests.
  - ✓ **Enhanced user satisfaction:** personalization leads to a more satisfying search experience.
- Challenges:
  - × **Privacy concerns:** Balancing personalization with user privacy.
  - × **Serendipity:** balance between relevance and introducing users to new content.
  - × **Cold start problem:** challenges happens when there is limited user history.

Personalized Search


I would like to buy a new cheap smartphone
Search Query
NLP-based Seach Retrieval Model
Search Score
Combining Model
Final Score
i
User
Recommender Retrieval Model
Recommendation Score

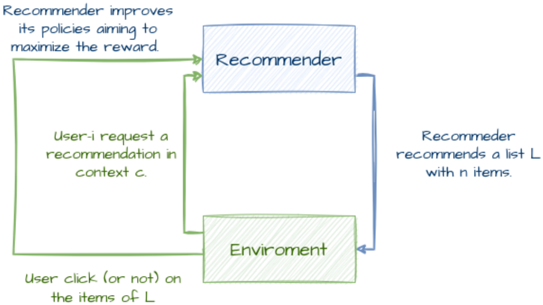
Reinforcement Learning

- Area of machine learning concerned with how intelligent **agents** take **actions**.
- Basic setup:



- RL is very popular in recommender systems since the system essentially represents **an agent** in a dynamic environment of users and items.

RL’s Recommendation Setup



Recommender improves its policies aiming to maximize the reward.

Recommender

User-i request a recommendation in context c.

Recommeder recommends a list L with n items.

Enviroment

User click (or not) on the items of L

Reinforcement Learning

- The purpose of RL is to create ways for the agent to learn **optimal policies** to take **actions** that optimize the reward.
- Based on the current stage and the previous rewards, the **agent** chooses the next **action**.
- The environment could be deterministic. However, often it is stochastic with a complicated reward distribution.
- Often, it is also not just stochastic, but the reward distribution can change over time.

Suppose you live in Prague. You (**the agent**) want to commute for the next 30 days from home to university. You have three options: **(A)** Walking; **(B)** By public transportation; and **(C)** by Bike. Each day you can arrive on time (**reward** $= 1$) or be delayed (**reward** $= 0$). Suppose you do not have an idea which of the three methods is the more convenient one to **optimize the reward** during these 30 days. Note that you have uncertainty even if you choose a single option. **(1)** How would you interact with your environment aiming to arrive the **maximum number of days** on time at university? **(2)** How would traditional ML deal with this problem? **(3)** Does the best option exist?

Exploration $\times$ Exploitation Dilemma

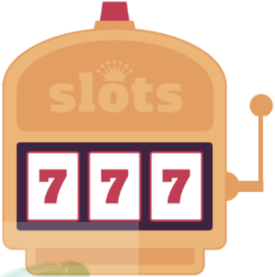
- If we do not know anything about the environment, we need to **explore** it.
- For example, we could go by public transport the first day, walking the second, and biking the third.
- Suppose that we arrive by bike **on time** while the other options do not. Do we have enough evidence to go with just the bike?
- After 15 trials, we have $\{P, W, B\} = \{4, 2, 3\}$. What would you say now?
- When should we start to **exploit** our knowledge?

Exploration $\times$ Exploitation Dilemma

We frequently face the exploration versus exploitation trade-off when deciding between options. Should you choose what you know and expect similar results ('**exploit**'), or should you choose something you're not sure about and possibly learn more ('**explore**'), and thus expect better results?

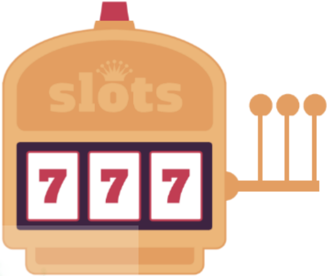
- After 27 trials, we have $\{P, W, B\} = \{8, 3, 4\}$. We would be very confident now.
- A new metro line, with a station close to your place, is constructed.
- Would it affect the **reward distribution**?
- Reinforcement Learning frequently deals with such questions. So do recommenders.
  - Should I offer new items to users to discover more about their taste or offer the ones I am confident they would like?
  - A new iPhone is launched. Would it change the view distribution of an online mobile store?

One-Armed Bandit



- A one-armed bandit (or slot machine) is a gambling machine that creates a game of chance for its customers.
- Sittman and Pitt of Brooklyn, New York, developed a gambling machine in 1891.
- Modern slot machines enable manufacturers to assign a different probability to each symbol on each reel.
- A winning symbol may appear to the player to be close when the probability is much lower.
- Let’s call the probability of winning for a given slot machine $P(a)$.

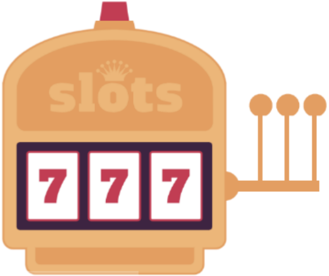
Multi-Armed Bandit



- Now, imagine a slot machine that, instead of one arm, has k arms.
- Each arm a_i has a probability $P(a_i)$ of giving a reward to the gambler.
- Suppose you have N coins to play, but you do not have any idea of $P(a_i)$ for all arms a_i.
- How do you choose the best arm?

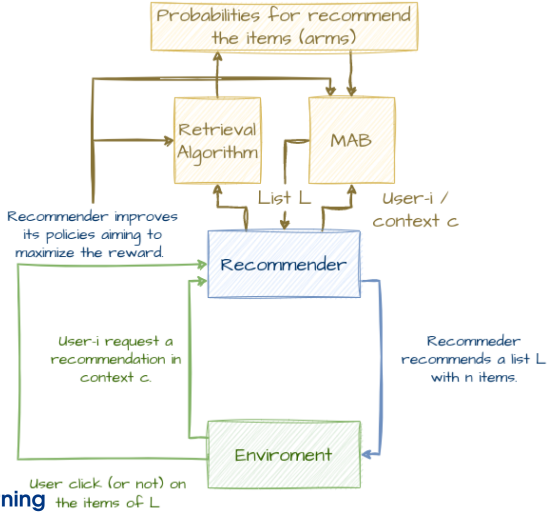
Once again, we will face the **exploration**/**exploitation** dilemma.

Multi-armed Bandit

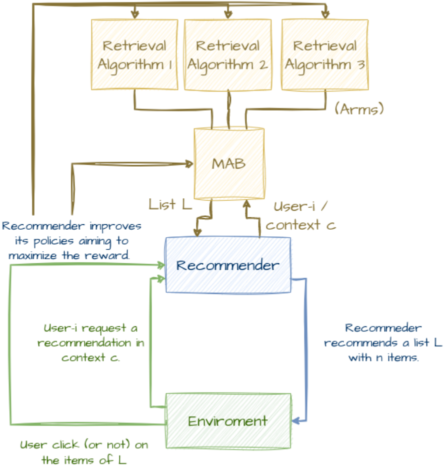


- The multi-armed bandit problem can be a metaphor for a a variety of applications , including **recommendation systems**:
- A variety of algorithms to deal with MABs:
  - Greed strategy
  - ϵ-greedy strategy
  - Upper Confidence Bounds
  - . . .

MAB – Variation 1


Probabilities for recommend the items (arms)
Retrieval Algorithm
MAB
List L
User-i / context c
Recommender improves its policies aiming to maximize the reward.
Recommender
User-i request a recommendation in context c.
Recommeder recommends a list L with n items.
Enviroment
User click (or not) on the items of L

MAB – Variation 2


Retrieval Algorithm 1
Retrieval Algorithm 2
Retrieval Algorithm 3
(Arms)
MAB
List L
User-i / context c
Recommender improves its policies aiming to maximize the reward.
Recommender
User-i request a recommendation in context c.
Recommeder recommends a list L with n items.
Enviroment
User click (or not) on the items of L

Reinforcement Learning in RS

- RL algorithms are often **challenging to evaluate with offline data** since we need online evaluation.
- To help understand how the algorithms work, we often perform simulations based on offline datasets.
- There are multiple RL algorithms, not just MABs, used in recommender systems.
- It is a relatively new area with a lot of research opportunities.

LLMs in Recommender Systems

- Large Language Models (LLMs) brought a paradigm shift in how we represent users and items.
- Instead of using only IDs or structured features, LLMs allow us to encode **textual, behavioral, and contextual** information.
- LLMs can process queries, descriptions, reviews, and conversations, enriching the recommendation pipeline.
- They naturally support **zero-shot** and **few-shot** scenarios, addressing cold-start challenges.

LLMs turn unstructured text into powerful semantic signals for recommendation. **However, they are not yet scalable for several recommendation scenarios**

How LLMs Transform Recommendation

- **Richer user modeling**: user history can be summarized or rewritten in natural language.
- **Better item understanding**: descriptions, reviews, attributes, and metadata are integrated into dense representations.
- **Conversational recommendation**: LLMs allow interactive, dialogue-based preference elicitation.
- **Generative recommendation**: models generate item rationales or even novel item combinations.
- **Alignment with goals**: LLMs can be prompted to prioritize diversity, novelty, serendipity, or constraints.

LLMs bridge recommendation, search, and natural interaction with users.

Agentic AI in Recommender Systems

- Agentic AI refers to AI systems capable of **planning**, **acting**, and **reflecting** autonomously.
- Traditional recommenders are mostly passive: they predict items based on observed preferences.
- Agentic recommenders take **initiative**: they ask questions, explore alternatives, and refine goals.
- These systems perform multi-step reasoning and interact with tools or APIs to enhance recommendations.

Agentic AI moves RS from mere prediction to **adaptive, proactive decision-making**.

Capabilities of Agentic Recommenders

- **Proactive Personalisation**: systems initiate interactions to reduce uncertainty (e.g., asking clarifying questions).
- **Multi-step planning**: instead of a single recommendation, the agent builds sequences (e.g., learning paths, onboarding flows).
- **Autonomous exploration**: balances exploration and exploitation similarly to RL-based recommenders.
- **Reflection loops**: the system analyses failures, updates internal strategies, and improves its policy.
- **Safety & alignment**: agentic autonomy introduces risks; constraints and human oversight remain essential.

Obrigado :) - Faculty of Information Technology