MLHB 21-22: Final Project Description

Guidelines:

- ***Research Topics:***
  Begin by choosing one of the recommended **research topics** (these can be found below). Topics differ in what units they relate to, what they focus on (e.g., modeling, experimentation, algorithmics), what prior knowledge they require (e.g., working with tensorflow or pytorch), how ambitious they are, etc. So plan ahead and choose well! (If you have good reason, you may also choose a topic that does not appear on the list, but must coordinate this with us first)
- ***Research Question:***
  For your chosen topic, clearly state a well-defined **research question**. Choose a research question that has real-world significance, but that at the same time, raises ***testable hypothesis(s)*** and/or ***concrete conjecture(s)***. Explicitly state your hypothesis - these can regard phenomena you expect to observe, performance you hope to obtain, tradeoffs you anticipate will arise, etc. Be as precise and as targeted as possible. Part of the project will be to validate these hypotheses (prove, disprove, or just give supporting evidence). Go for quality - not quantity.
- ***Setting/Environment:***
  Choose a **setting/environment** in which you plan to experiment (test your approach, simulate behavior, explore phenomena under different conditions, etc.). Remember that **simpler is better**: choose the minimal setup that still captures the core elements that are important for your goals (this can be harder than you think!). **Isolate** the parts crucial to testing your hypothesis, and **abstract away everything else**.
  - You are free to choose a setting that is as **real** or as **synthetic** as you like. The optimal tradeoff point between real and synthetic is that which best serves your goals in the project. Remember that it's very hard to say anything about recommendation without using some form of simulation.
- ***Submission Format:***
  The outcome of your project should be a **notebook** that explores your research question and related hypotheses. The notebook should tell a **story**:
  - Start with giving relevant background.
  - Describe the setup and environment in which you chose to experiment
  - State which parts are real (e.g., real data) and which are synthetic (e.g., simulated user behavior).
  - Present your approach.
  - Run experiments.
  - Report your results - use lots of plots!
  - Explain your results and how they relate to your hypotheses.
  - End with a discussion. Discuss the merits and limitations of your approach and results. If possible, draw practical conclusions.
- ***General Notes(!):***
  - The goal of the project is for you to demonstrate clear understanding, deep thinking, and an open mind to research. Don't worry if things don't work out as planned; if this happens, give concrete explanations for *why*. Focus on showing you've fully thought things through - whether they work as expected, or not.

- Be prepared to explain and ground all of your choices. Below you will find a form with guiding questions - fill this form and **submit** before you start working on your project.
- Finally - read these guidelines several times over, and again at different times through the course of the project. Reiterate and improve your choices and ideas in light of these guidelines (especially the parts about simplification, abstraction, and isolation).

Topics:

Below we list several research topics that you can base your project on. The topics vary considerably in multiple aspects, so be sure to plan ahead: think through the different steps of the project (as described above), and make sure you have a concrete plan for each step.

Important note: Please remember that this is the first time we give out these projects, so it's hard to fully anticipate how difficult or feasible the different topics are. Nonetheless, all of them have concrete potential to develop into full research projects - some more, some less (if this is of interest to you, talk to us about it so that we can help you choose). Know that some are safer bets, others are riskier. Choose with care!

1. **Set dependent collaborative filtering:**
   Collaborative filtering (CF) is a highly popular approach for predicting item relevance in recommendation settings. The underlying assumption is that relevancies are inner products between (latent) user features x and (latent) item features z. For binary relevance data (i.e., relevance inputs are in {+1,-1}), it seems plausible to use the conventional argmax approach and model choices as y=sign(<x,z>). But, as we've learned, this is akin to assuming users are rational, and in particular, means that this approach is IIA.
   The goal of this project is to devise a CF algorithm that allows for deviations from IIA, i.e., allows for user choices to depend on context - here, in the form of alternatives in choice set (note that the set of choices is not modeled in the standard CF approach). One way to approach this is to merge the standard CF framework with a known non-IIA approach for binary prediction; but other approaches are also welcome.

2. **Discrete choice and strategic dynamics**:
   The goal of this project is to explore a setting in which behavioral consumers and strategic suppliers clash. As we've learned, users are susceptible to certain behavioral context effects (e.g., attraction effect, compromise effect). These can be adding "bias" to the way in which (rational) users make choices. We've also learned that suppliers often act strategically to promote their own self interest, such as by modifying their items in ways which (they believe) increase profits or traffic to their items. This can be a source of conflict. Some questions that arise:
   a. Can strategic suppliers leverage the behavioral quirks (or weaknesses) of users (here, context effects) in order to get more traffic at reduced costs?
   b. What happens when multiple strategic agents try to exploit the same behavioral effects and at the same time? What happens in a dynamical setting? How does the system evolve over time, and does it converge?

c. What is the relation between the results and the “strength” of each behavioral effect in the environment we simulate? How do the results compare to an environment without behavioral effects? Do we observe a *qualitative* change of behavior when the “strength” of each behavioral effect is varied?

3. **Accuracy as a self-fulfilling prophecy**:
Predictive systems often display improved performance over time. In most cases, this is attributed to the investment of intentional effort aimed at obtaining improvement - collecting more data, retraining, modifying the algorithm, running A/B tests, etc. But in practice, it is very hard (or impossible) to determine what *truly* causes improvement. The goal of this project is to explore an alternative explanation for improved performance in recommendation systems, which is that accuracy improves over time *because prediction becomes easier*. The conjecture is that, as models are learned and deployed over time, recommendations cause the input data to shift towards distributions that are concentrated on easy-to-predict items (think why this should hold!). If true, learning over time would result in a "delusion" of improved predictive performance. The aims of this project are to establish and explore this conjecture, and to find conditions under which it holds - or does not.

4. **Emerging and vanishing phenomena:**
Learning to predict is in principle a passive operation - but ceases to be so once predictions are used for recommendations. Recommendations are like treatments, and as such, can change the environment they operate in. And since predictions rely on what class of predictive models are learned - the choice of model class itself may have implications on changes to the environment. In this project you can explore one of two such effects:
    a. **Independence of Irrelevant Alternatives (IIA):** Simple predictive models do not account for context effects, such as the dependence of choice behavior on the set of alternatives in a choice set. At the same time, real world data does not seem to exhibit set dependence. This bears the question: are user choices in these domains really set-independent, or do they only *appear* to be IIA because learning is done with IIA models? Are these just correlations, or does learning with IIA models actually *cause* choices to be more and more IIA (e.g., over time)? How can you tell? And will learning non-IIA models mitigate this effect, or will IIA still “creep in” through some back door?
    b. **Order dependence**: Simple predictive models do not account for the order in which items are displayed, but more advanced methods do. Will learning with order-invariant models cause choices to be less dependent on order over time? Conversely, will learning with order-sensitive methods reveal a true underlying dependence of users, if it exists? And what happens if it does not? Can we disconnect the cause-and-effect relations between recommendations and choices in terms of the importance of ordering?

5. **Model-induced distribution shift:**
By now, it should be clear that recommendations actively change the input distribution, and that retraining induces a dynamic process of distributional change. Some dynamics converge, others don’t; some fixed points are good, others are bad; some trajectories are favorable to others; and some outcomes are good for users,

while others are good for the system (sometimes for both!).

One approach towards understanding dynamics is to model model-induced distributional changes through an operator $D(\theta)$ that maps the current distribution to the following distribution as a function of the parameters of the learned model, θ.

Some interesting questions arise (you can choose to target one or more):

a. If you knew $D(\theta)$ - how would you use it?
b. How would you model $D(\theta)$ as a parameterized, learnable function? (remember it maps from distributions to distributions)
c. How can you learn $D(\theta)$? What sort of data would you need? When can we, and when can't we learn it?
d. What simplifying structural assumptions can you make about $D(\theta)$? (e.g., that it decomposes over individual user responses)
e. For a learned model, how can you express its uncertainty regarding predictions? (again, remember it maps from distributions to distributions)
f. Can you devise a greedy (e.g., retraining) procedure that learns both θ (for prediction) *and* $D(\theta)$? Can it help guide the trajectory? Can it aid in reaching stable points? How about *good* stable points? For what definition of “good”?

Guiding questions:

The questions below are intended to guide you in putting together the different pieces of your projects. Please fill in these questions and submit them (see link on course website). This will account for 10% of your project’s grade, so invest time and effort appropriately. (Note: these questions may not be a perfect fit for all topics and projects; if you feel yours does not fit well within these guidelines - talk to us about it)

1. Chosen topic:

   ____

2. Relevant course unit(s):

   ____

3. What is your project about? State concrete and testable research question(s).

   ____

   ____

   ____

4. State your hypotheses and/or conjectures. For each, explain how you plan to test it.

   ____

5. Describe the setting or environment you plan to experiment in. Give concrete details, and state which parts are based on real data (if at all), and which are simulative (remember that fully simulative is also fine if it justifies its purpose).

6. Describe your approach. Which learning algorithms will you use? How do you intend to construct the simulation? What experiments do you plan to run? What parameters do you intend to vary? Connect these to your research goal and hypotheses and/or conjectures.

7. What code do you plan to use, and from what sources? E.g., public packages/repos, code from homework/workshops, new code (if so, describe it in brief).

8. List three potential pitfalls that you anticipate may occur. Try to plan your response.

    1.

    2.

    3.

Good luck!
MLHB 21-22 team