Evaluation

Recommender Systems

Dimitris Sacharidis

need for evaluation

- When adding a recommender to an online platform
  - Variety of algorithms exist
  - Considering data and system constraints (type, timeliness and reliability of the available data, allowable memory and CPU footprints)
  - Which algorithm to employ?
  - The best performing one, having considered the constraints
- When suggesting a new algorithm
  - Comparing it with the existing one
  - Showing that it is worth using

- *Evaluation is to apply some **metric** that provides a (relative) assessment of a system*

evaluation taxonomy

- **Offline experiments**
  - Using existing datasets, collected traces, logs, etc.
  - Evaluating performance measures
    - system (prediction/classification/ranking) accuracy, others
- **User studies**
  - A small set of users is asked to perform a set of tasks
  - Focusing on user experience
  - More expensive than the first option
- **Online experiments**
  - Large scale experiments on a deployed system
  - Evaluate performance on real users
  - Users are oblivious to the conducted experiment

offline experiments – beyond accuracy

beyond accuracy

- Most recommenders are evaluated by their predictive power
  - The ability to accurately predict the users’ choices
- Accurate predictions are crucial, but insufficient
- People use recommender systems for more than an exact anticipation of their tastes:
  - discover new items
  - explore options
  - preserve privacy
  - …
- It is crucial to identify **properties** that may influence the **success** of a recommender system in the **context** of a specific application

item coverage

- accuracy typically increases as amount of data grows
- but not uniformly: for items with lots of feedback
- this problem is named long tail or heavy tail problem

- **item (or catalog) coverage** = proportion of items that a system can recommend
  - e.g., in CF, there are cold-start items for which the system has little feedback

- **item coverage inequality** = how skewed is the distribution over recommended items
  - measured by Gini index, distribution divergence metrics

user coverage

- in some cases system may not provide recommendations for some users due to:
  - low confidence in the accuracy of predictions
  - cold-start users

- **user coverage** = proportion of users that a system can recommend to

- other measures, like the average number of items a user must rate before receiving recommendations

trust

> *When the system recommends a few items that a user already knows and likes, even though the user gains no value, they observes that the system provides reasonable recommendations, which may increase their trust in the system.*

- hard to evaluate and measure trust
  - hard to separate trust from other factors of user satisfaction
- approaches:
  - user study to directly ask users
  - online test, measure number of recommendations that were followed, or repeated users
- a good way to increase trust is to provide <u>recommendation explanations</u>
  - why was this item recommended? e.g., popular, similar users, past consumption

diversity

A user is looking for a vacation recommendations, where the system recommends vacation packages. Presenting a list of five recommendations, all for the same location, varying only on the choice of hotel / attraction, may not be as useful as suggesting five different locations.

- **diversity** is a measure of **dissimilarity** among the recommended items (intra-list diversity)
  - dissimilarity is distance (opposite of similarity)
  - for a pair of items, computed using item content, item features, etc.
  - for a list, computed as the aggregate (e.g., mean, minimum) pairwise dissimilarity

novelty – serendipity

- **novelty** captures how **dissimilar** are the recommendations to the user's **history**
  - note the difference: diversity is dissimilarity *within* a list; novelty is dissimilarity of list to past

- **serendipity** is more subtle than novelty, it adds an element of surprise:

  A user has rated positively many movies where a certain star actor appears, recommending the new movie of that actor may be novel, because the user may not know of it, but is hardly surprising.

- but often serendipity = novelty

user studies

benefits

- evaluating recommenders is about learning from the interaction of users with the system

- hard to simulate this with offline experiments
  - user is not available
  - interaction with real users is more valuable

- sometimes preferred over online experiments
  - get to learn more about the users and their experience

how-to

1. Recruit a set of test subjects
2. Ask subjects to perform a set of tasks
   - Interaction with a recommender system
3. Observe, and record subjects’ behavior while doing their tasks
   - Collect quantitative measurements (e.g., portion of the task completed, accuracy of the results, time taken to perform the task…)

- Questionnaires before, during and after the task
  - Quantitative or qualitative questions
  - Enables us to collect data that is not directly observable (e.g., how enjoyable the user interface was, the difficulty of the task, …)

example

Test the influence of a recommendation algorithm on the users' browsing behavior of news stories

- The subjects are asked to read a set of stories interesting to them
- The subjects are split in two groups (**treatment** and **control**)
  1. Content that includes recommendations
  2. Content without recommendations
- With the user study we want to answer:
  1. Whether or not the recommendations are followed?
  2. Whether users read different stories (with and without recommendations)
- Data to be collected:
  1. e.g., How many times a recommendation was clicked
  2. Or, e.g., Track eye movement to see whether users looked at recommendations
  3. Qualitative questions about recommendations relevance

types of user studies

- **Between subjects** (A-B testing or All Between)
  - Each subject is assigned to a candidate recommender and experiments with it
  - Provides a setting closer to the real system (each user experiments with only one candidate)
  - Appropriate for testing long term effects of using the system (e.g., how the user becomes accustomed, estimate a learning curve of expertise)
  - More data is needed for reliable results -> costs more
- **Within subjects**
  - Each subject tests a set of candidates on different tasks
  - It is more informative (the superiority of one method cannot be explained by a biased split of users between candidate approaches)
  - Comparative questions (e.g., which candidate the subject preferred)
  - Users are more conscious of the experiment and hiding distinctions between candidates is hard

online experiments

benefits

- online experiments allow:
  - Direct measurement of overall system goals (e.g., long-term profit)
  - Understand how system properties (e.g., accuracy or diversity of recommendations) influence the overall system goals
  - Understand the trade-off between system properties

- provides the strongest evidence of the system’s true value
  - The system is used by real users that perform real tasks
  - Users are not aware of the experiment

how-to

- online experiments are usually employed:
  - After an extensive offline evaluations (provides evidence that candidate approaches are reasonable)
  - After a user study (measures the user's attitude towards the system)
  - Gradual process reduces the risk of significant user dissatisfaction

- many real-world systems employ an online testing system
  - Redirecting a small percentage of the traffic to different alternative recommendation engines
    - Sampling (redirecting) users must be random -> fair comparisons
  - Recording the users' interactions with the different systems

- such experiments are risky
  - e.g., an alternative recommendation engine (a bad one) might discourage the test users from using the real system ever again

reliable conclusions

choosing the best recommender

- simple procedure:
  - consider an evaluation metric
  - compare alternative recommenders using that metric
  - choose the one performing the best
- have we made the right choice?
  - level of uncertainty: will this system work as well in real-world?
- what is our confidence of the **performance gap**?
  - are there cases, where we cannot draw a clear conclusion?
- to draw reliable conclusions:
  - repeat experiment over multiple subjects and present the **standard deviation** besides the mean of metric
  - go one step further and test for **significance**

significance testing

- assume that a **simple random mechanism** (the null hypothesis) produces the results we see
- compute the probability with which this mechanism produces these results, or more extreme
- this probability is the **p-value** of the test

- if p-value is **low** (typically < 0.01 or 0.001), this means:
    - most likely the null hypothesis is not responsible for the results we see
    - the conclusions we draw are considered reliable (alternative hypothesis)

- the threshold we use for p-values is called the **significance level**

Acknowledgements

some content from:

- Amra Delić