

Evaluating Recommender Systems

If You want to double your success rate, you should double your failure rate.

<https://www.ksi.mff.cuni.cz/>

Today's Content

- **Organization + Recap**
- **Evaluating RS**
 - Motivation
 - Subjects, Variables, Settings, Design
 - Variants
 - Online Evaluation
 - User/Lab Studies
 - Offline Evaluation
 - Best practices, evaluation „f**k-ups“

Recap

- Basic CB with meta-data?
- Vector Space Model (Rocchio's Method)
- What is wrong here?



Organization

- **Semestral work. Details to be discussed during labs**
 - Using EasyStudy, construct a new user study with certain extensions of the original framework
 - Define research questions and hypotheses (what should work better and why)
 - Enhance EasyStudy by
 - Adding a new algorithm or a post-processing method (e.g., diversity enhancement)
 - Run the study locally (collect dummy data)
 - Create basic evaluation scripts / Evaluation screen for users
 - Select some of the pre-defined variants, or propose your own
- **Deadlines:**
 - By May 15: finalize the proposal of what you plan to do (*get approval*)
 - By May 31 or exam date (whatever is sooner): finish the project if you want to get a bonus for exam
 - Later finalization possible (no bonuses)



Evaluating Recommender Systems

Evaluating Recommender Systems



- **A myriad of techniques has been proposed, but**
 - Which one is the best in a given application domain?
 - What are the success factors of different techniques?
 - When/why certain methods do not work?

- **Possible research questions:**
 - Is a RS more **effective** with respect to a specific criteria like **accuracy**, user satisfaction, response time, serendipity, diversity, online conversion, ramp-up efforts, retention of users
 - Do **customers** like/**buy** recommended items?
 - Do customers buy items they **otherwise would have not**?
 - Are they **satisfied** with a recommendation after purchase?
 - Do they **keep using** our site **thanks** to the **recommendations** they're receiving?

(Can we assure that the improvements were caused by the RS?)

Background: Requirements on Evaluation in Science

- Scientific experiments must ensure **reproducibility** in order to verify the findings
- How an evaluation research is done?
 - Thoroughly describing the **methodology**
 - Following a **systematic procedure**
 - **Documenting** decisions made
- Criteria that must be pursued
 - Validity (internal and external)
 - Reliability
 - Sensibility



Evaluation research – Criteria to pursue

- **Internal validity**

- The observed effects are due to the controlled test conditions instead of differences in the set of participants (aka predispositions), or the environment
 - *I.e., results are better because we applied a better method*

- **External validity**

- Results are generalizable to other user groups or situations
- The scenario is representative of a real-world situation
 - *This works for me & my buddies*, vs. *this works for everyone*

- **Reliability**

- Absence of inconsistency and errors in data and measurements

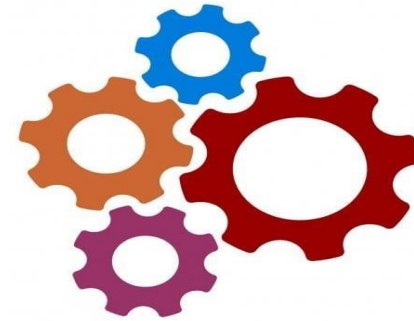
- **Sensibility**

- Different evaluations of the observed aspects reflect in difference in the measurements

Evaluation research - Components



Subjects



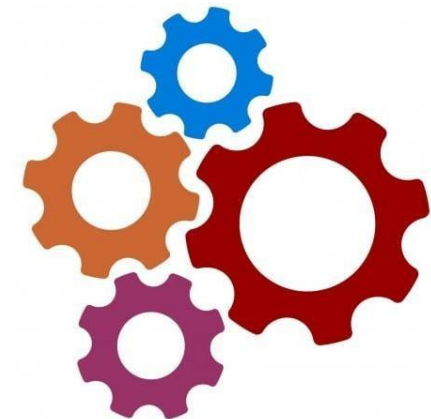
Settings



**Measured
Variables**



Design



Evaluation research - Settings

Depending on to the conditions in which the data are collected:

- **Offline studies**

- Conducted on real-world data, aiming to predict hidden part of the data
- But analysis is done after the fact...
- *A lot of improvements recently towards de-biasing & multi-criterial evaluation*
 - *But still not very reliable*

- **Field studies (Online evaluation)**

- Conducted in an preexisting real-world environment (production websites)
- Users are intrinsically motivated to use a system
 - ***But experiments cost you (potentially a lot)***

- **Lab studies (User Studies)**

- Expressly created environment for the purpose of the study
 - ***We can get more/better feedback***
- Variables can be controlled more easy by selecting study participants
- But doubts may exist about participants motivated by money or prizes
 - ***Do they behave in the same way as your customers?***

Evaluation research - Subjects

- The subjects of recommender systems research are usually specific subgroups of people
 - Online customers, web users, who receive adaptive and personalized recommendations
 - **Need for balance:** ideally, groups of users with the same properties should evaluate each tested condition
 - Simple for offline settings, more challenging for field/lab studies



Evaluation research - Measured Variables

- **Aspects of the users** (or subjects) that are relevant in the context of the recommendation system and the research question examined
 - Demographics, background check, psychological profiling,...
- **Independent variables**
 - Variables that are static throughout the course of the experiment
 - *E.g., the way how individual items are displayed*
- **Control conditions**
 - Controlled by the evaluation design (*e.g., the recommendation algorithm*)
- **Dependent variables**
 - Assumed to be influenced by the independent variables and control conditions
 - *E.g., higher reported satisfaction with certain algorithm / higher nDCG@10 / more conversions...*



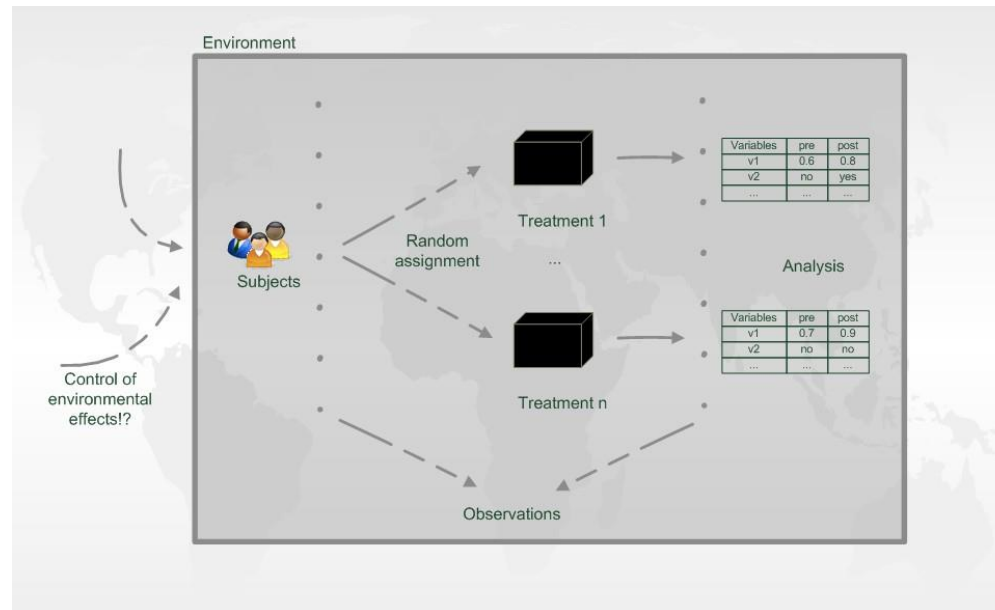


Evaluation research - Design

- **Experimental**
- Quasi-Experimental
- Non-Experimental
- Cross-Sectional
- Case-Studies

Experimental Research Design

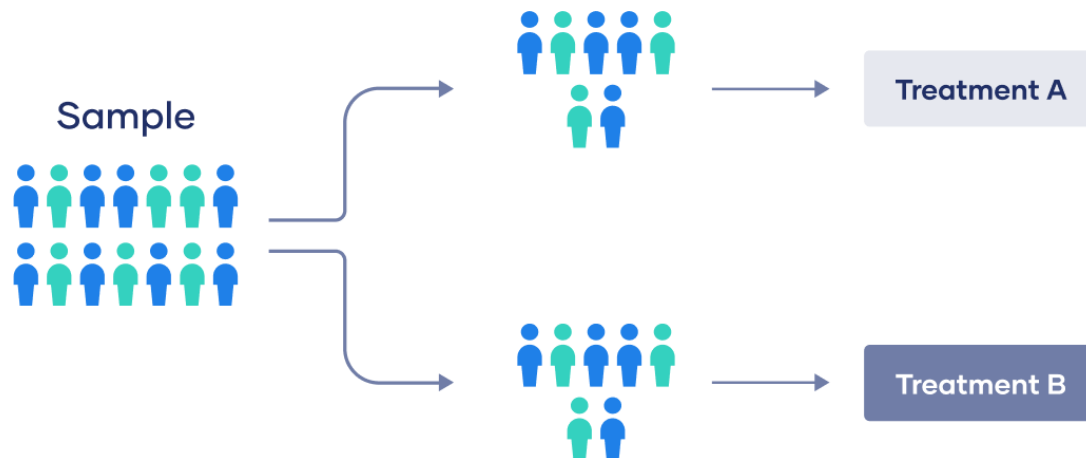
- One or more of the independent variables are manipulated to ascertain their impact on the dependent variables
 - ***The most common approach for RecSys research***



- *A/B testing in online settings*
 - *users are randomly directed to recommendations from algorithm A or B*

Experimental Research Design

- One or more of the independent variables are manipulated to ascertain their impact on the dependent variables
- Between-subject design
 - Each participant experiences only one treatment
 - Simplest variant, suitable also for field studies
 - The critical part is to balance the sub-samples well (or make them large enough😊)



Experimental Research Design

- One or more of the independent variables are manipulated to ascertain their impact on the dependent variables
- Within-subject design
 - Each participant experiences all the treatments, typically in sequential order
 - Alternatively, the display provide results of multiple variants
 - Sequential design not very suitable for field studies (carry-over effects)

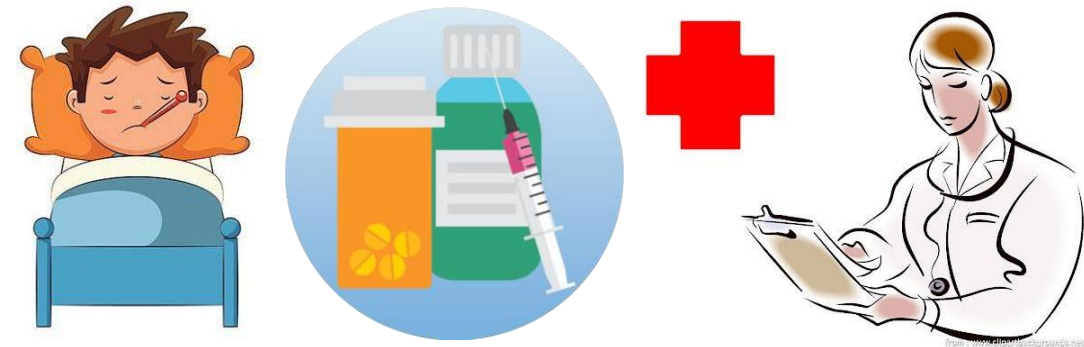


- *Implicitly, this is what you do for offline evaluation*
- *Mixed designs are also possible (typically when having multiple independent variables)*
 - *Some variables treated as within-subject, others as between-subject*

Quasi-Experimental Design

- Different from Experimental design from its lacking of random assignments of subjects to different treatments

- Subjects decide on their own about their treatment



- Results must be interpreted with utmost circumspection

- *Differences between treatment / no treatment groups*

- Conclusions need to be drawn very carefully

- *But, may be suitable to test optional parts of design*

- e.g., controlling interface



Non-Experimental Design

- Quantitative research
 - Numerical measurements of different aspects of objects
 - For instance, asking users different questions on the perceived utility of the recommendation applications
 - Answers usually on a Likert scale
- **Qualitative research**
 - Interviews with open-ended questions, record think-aloud protocols, focus group of discussion
 - For instance, to explore user's motives for using a recommender system
 - Understand what features must a good recommendation have
 - „What does serendipitous recommendations mean for you?“
- Longitudinal research
 - The entity under investigation is observed repeatedly over time
 - For instance, to evaluate the impact on the long-term of the use of a recommender system



Other Designs

- Cross-Sectional design
 - Analyze relations among variables that are simultaneously measured in different groups
 - Allows generalizable findings from different application domains to be identified



- Case Studies
 - Combine whichever types of quantitative and qualitative methods necessary to investigate contemporary phenomena in their real-life contexts
 - Focus on answering research questions about how and why

Questions???



Experimental settings

Online experiments



Online Experimental Settings

- **A/B testing**
 - *Evaluate metric as close to the actual target behavior as possible (KPI, key performance indicators)*
 - Retailer's target variable is (long-term) profit
 - i.e. Netflix's target variable is monthly subscribes
 - Usually, larger overall consumption increase profit (but beware of extreme cases)
 - Broadcaster's target variable may be the overall influence / total mass of readers /... but also maintaining editorial values
- **The direct effect on KPIs may be too small**
 - How much does one small parameter change affect retention of users?
- **The target variables may be hard to measure directly**
 - E.g. has long-term effect only / cannot extrapolate all external variables
- **Proxy variables**
 - Loyalty of user, Conversions rate, Basket size / value, Click through rate, Shares / Follows /...

Rules to live by 😊

- *Always design evaluation metrics with respect to your target variables*
 - *However select something, where the effect is measurable*
 - **In either case, be careful about long term effects**
 - **Cascade of evaluation metrics**
 - From high to low detectability of changes
 - From low to high impact on your true target
 - **Continuously measure the effectivity of your recommendations**
 - Even when no experiment is currently running
 - Apply automated alerts when some metrics substantially deviate
-

Common on-line evaluation metrics – E-commerce

A light blue arrow pointing upwards, indicating increasing potential impact.

Higher potential impact

- **Recommending correctness**
 - Visit (once) recommended object (i.e. ignore page layout)
- **Click-through rate (CTR)**
 - **Recommend** -> **Click** / Click and do sth. (do not leave immediately)
- **Conversions rate (CVR)**
 - Recommend -> Purchase / **Click -> Purchase** / Recommend-> Click -> Purchase
 - Share / follow / like / ask about... recommended item
- **Cross-sale increase**
 - Add to cart -> Recommend -> Add another (recommended) to cart
 - Returning buyers, viral effect, market shares...

A light blue arrow pointing downwards, indicating a better proxy of target variable.

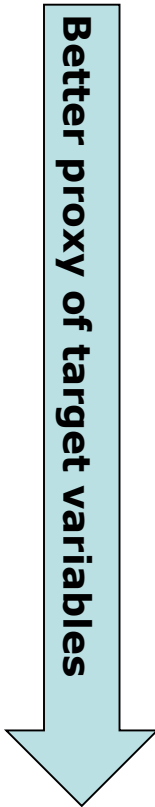
Better proxy of target variable

Common on-line evaluation metrics – Broadcaster/News/Info



Higher potential impact

- **Recommending correctness**
 - Visit (once) recommended object (i.e. ignore page layout)
 - Reflecting editorial values (diversity, significance, minority voices...)
- **Click-through rate**
 - **Recommend** -> **Click** / Click and do sth. (do not leave immediately)
 - [beware of clickbaits]
- **„Conversions“ (engagement) rate**
 - Share / follow / like / comment... recommended item
- **Value per user**
 - Total time / number of visited objects / displayed ads... per user or per session
 - Returning rate of users / user loyalty
 - Premium subscription rate



Better proxy of target variables

Beyond performance indicators – Technical metrics

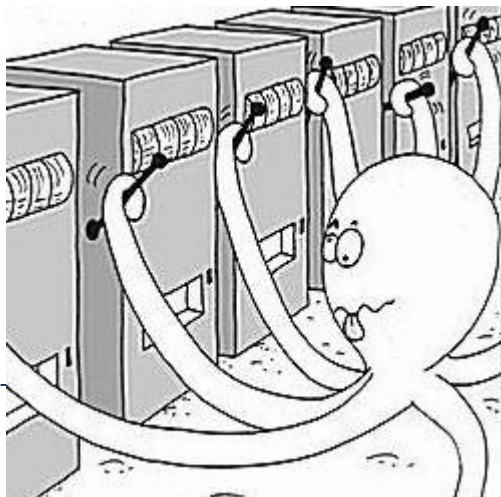
- **!!Response time!!**
- **Train / re-train / model update time**
- **Memory/CPU/GPU consumption**
 - How large can we grow with current infrastructure?
- **Recall on objects**
 - Is portion of your objects ignored? Are there too many low-profit bestsellers?
- **Ability to predict**
 - Can you calculate recommendations for all users?
 - For which groups of users are we better than baseline?

Beyond performance indicators – Fairness

- **Recommender environments have multiple stakeholders**
 - Consider, e.g., Amazon or Spotify
 - Users who consume products
 - Providers who supply products
 - The platform itself
- **Provider fairness:**
 - Products that are similarly good for the user should receive similar attention
 - The platform should not discriminate any subgroup of providers (e.g., small local bands)
- **Consumer fairness:**
 - The recommendations should be similarly good for all subgroups of users
 - *Only suggesting the cheapest pubs for anyone from Ostrava is not a good idea 😊*

Multiarmed bandits evaluation

- **Bad side of A/B testing? It costs a fortune**
 - Organizational requirements (**Necessary**... Study builder frameworks may help to some extent)
 - Users receive sub-optimal recommendations (**we can do sth. about that**)
- **Multiarmed bandits**
 - Alternatives that seems less relevant gradually receives less attention
 - Epsilon-greedy, Thompson sampling, upper confidence bounds (UCB)



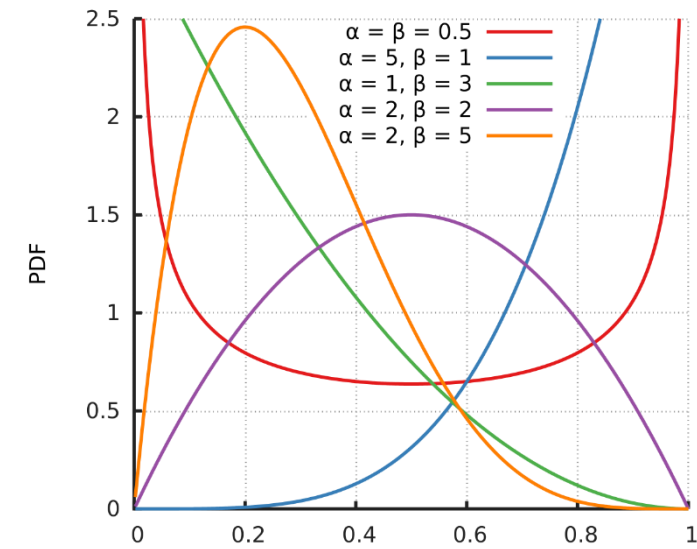
Multiarmed bandits evaluation

- **Epsilon-greedy**
 - Calculate average success (sum of reward / # of trials) for each arm
 - Select the current best arm with the probability of ϵ , while with $(1 - \epsilon)$ probability select at random
 - ϵ may increase in time (supposed cooling effect)
 - Simple & somehow works
 - But, no optimality guarantees

Multiarmed bandits evaluation

■ Thompson sampling

- Assume Bernoulli (binary) rewards, for each arm keep
 - α = number of successes (positive outcomes) + 1
 - β = number of failures (negative outcomes) + 1
- Assume that the distribution of rewards follows a Beta(α , β) distribution
 - For each arm, sample a value from corresponding Beta distribution
 - Select the arm with the highest sampled number
- Update α and β of the selected arm (based on the result of the trial)
- Guarantees to minimize regret under some conditions
- Rather quick convergence (fairly soon, only the best arm is being selected)
 - This may be a problem for long-term effects (limit history)



Multiarmed bandits evaluation

- **Upper Confidence Bound (UCB)**

- At each step, select the arm with **with the highest upper confidence bound** based on the observed rewards.

- **UCB = estimated reward + certainty threshold**
- **Estimation = mean reward**
- **Certainty threshold determined by the number of trials**

$$UCB_i(t) = \hat{\mu}_i(t) + \sqrt{\frac{2 \ln t}{N_i(t)}}$$

- $\hat{\mu}_i(t)$ is the estimated average reward of arm i at time t .
- $N_i(t)$ is the number of times arm i has been pulled up to time t .
- $\ln t$ is the natural logarithm of time step t

- **Selects either the arm that is good (and you are fairly certain about it), or the arm, which did not seem so good so far, but you did not sufficiently test it yet.**
- **Time factor increases confidence bounds over time (soft nudge to forget)**

Wellcome to the ***DARK SIDE***

- ***It is possible to incorporate some providers metrics into the recommendation proces***
 - *i.e. recommend items with higher profit*
- ***Do that wisely (or not at all)***
 - *Your credibility is at stake if someone finds out*
 - *User trust in recommendations is one of the most important features determining the long-term effect of recommender systems*
 - *Do not behave like ALZA a couple years ago...*

Alza dostala pokutu za přidávání zboží do košíku. A není jediná

🕒 12. prosince 2018 16:09



Společnost Alza.cz bude muset zaplatit 150 tisíc korun za to, že zákazníkům přidávala do košíku zboží, na které sami neklikli. Pokuta byla udělena za případy z jara 2018, stejné zkušenosti měli zákazníci i nyní v prosinci. Za stejný prohřešek byl v posledních týdnech potrestán i e-shop Kasa.cz.

Evaluating Recommender Systems - II

If You want to double your success rate, you should double your failure rate.

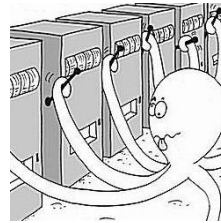
<https://www.ksi.mff.cuni.cz/>

Recap

- Variants of RS Evaluation Settings



- Multi-armed bandits



- Quasi-experimental design?

Organization

- **Active Reading #2**

- Knowledge graphs: 12

- Tuning and computational complexity

- Unclear hyperparam settings for other methods

- (Everyone's a Winner paper @RecSys23, <https://dl.acm.org/doi/10.1145/3604915.3609488>)

- Is 1% improvement „notable“?

- Visual BPR: 16

- Only handle item cold start, only implicit feedback, only quantitative evaluation

- Do we trust our sources (fraudulent images)

- Similar vs. Complementary issues (*Outfit prediction RS branch*)

- **Asking ChatGPT about some submitted texts:**

- I'm highly confident that the text you provided is generated by a model similar to ChatGPT, so I would rate my certainty as a 9 out of 10.*

- I may ask about the papers during exams...



Organization

- **Active Reading #3: post-processing algorithms**
 - **Harald Steck, Calibrated Recommendations:**
<https://dl.acm.org/doi/10.1145/3240323.3240372>
(how to make recommender systems to follow distributions in the user profiles)
 - **Rodrygo Santos, Explicit web search result diversification:**
<https://dl.acm.org/doi/abs/10.1145/2492189.2492205>
(how to provide diversified recommendations - xQUAD algorithm)
 - **Kunaver & Požrl, Diversity in recommender systems – A survey:**
https://papers-gamma.link/static/memory/pdfs/153-Kunaver_Diversity_in_Recommender_Systems_2017.pdf
(how to provide diversified recommendations - a broader yet compact survey with several particular solutions and their comparison)

- **Deadline: April 28**

Organization

- **Active Reading #4: Beyond Algorithms & Impact of RS**
 - **Jannach & Bauer, Escaping the McNamara Fallacy: Toward More Impactful RS Research:**
<https://onlinelibrary.wiley.com/doi/10.1609/aimag.v41i4.5312>
(How can we measure the real value of recommender systems?)
 - **Tomlein et al., An Audit of Misinformation Filter Bubbles on YouTube: Bubble Bursting ...:**
<https://dl.acm.org/doi/10.1145/3460231.3474241>
(How is disinformation spread in real-world recommendation services?)
 - **Sürer et al., Multistakeholder Recommendation with Provider Constraints:**
<https://dl.acm.org/doi/pdf/10.1145/3240323.3240350>
(There are other entities the RS should take into account - beyond the end-users)

- **Deadline: May 19**

Lab Studies



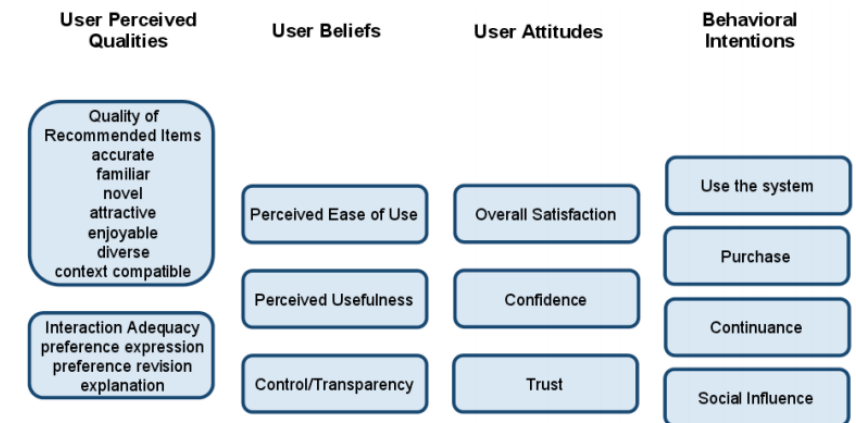
Lab studies settings

- **Same things possible as in online experiments**

- Preference elicitation phase often needed (no history for participants)
- External knowledge for collaborative algorithms needed

- **In addition to online experiments**

- Questionnaire / Interview
 - **Features otherwise hard to observe**
 - Helpfulness / Ease of use / Relevance
 - Trust, Novelty to the user etc.
 - Reasoning behind actions/inaction
- Physiological response
 - **Eye tracking**, facial expression detection etc.



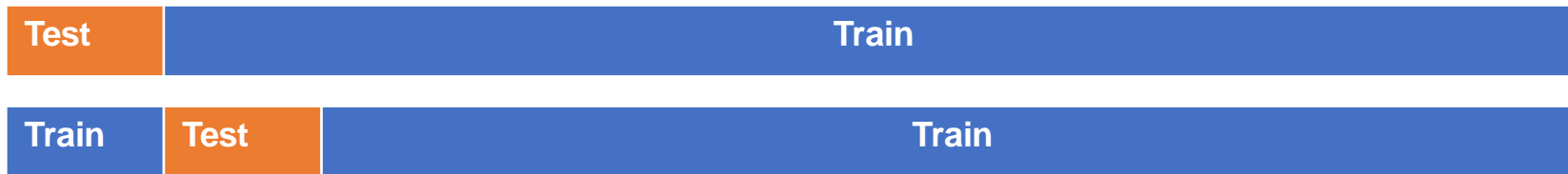
ResQue framework,

<https://dl.acm.org/doi/10.1145/2043932.2043962>

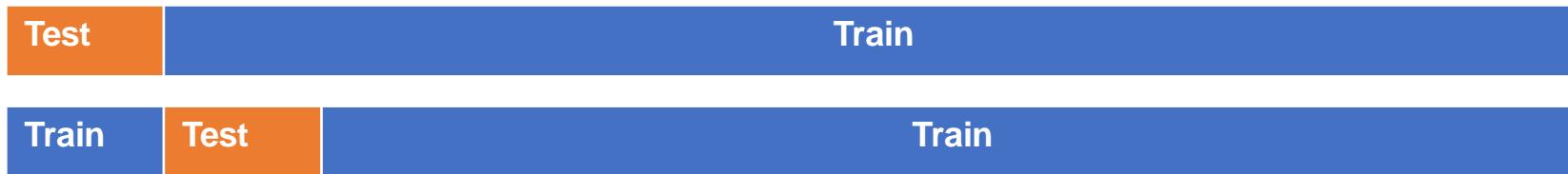
- **Key criterion in lab studies is that subjects should well approximate behavior of your real users**

- This may be harder than it seems (questionable external validity)
- Carefully consider what tasks to give them and then re-think it once more 😊

Offline evaluation



Offline evaluation



Offline evaluation settings

- **Largely used in experimental evaluations for recommender systems**
 - Cheap to conduct, Easy to scale/parallelize, Easy to replicate
 - No need to have an access to real-world services
 - No need to contract human users of the system
- **Common working principle:**
 - Existence of some „correct“ (gold truth) but hidden data (e.g., a portion of users ratings)
 - RS aim to predict such hidden data (e.g., recommend items user did rate positively)
- **Ingredients needed:**
 - Dataset
 - Methodology (e.g., data splits)
 - Metrics to evaluate

Datasets



- Collection of synthetic or historical user interaction data
 - Ratings, purchase transactions, clicks, impressions
 - May also contain other data (e.g., item meta-data, item external identifiers etc.)
- *Synthetic datasets ensure certain properties (distribution, size, sparsity)*
 - *Risk that such datasets are biased toward the design of a specific algorithm and that they may treat other algorithms unfairly*
- Natural datasets include historical interactions record from real users
- Worth to analyse
 - Dataset size and sparsity
 - Dataset domain
 - Available information
 - CB data, impressions, feedback granularity,...



Datasets

Dataset	Domain	Feedback	Size
Netflix Prize Dataset	Movies	Ratings	17770 movies, 480189 users
Movielens 20m	Movies	Ratings	27278 movies, 138493 users
DePaul Movie	Movies	Ratings + Contextual information	79 movies, 97 users
LDOS-CoMoDa	Movies	Ratings + Contextual information	4000 movies, 200 users
MillionSongs Dataset	Music	Listening history	> 1M users
The Million Playlist Dataset	Music	Listening history	1M playlists, over 2M unique tracks from Spotify users
Kgrec Music Dataset	Music	User's interactions	8,640 items, 5,199 users, 751,531 interactions
South Tyrol Suggests	Tourism	Ratings	Not specified
Job Recommendation Challenge Dataset	Job Recommendations	User's applications and work history	Not specified

<https://github.com/RUCAIBox/RecSysDatasets>

<https://github.com/caserec/Datasets-for-Recommender-Systems>

<https://github.com/otto-de/recsys-dataset>

RecSys Challenges

(<https://recsys.acm.org/recsys24/challenge/>)

Methodology

- Define how the data available in the dataset are used to train the system and to evaluate its performances

Usually splitted into two sets:

- **Training set**
 - Data used to train a machine learning (ML) model
 - We use the feedback contained in this subset of data to determine the model in order to fit the data
- **Testing set**
 - Data used to evaluate the model
 - We use the trained model to predict the expected feedbacks
 - We compare the predictions with the real data to assess the performances

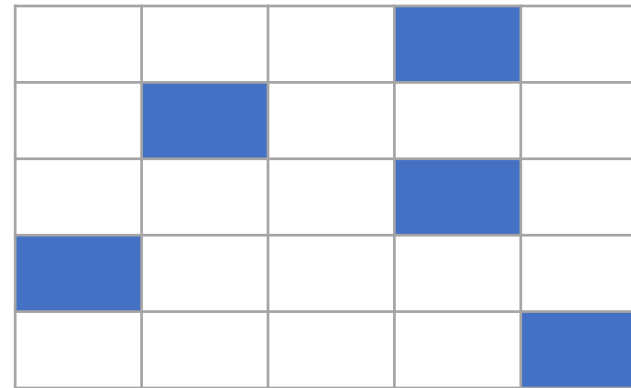


How can we split our data into training and testing sets to increase the reliability of our results?



Holdout

- A random set of ratings is withheld from user-item matrix, and it is used as test set (x%)
- The remaining ratings are used as training set (100-x%)
- Risk of **overfitting**
 - tested users are not totally unknown to the model
- 20/80 holdout
- Sometime **Stratified**
 - For instance, we want to ensure that for each user we have data in both training and test set



			■	
	■			
			■	
■				
				■

N-fold cross-validation

- Users in the user rating matrix are divided into N partitions
 - Normally 10, common splits 8-2 or 9-1 (80/20, 90/10%)
 - Other alternatives, e.g. Monte-Carlo CV
 - N-1 partitions are used for the training
 - the remaining partition is used for the test
 - average of all splits produce the result

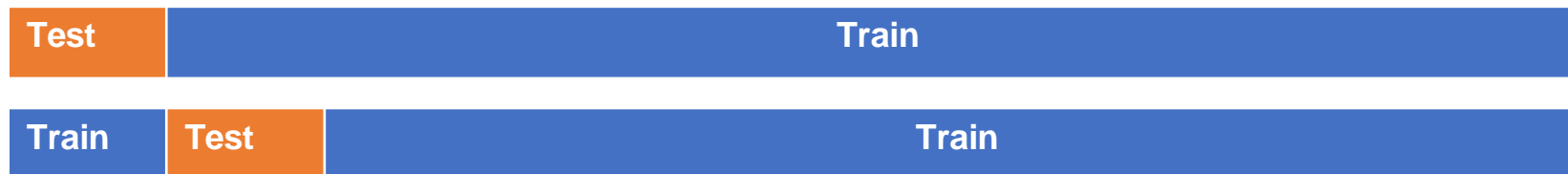
N-fold cross-validation

- Users in the user rating matrix are divided into N partitions
 - Normally 10, common splits 8-2 or 9-1 (80/20, 90/10%)
 - N-1 partitions are used for the training
 - the remaining partition is used for the test
 - average of several ways of splitting the data



N-fold cross-validation

- Users in the user rating matrix are divided into N partitions
 - Normally 10, common splits 8-2 or 9-1 (80/20, 90/10%)
 - N-1 partitions are used for the training
 - the remaining partition is used for the test
 - average of several ways of splitting the data



...

N-fold cross-validation

- Users in the user rating matrix are divided into N partitions
 - Normally 10, common splits 8-2 or 9-1 (80/20, 90/10%)
 - N-1 partitions are used for the training
 - the remaining partition is used for the test
 - average of several ways of splitting the data



Leave-P-out validation

- Usually leave-1-out
- During testing, for each user, test one rated item at a time for the test set
- Leave-p-out: test predicted ratings for p items

		?		
		?		
		?		
		?		
		?		

		?		?
		?		?
		?		?
		?		?
		?		?

- *For session-based RS: leave last interaction out and try to predict that one*

Hyperparameters fine-tuning

- The trained model behavior depends on its hyperparameters
 - KNN – Number N of selected neighbors
 - Linear SVM – C specifying “width of margins” of the decision boundary
 - Decision Tree – maximum height of the tree
 - Random Forest – Number of trees
 - User-based CF – number of similar users
- How to find the “best hyperparameters”?
 - We try several possibilities during the training using a part of the training set



- Once we find the best hyperparameters, we train the model on the whole training set



Hyperparameters fine-tuning

- The trained model behavior depends on its hyperparameters

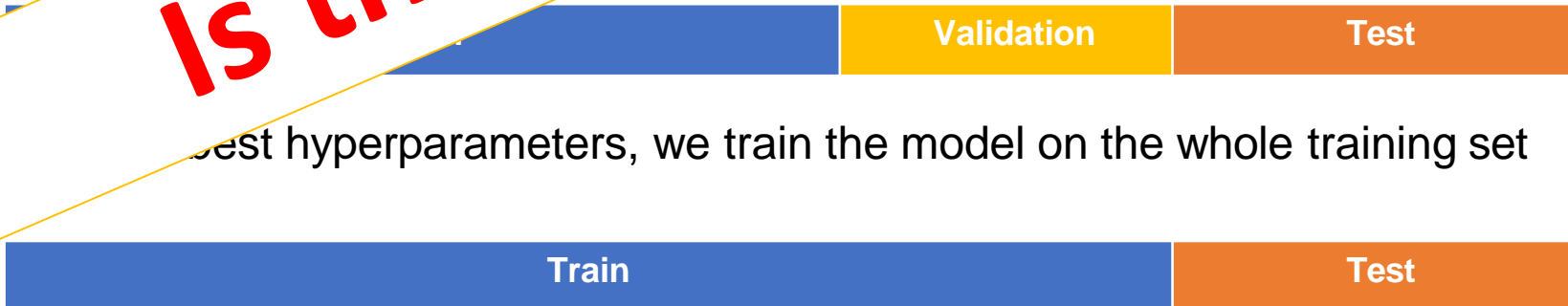
- KNN – Number N of selected neighbors
- Linear SVM – C specifying “width of margins” of the decision
- Decision Tree – maximum height of the tree
- Random Forest – Number of trees
- User-based CF – number of similar users

- How to find the “best” hyperparameters

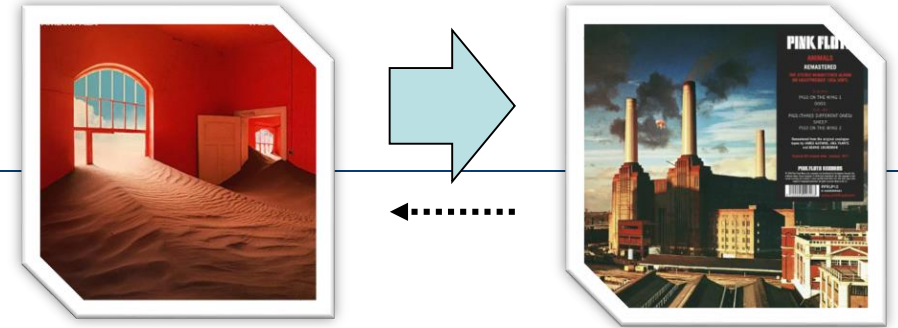
- We try several hyperparameters, training using a part of the training set

Is this good enough?

For the best hyperparameters, we train the model on the whole training set



Offline evaluation settings



■ Beware of causality and biases

- Having $A \rightarrow B \rightarrow C$, predicting C from A may be more difficult than predicting A from C (user-wise causality)

- When did people started liking the item? (dataset-wide causality)



- How were the impressed items selected (dataset biases)

■ Mitigation strategies

- Use temporal data splits, or simulation-style evaluation
- In your data: record impressions and work with them during evaluations
- In 3rd party data: apply appropriate debiasing strategies (e.g. popularity debiasing)

Evaluation of Offline experiments

- **Temporal splits**

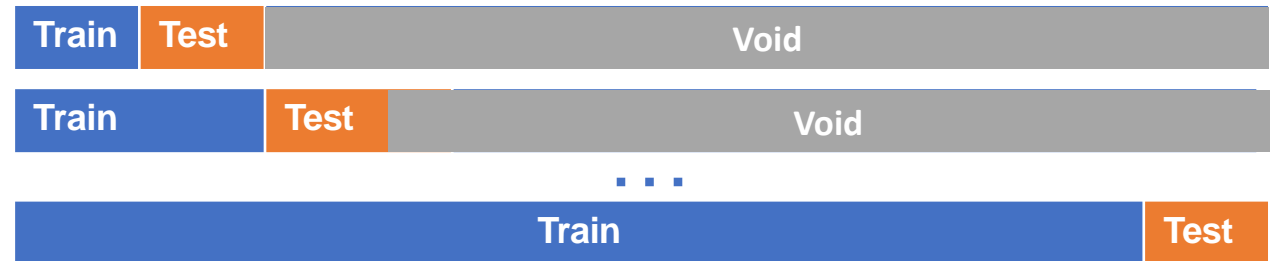
- Older data as train set, newer as test set

- Per-user splits

- Global splits



- Multiple temporal splits



Evaluation of Offline experiments

■ Simulation-style evaluation

- Keep the stream of events as it was in the original data (recommendation prompts included)
 - Try to re-play what recommendations would the new policy give
 - Probably most precise, but rather costly

■ Biases

- Algorithm biases:
 - Users are less likely (often not at all likely) to provide feedback on items that were not recommended
 - Limit predictions on impressed data only (or, impressed plus some random subset)
 - Use some well-known properties of algorithms (e.g., popularity amplification)
 - More subtle issues:
 - Previous algorithm was able to gain richer user profiles. Current algorithm is better at utilizing it (exploitation), but inferior in gaining it (exploration)
 - Difficult even for online evaluation... Train data sampling w.r.t. Target algorithm distribution?
-

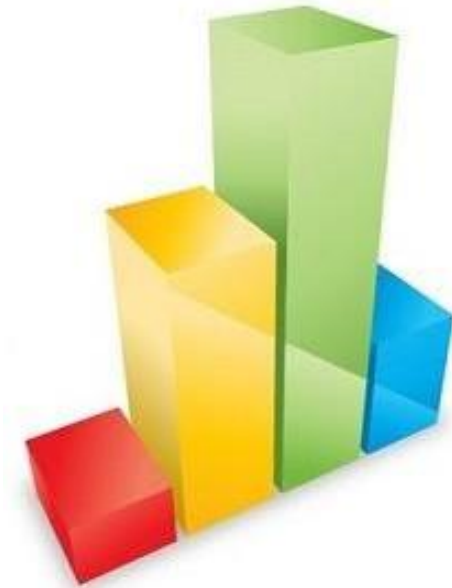
How we measure the results?



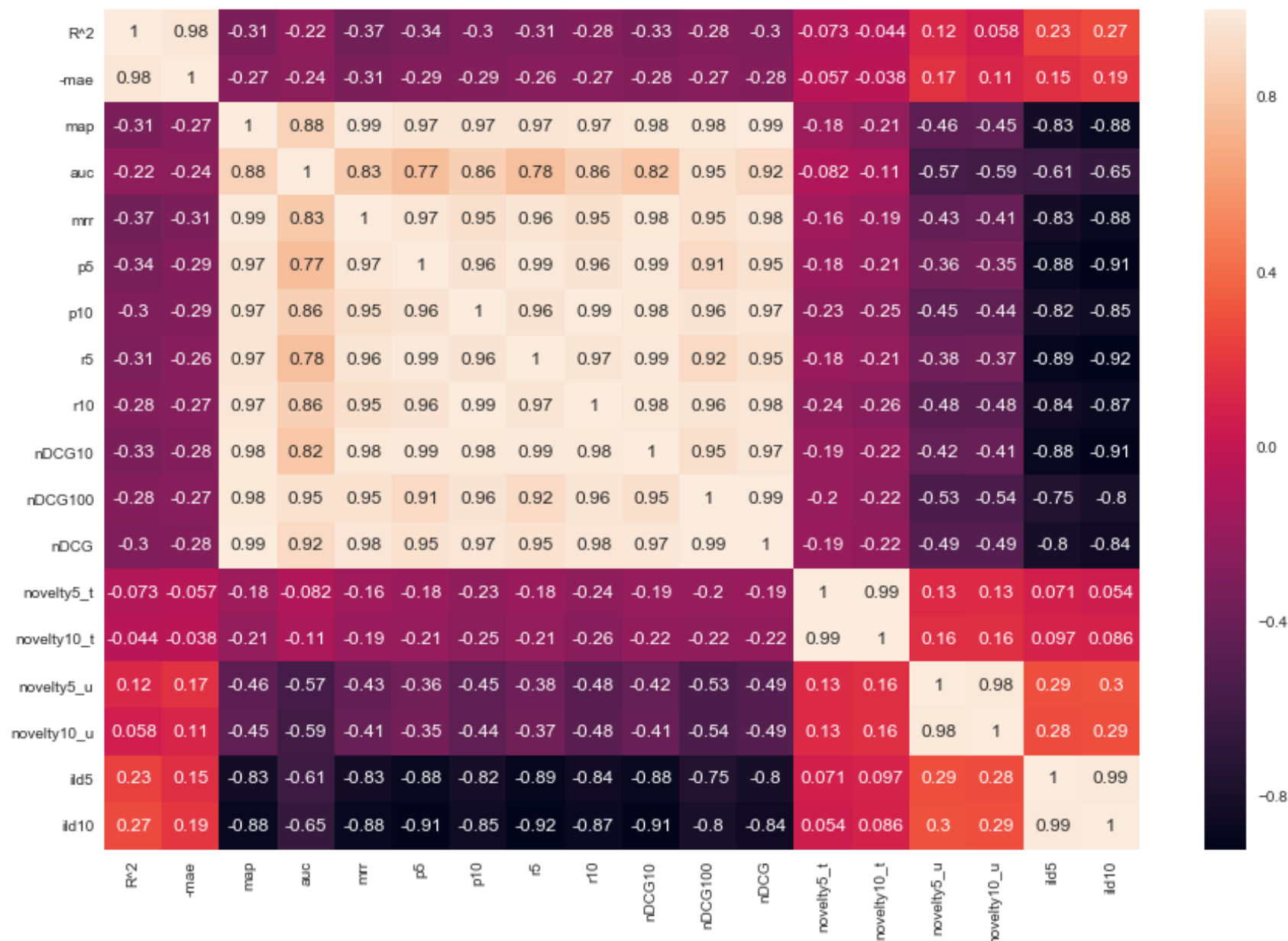
Offline Evaluation Metrics

Metrics selection depend on our research questions and hypotheses we want to test

- **Relevance of the recommended objects / Ranking metrics**
 - User visited / rated / purchased... the objects, which the method recommends
 - **nDCG**, **MAP**, Precision, **Precision@top-k**, Recall, Liftindex, RankingScore,...
 - **Rating error metrics**
 - MAE, RMSE,... Not very relevant nowadays
 - **Novelty**
 - Was the object new for the user? (May be both positive and negative depending on the task)
 - However, recommending previously known items is rather trivial
 - Several views: novel w.r.t. creation time, novel w.r.t. volume of interactions (= long tail items)
 - **Diversity**
 - Are all the recommendations similar to each other? (*Harry Potter issue*)
 - Generic **Intra-List Diversity** (pairwise diversity of items)
 - Can be expressed w.r.t. both content-based and collaborative features
-
- **Coverage, Serendipity, popularity bias...**



Metrics: Comparison



Beyond-accuracy / beyond-relevance metrics



Off-line Evaluation Metrics

▪ Novelty

- Items not known (does not have feedback / were not recommended) by the user
- Well known items (blockbusters in movies/books), based on overall consumption
- Items that are new (have been added recently)

$$IP = \frac{\text{number of users who have rated the item}}{\text{number of users}} \quad (6)$$

An item's novel value (*INV*) is then measurable by taking the log of the inverse *IP*:

▪ Diversity

$$INV = -\log_2(IP) \quad (7)$$

– Intra-List Diversity

- Average similarity of all pairs of recommended items
- Both content-based and collaborative variants are plausible

$$div_{sim}(u) = \frac{\sum_{\forall o_i, o_j \in O_u; i \neq j} 1 - sim(o_i, o_j)}{|O_u| * (|O_u| - 1)}$$

- Subtopics recall, redundancy penalization, intent-aware metrics...

Off-line Evaluation Metrics

- **Coverage**
 - Number of unique items recommended at least once / Total volume of items

- **Popularity bias / popularity lift**
 - A.k.a. Novelty of the whole recommendation process
 - Normalized by the distribution of feedback among items

$$PopLift = \frac{mPop_{rec} - mPop_{data}}{mPop_{data}} \quad (13)$$

The $mPop_{rec}$ and $mPop_{data}$ stands for the mean popularity of items that were recommended and items that occurs in the dataset respectively. Formally, suppose to have a list of positive feedback events in a dataset $f_i(u, o) \in \mathcal{F}^+$. Each event is triggered by a user u on an item o . We can use the notation $o_j \in f_i$ meaning that the item o_j is a target in the event f_i . Then popularity of an item is defined as

$$pop(o_j) = \frac{|\{f_i : o_j \in f_i\}|}{|\mathcal{F}^+|}$$

Now, suppose that O_{rec} contains a concatenated list of all recommendations (irrespective of users) and O_{data} contains a list of target items for all events $f_i(u, o) \in \mathcal{F}^+$. Then

$$mPop_{rec} = \frac{\sum_{o_j \in O_{rec}} pop(o_j)}{|O_{rec}|} \text{ and } mPop_{data} = \frac{\sum_{o_j \in O_{data}} pop(o_j)}{|O_{data}|}.$$

Enhancing novelty / diversity

- **Serendipity**

- = items that are both relevant and unexpected
 - relevance is typically the predicted RS relevance (or actual presence in test data)
 - unexpectedness evaluated as diversity to the w.r.t. current user profile
 - Serendipity = relevance * unexpectedness

Enhancing novelty / diversity

- **Novelty, coverage**

- Evaluated on individual items => simple weighted models will do
- $\text{argmax } \alpha * \text{relevance} + (1 - \alpha) * \text{novelty}$

- **Diversity**

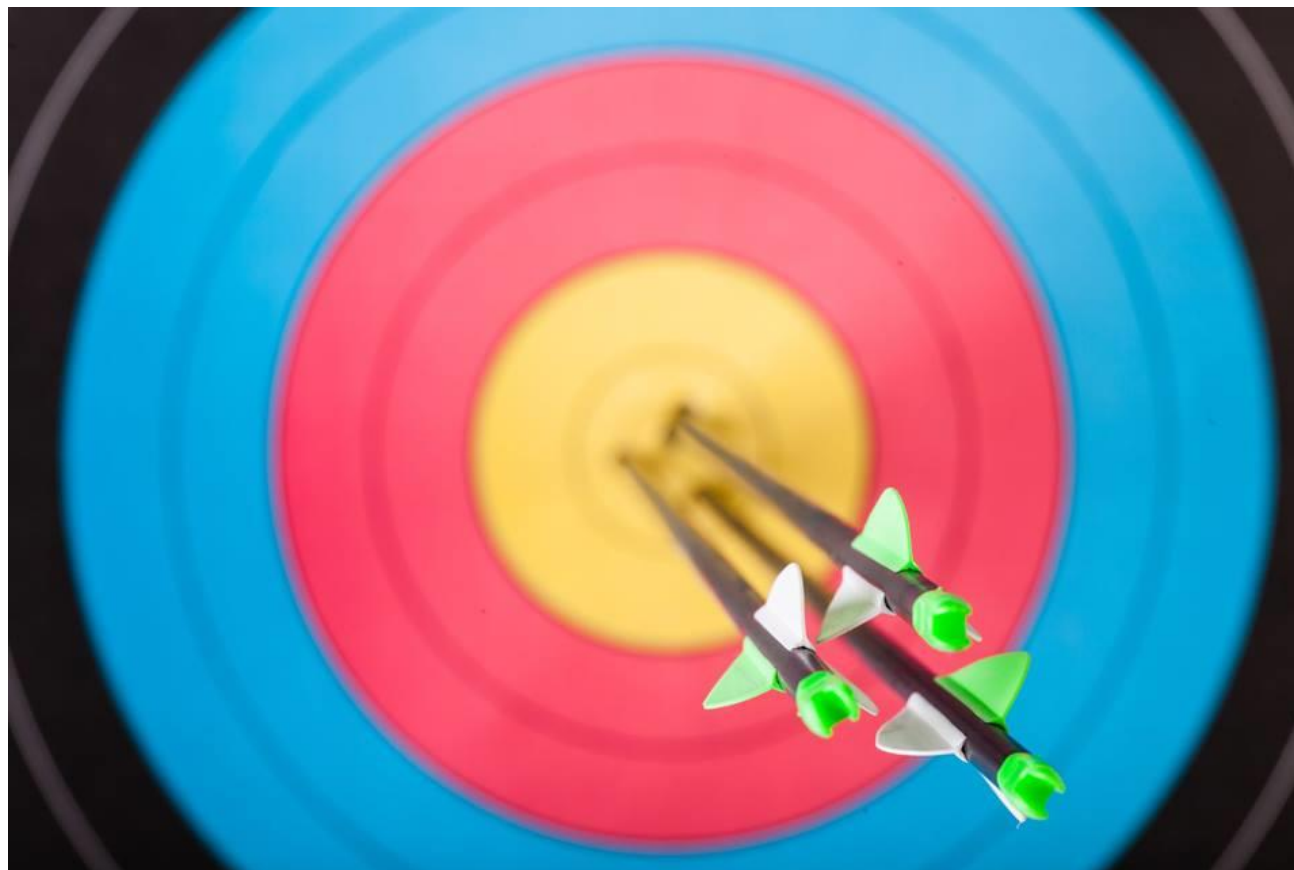
- Diversity is a feature of the list
- **Maximal Marginal Relevance**, xQUAD,...
 - Reduce relevance of the next item by its similarity to the closest already selected item

$$\text{MMR} = \arg \max_{D_i \in R \setminus S} \left[\lambda \text{Sim}_1(D_i, Q) - (1 - \lambda) \max_{D_j \in S} \text{Sim}_2(D_i, D_j) \right]$$



**Estimated relevance
in our case**

Accuracy / relevance metrics



Evaluation in information retrieval (IR)

- **Historical Cranfield collection (late 1950s)**
 - 1,398 journal article abstracts
 - 225 queries
 - Exhaustive relevance judgements (over 300K)
- **Ground truth established by human domain experts**

		Reality	
		Actually Good	Actually Bad
Prediction	Rated Good	True Positive (tp)	False Positive (fp)
	Rated Bad	False Negative (fn)	True Negative (tn)

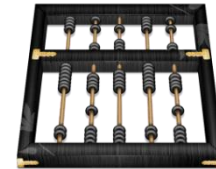
All recommended items

All good items

Metrics: Precision and Recall

- **Recommendation is viewed as information retrieval task:**
 - Retrieve (recommend) all items which are predicted to be “good”.
- **Precision: a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved**
 - E.g. the proportion of recommended movies that are actually good

$$Precision = \frac{tp}{tp + fp} = \frac{|good\ movies\ recommended|}{|all\ recommendations|}$$



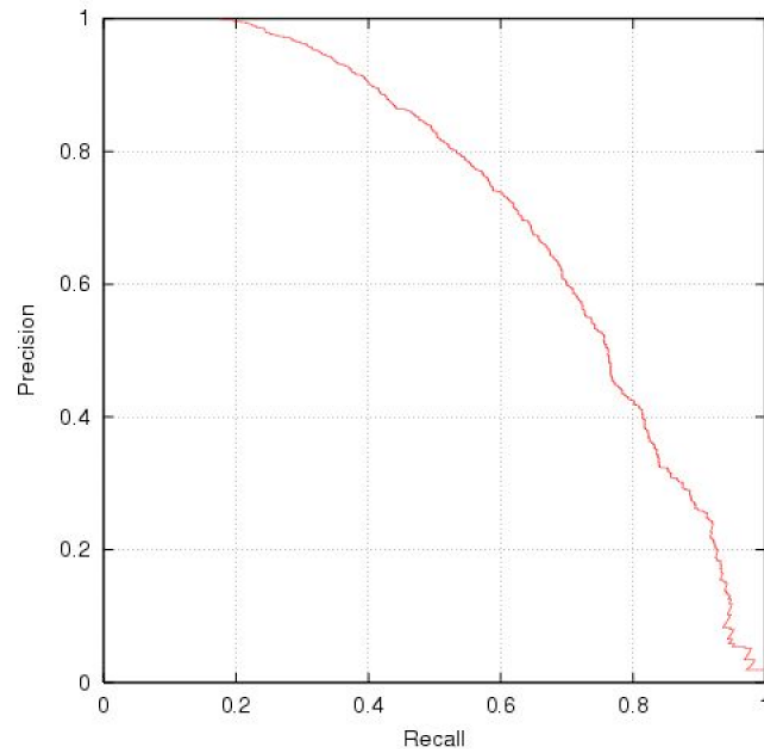
- **Recall: a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items**
 - E.g. the proportion of all good movies recommended

$$Recall = \frac{tp}{tp + fn} = \frac{|good\ movies\ recommended|}{|all\ good\ movies|}$$



Precision vs. Recall

- E.g. typically when a recommender system is tuned to increase precision, recall decreases as a result (or vice versa)
- **AUPR**
Area under
Prec. vs. Recall
- **AUC:**
Area under ROC
(TP vs. FP)

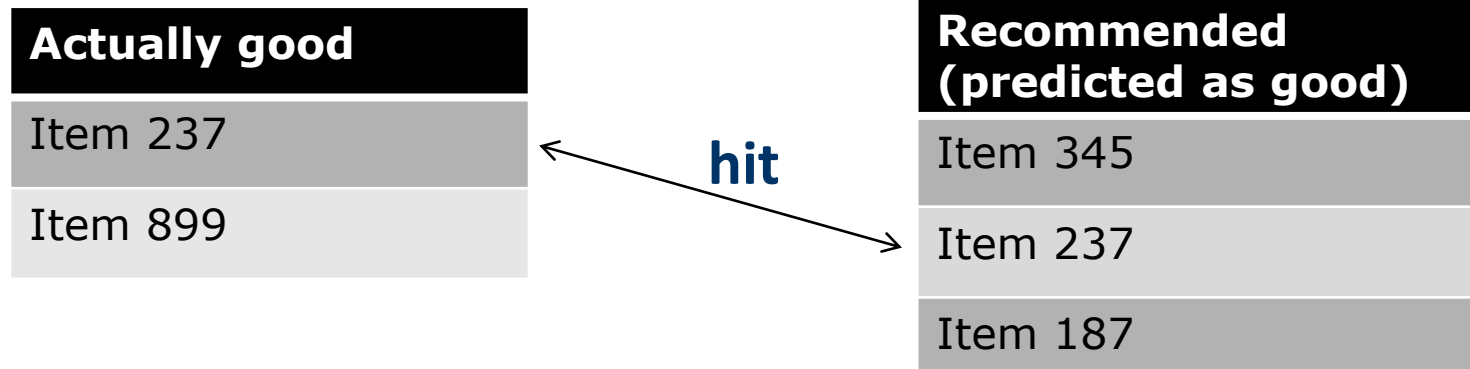


Precision and Recall in Recommender Systems

- **Limit on top-k displayed items (i.e., no longer a classification, but rather a ranking problem)**
 - Hits@top-k
 - Precision@top-k
 - Recall@top-k
 - How to treat recall if only limited items are shown
 - E.g., shown: 20, relevant shown: 20, total relevant: 200 => recall@20 = 0.1 or 1.0?
 - Unbounded/Bounded recall variants (hard to distinguish in papers or frameworks)
- **Position within top-k does not matter**
 - The list is short enough that user observe it all
 - With increasing k, this became less applicable

Metrics: Rank position matters

For a user:



- **Rank metrics**

- Relevant items are more useful when they appear earlier in the recommendation list
 - This might be partially violated e.g. in 2D settings
- Particularly important in recommender systems as lower ranked items may be overlooked by users

Metrics: Rank Score (rarely used)

- Rank Score extends the recall metric to take the positions of correct items in a ranked list into account
 - Particularly important in recommender systems as lower ranked items may be overlooked by users
- Rank Score is defined as the ratio of the Rank Score of the correct items to best theoretical Rank Score achievable for the user, i.e.

$$rankscore = \frac{rankscore_p}{rankscore_{\max}}$$

$$rankscore_p = \sum_{i \in h} 2^{-\frac{rank(i)-1}{\alpha}}$$

$$rankscore_{\max} = \sum_{i=1}^{|T|} 2^{-\frac{i-1}{\alpha}}$$

Where:

- h is the set of correctly recommended items, i.e. hits
- $rank$ returns the position (rank) of an item
- T is the set of all items of interest
- α is the *ranking half life*, i.e. an exponential reduction factor

Metrics: Liftindex (rarely used)

- Assumes that ranked list is divided into 10 equal deciles S_i , where

$$\sum_{i=1}^{10} S_i = |h|$$

- Linear reduction factor

- Liftindex:**

$$liftindex = \begin{cases} \frac{1 \times S_1 + 0.9 \times S_2 + \dots + 0.1 \times S_{10}}{\sum_{i=1}^{10} S_i} & : \text{ if } |h| > 0 \\ 0 & : \text{ else } \end{cases}$$

» h is the set of correct hits

Metrics: Normalized Discounted Cumulative Gain

- **Discounted cumulative gain (DCG)**

- Logarithmic reduction factor

$$DCG_{pos} = rel_1 + \sum_{i=2}^{pos} \frac{rel_i}{\log_2 i}$$

Where:

- pos denotes the position up to which relevance is accumulated
- rel_i returns the relevance of recommendation at position i

- **Idealized discounted cumulative gain (IDCG)**

- Assumption that items are ordered by decreasing relevance

$$IDCG_{pos} = rel_1 + \sum_{i=2}^{|h|-1} \frac{rel_i}{\log_2 i}$$

- **Normalized discounted cumulative gain (nDCG)**

- Normalized to the interval [0..1]

$$nDCG_{pos} = \frac{DCG_{pos}}{IDCG_{pos}}$$

Example

Rank	Hit?
1	
2	X
3	X
4	X
5	

- Assumptions:

- $|T| = 3$
- Ranking half life (alpha) = 2

$$rankscore = \frac{rankscore_p}{rankscore_{\max}} \approx 0.71$$

$$nDCG_5 \frac{DCG_5}{IDCG_5} \approx 0.81$$

$$liftindex = \frac{0.8 \times 1 + 0.6 \times 1 + 0.4 \times 1}{3} = 0.6$$

$$rankscore_p = \frac{1}{2^{\frac{2-1}{2}}} + \frac{1}{2^{\frac{3-1}{2}}} + \frac{1}{2^{\frac{4-1}{2}}} = 1.56$$

$$rankscore_{\max} = \frac{1}{2^{\frac{1-1}{2}}} + \frac{1}{2^{\frac{2-1}{2}}} + \frac{1}{2^{\frac{3-1}{2}}} = 2.21$$

$$DCG_5 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} = 2.13$$

$$IDCG_5 = 1 + \frac{1}{\log_2 2} + \frac{1}{\log_2 3} = 2.63$$

Example cont.

- Reducing the ranking half life (alpha) = 1

Rank	Hit?
1	
2	X
3	X
4	X
5	

$$rankscore = \frac{rankscore_p}{rankscore_{\max}} = 0.5$$

$$rankscore_p = \frac{1}{2^{\frac{1}{1}}} + \frac{1}{2^{\frac{1}{2}}} + \frac{1}{2^{\frac{1}{3}}} = 0.875$$
$$rankscore_{\max} = \frac{1}{2^{\frac{1}{1}}} + \frac{1}{2^{\frac{1}{2}}} + \frac{1}{2^{\frac{1}{3}}} = 1.75$$

Rankscore (exponential reduction) < Liftscore (linear red.) < NDCG (log. red.)

Average Precision

- Mean Average Precision (*MAP*) is a ranked precision metric that places emphasis on highly ranked correct predictions (hits)
- Essentially it is the average of precision values determined after each successful prediction, i.e.

Rank	Hit?
1	
2	X
3	X
4	X
5	

$$AP = \frac{1}{3} \left(\frac{1}{1} + \frac{2}{4} + \frac{3}{5} \right) = \frac{21}{30} = 0.7$$

Rank	Hit?
1	X
2	
3	
4	X
5	X

$$AP = \frac{1}{3} \left(\frac{1}{2} + \frac{2}{3} + \frac{3}{4} \right) = \frac{23}{36} \approx 0.639$$

**All relevant
items**

Metrics: Mean average precision

Average Precision

OK are you ready for Average Precision now? If we are asked to recommend N items, the number of relevant items in the full space of items is m , then:

$$AP@N = \frac{1}{m} \sum_{k=1}^N (P(k) \text{ if } k^{th} \text{ item was relevant}) = \frac{1}{m} \sum_{k=1}^N P(k) \cdot rel(k),$$

where $rel(k)$ is just an indicator that says whether that k^{th} item was relevant ($rel(k) = 1$) or not ($rel(k) = 0$). I'd like to point out that instead of recommending N items would could have recommended, say, $2N$, but the AP@N metric says we only care about the average precision up to the N^{th} item.

Examples and Intuition for AP

Let's imagine recommending $N = 3$ products (AP@3) to a user who actually added a total of $m = 3$ products. Here are some examples of outcomes for our algorithm:

Recommendations	Precision @k's	AP@3
[0, 0, 1]	[0, 0, 1/3]	$(1/3)(1/3) = 0.11$
[0, 1, 1]	[0, 1/2, 2/3]	$(1/3)[(1/2) + (2/3)] = 0.38$
[1, 1, 1]	[1/1, 2/2, 3/3]	$(1/3)[(1) + (2/2) + (3/3)] = 1$

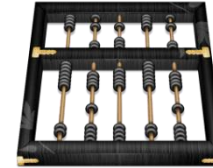


Questions?

Evaluation in RS – rating based

- **Datasets with items rated by users**

- MovieLens datasets 100K-10M ratings
- Netflix 100M ratings



- **Historic user ratings constitute ground truth**

- **Metrics measure error rate**

- Mean Absolute Error (*MAE*) computes the deviation between predicted ratings and actual ratings

$$MAE = \frac{1}{n} \sum_{i=1}^n |p_i - r_i|$$

- Root Mean Square Error (*RMSE*) is similar to *MAE*, but places more emphasis on larger deviation

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - r_i)^2}$$

De-biasing Off-line Evaluation

■ What are the biases in our data?

– Presentation bias

researchgate.net/publication/2961924_Search_Engines_that_Learn_from_Implicit_Feedback

– ... <https://www.youtube.com/watch?v=brr8cZHf2ec> (RecSys 2020 keynote)

Dependencies: A Cascade of Biases!

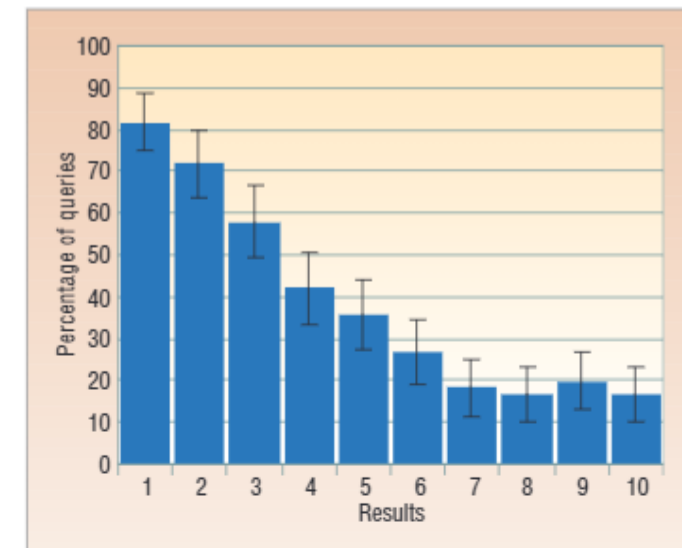
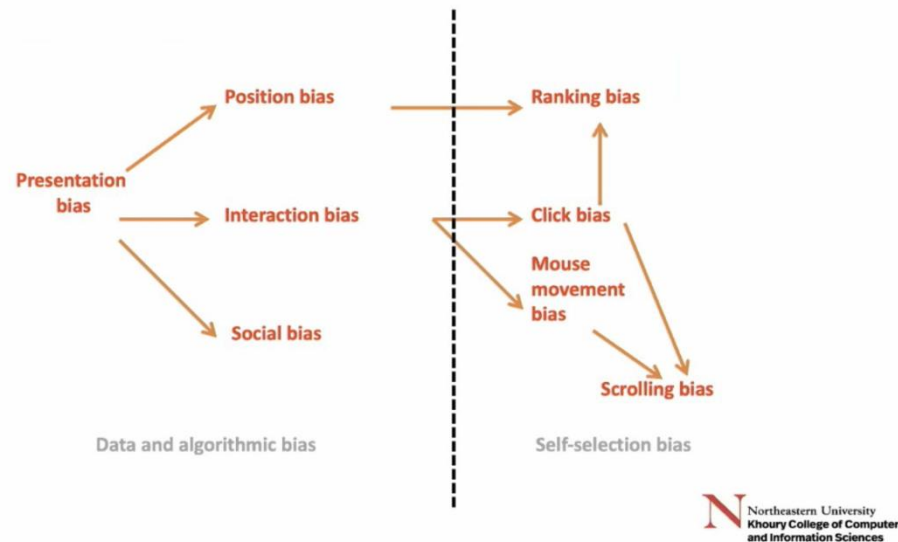


Figure 2. Rank and viewership. Percentage of queries where a user viewed the search result presented at a particular rank.

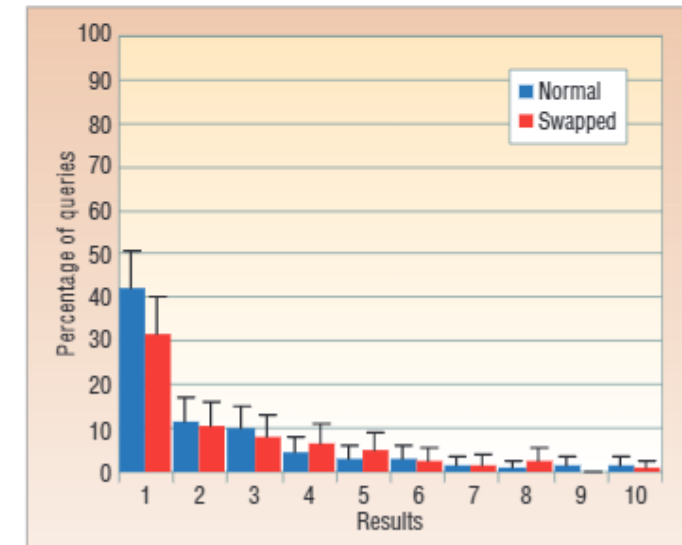


Figure 3. Swapped results. Percentage of queries where a user clicked the result presented at a given rank, both in the normal and swapped conditions.

De-biasing Off-line Evaluation

- **Presentation bias**
 - What (where) was shown to the user affects what feedback we received
 - How to evaluate novel RS that would recommend something else
 - Than what users received & evaluated
 - Hard question in general, intensive research topic
 - *Off-policy evaluation* or *counterfactual evaluation* (what would be the results, if policy B was applied)
 - Part of the feedback received on random(ized) data
 - <https://arxiv.org/pdf/1003.0146.pdf> (Section 4: evaluation, assuming independence of events)

De-biasing Off-line Evaluation

- **Missing not at random for Implicit feedback**

- Classical off-line evaluation expects Missing at random data
- absence of (positive) feedback is either because the item is **unknown positive** or **irrelevant**
 - For aggregational statistics to be valid, it is important that the chance of being unknown positive or irrelevant is the same for all data with no feedback
 - **Not true in real-world for some cases**
 - *Probability that the item is known by the user => propensity*
 - Well-known movies may have higher propensity (users ignore them willingly)
 - MISSING NOT AT RANDOM
 - Evaluation should be de-biased accordingly (we need to estimate the propensity)

Thorsten Joachims et al. <https://arxiv.org/pdf/1608.04468.pdf>;
<https://ylongqi.com/paper/YangCXWBE18.pdf>

De-biasing Off-line Evaluation

■ <https://arxiv.org/pdf/1608.04468.pdf>

For concreteness and simplicity, assume that relevances are binary, $r_i(y) \in \{0, 1\}$, and our performance measure of interest is the sum of the ranks of the relevant results

$$\Delta(\mathbf{y}|\mathbf{x}_i, \mathbf{r}_i) = \sum_{y \in \mathbf{y}} \text{rank}(y|\mathbf{y}) \cdot r_i(y). \quad (2)$$

Analogous to (1), we can define the risk of a system as

$$R(S) = \int \Delta(S(\mathbf{x})|\mathbf{x}, \mathbf{r}) dP(\mathbf{x}, \mathbf{r}). \quad (3)$$

In our counterfactual model, there exists a true vector of relevances \mathbf{r}_i for each incoming query instance $(\mathbf{x}_i, \mathbf{r}_i) \sim P(\mathbf{x}, \mathbf{r})$. However, only a part of these relevances is observed for each query instance, while typically most remain unob-

erved. Let \mathbf{o}_i denote the 0/1 vector indicating which relevance values were revealed, $\mathbf{o}_i \sim P(\mathbf{o}|\mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i)$. For each element of \mathbf{o}_i , denote with $Q(\mathbf{o}_i(y) = 1|\mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i)$ the marginal probability of observing the relevance $r_i(y)$ of result y for query \mathbf{x}_i , if the user was presented the ranking \mathbf{y}_i . We refer to this

$$\begin{aligned} \hat{\Delta}_{IPS}(\mathbf{y}|\mathbf{x}_i, \mathbf{y}_i, \mathbf{o}_i) &= \sum_{y: \mathbf{o}_i(y)=1} \frac{\text{rank}(y|\mathbf{y}) \cdot r_i(y)}{Q(\mathbf{o}_i(y)=1|\mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i)} \\ &= \sum_{y: \mathbf{o}_i(y)=1} \frac{\text{rank}(y|\mathbf{y})}{Q(\mathbf{o}_i(y)=1|\mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i)}. \end{aligned}$$

This is an unbiased estimate of $\Delta(\mathbf{y}|\mathbf{x}_i, \mathbf{r}_i)$ for any \mathbf{y} , if $Q(\mathbf{o}_i(y) = 1|\mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i) > 0$ for all y that are relevant $r_i(y) = 1$ (but not necessarily for the irrelevant y).

$$\begin{aligned} \mathbb{E}_{\mathbf{o}_i}[\hat{\Delta}_{IPS}(\mathbf{y}|\mathbf{x}_i, \mathbf{y}_i, \mathbf{o}_i)] &= \mathbb{E}_{\mathbf{o}_i} \left[\sum_{y: \mathbf{o}_i(y)=1} \frac{\text{rank}(y|\mathbf{y}) \cdot r_i(y)}{Q(\mathbf{o}_i(y)=1|\mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i)} \right] \\ &= \sum_{y \in \mathbf{y}} \mathbb{E}_{\mathbf{o}_i} \left[\frac{\mathbf{o}_i(y) \cdot \text{rank}(y|\mathbf{y}) \cdot r_i(y)}{Q(\mathbf{o}_i(y)=1|\mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i)} \right] \\ &= \sum_{y \in \mathbf{y}} \frac{Q(\mathbf{o}_i(y) = 1|\mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i) \cdot \text{rank}(y|\mathbf{y}) \cdot r_i(y)}{Q(\mathbf{o}_i(y) = 1|\mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i)} \\ &= \sum_{y \in \mathbf{y}} \text{rank}(y|\mathbf{y}) r_i(y) \\ &= \Delta(\mathbf{y}|\mathbf{x}_i, \mathbf{r}_i). \end{aligned}$$

5.1 Position-Based Propensity Model

Search engine click logs provide a sample of query instances \mathbf{x}_i , the presented ranking \mathbf{y}_i and a (sparse) click-vector where each $c_i(y) \in \{0, 1\}$ indicates whether result y was clicked or not. To derive propensities of observed clicks, we will employ a click propensity model. For simplicity, we consider a straightforward examination model analogous to [17], where a click on a search result depends on the probability that a user examines a result (i.e., $e_i(y)$) and then decides to click on it (i.e., $c_i(y)$) in the following way:

$$P(c_i(y) = 1|\mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i) = P(e_i(y) = 1|\mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i) \cdot P(c_i(y) = 1|e_i(y) = 1, \mathbf{x}_i, \mathbf{y}_i, \mathbf{r}_i).$$

De-biasing Off-line Evaluation

■ <https://yongqi.com/paper/YangCXWBE18.pdf>

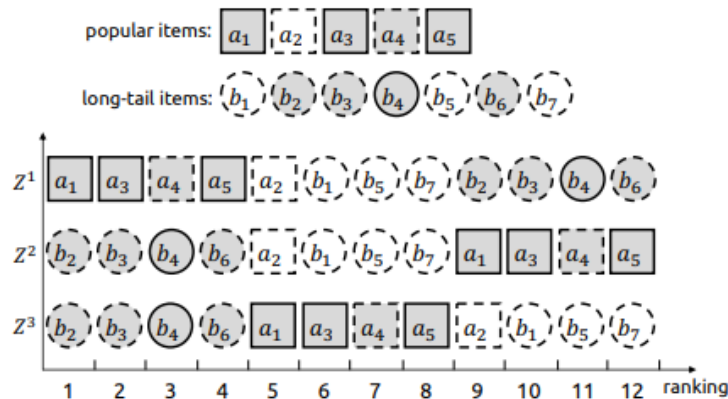


Figure 1: A hypothetical example to illustrate the evaluation bias that results from use of the AOA evaluator. Three recommenders generated distinct lists of recommendations, Z^1 , Z^2 and Z^3 , for the same user. Among the shaded items that were preferred by the user, the ones with a solid border were observed by recommenders. The performance was measured by DCG, and the results are presented in Table 1.

Table 1: The true and estimated DCG values for three recommenders in Fig. 1. $R(\hat{Z})$ denotes the ground truth, and $\hat{R}_{\text{AOA}}(\hat{Z})$ denotes the AOA estimations. The AOA estimator outputs larger values when popular items are ranked higher.

Estimator	Z^1	Z^2	Z^3
$R(\hat{Z})$	0.463	0.463	0.494
$\hat{R}_{\text{AOA}}(\hat{Z})$	0.585	0.340	0.390

3.1 Average-over-all (AOA) evaluator

In prior literature, $R(\hat{Z})$ was estimated by taking the average over all observed user feedback S_u^* :

$$\begin{aligned}\hat{R}_{\text{AOA}}(\hat{Z}) &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|S_u^*|} \sum_{i \in S_u^*} c(\hat{Z}_{u,i}) \\ &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{\sum_{i \in S_u} O_{u,i}} \sum_{i \in S_u} c(\hat{Z}_{u,i}) \cdot O_{u,i}\end{aligned}\quad (6)$$

nDCG, AUC, MAP,...

3.2 Unbiased evaluator

To conduct unbiased evaluation of biased observations, we leverage the IPS framework [16, 22] that weights each observation with the inverse of its propensity, where the term *propensity* refers to the tendency or the likelihood of an event happening. The intuition is to down-weight the commonly observed interactions, while up-weighting the rare ones. In the context of this paper, the probability $P_{u,i}$ is treated as the pointwise propensity score. Therefore, the IPS unbiased evaluator is defined as follows:

$$\begin{aligned}\hat{R}_{\text{IPS}}(\hat{Z}|P) &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|S_u|} \sum_{i \in S_u} \frac{c(\hat{Z}_{u,i})}{P_{u,i}} \\ &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|S_u|} \sum_{i \in S_u} \frac{c(\hat{Z}_{u,i})}{P_{u,i}} \cdot O_{u,i}\end{aligned}\quad (7)$$

Propensity score

$$\hat{P}_{*,i} \propto (n_i^*)^\gamma \cdot n_i,$$

where $n_i = \sum_{u \in \mathcal{U}} \mathbf{1}[i \in S_u]$ and $n_i^* = \sum_{u \in \mathcal{U}, i \in S_u^*} O_{u,i}$.

However, empirically, n_i is not directly observable. To solve this problem, we observe that n_i^* is sampled from a binomial distribution⁴ parameterized by n_i , that is, $n_i^* \sim \mathcal{B}(n_i, P_{*,i})$. The relationship between n_i and n_i^* can be built by bridging the generative model (eqn. 13) with the following unbiased estimator

$$\hat{P}_{*,i} = \frac{n_i^*}{n_i} \propto (n_i^*)^\gamma \cdot n_i$$

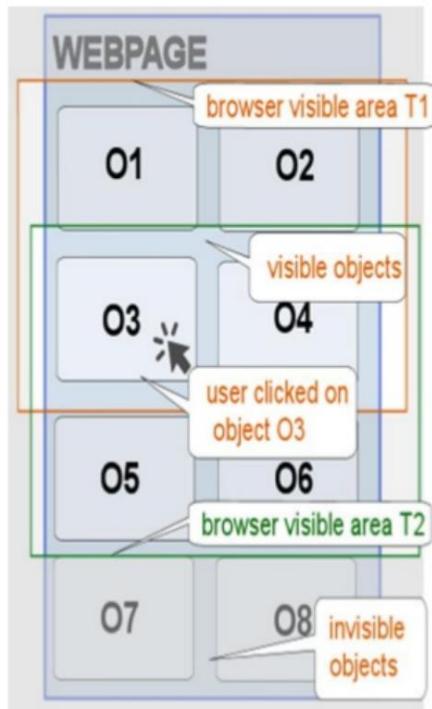
We use this as a replacement for $P_{*,i}$ in the IPS framework. This is an unbiased $\hat{P}_{*,i}$ that is determined by only the empirical counts of items.

$$\hat{P}_{*,i} \propto (n_i^*)^{\left(\frac{\gamma+1}{2}\right)}$$

De-biasing Off-line Evaluation (more thorough implicit feedback data)

- https://www.researchgate.net/publication/294139826_Using_Implicit_Preference_Relations_to_Improve_Recommender_Systems
- Used for pairwise relevance relations, but suitable for propensity scores as well

Collecting User Behavior – Example



- The list contains 8 objects
 - O1-O4 were visible for time T1, O3-O6 were visible for time T2
 - O7 and O8 were invisible all the time and thus do not form IPR relation
- User clicked on O3, thus O3 is „better“ than O1-O6
 - Intensity of IPR relation is based on how well was *ignored* objects evaluated

$$IPR_{rel}(\text{selected}, \text{ignored}, \text{intensity})$$
$$\text{intensity} := \min\left(\frac{\text{eval}(\text{ignored})}{\text{eval}(\text{selected})}, 1\right)$$

Additional Slides