Evaluating Recommender Systems

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

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?
  - Comparative analysis based on an optimality criterion?

- **Research questions are:**
  - Is a RS efficient with respect to a specific criteria like accuracy, user satisfaction, response time, serendipity, online conversion, ramp-up efforts, ….
  - Do customers like/buy recommended items?
  - Do customers buy items they otherwise would have not?
  - Are they satisfied with a recommendation after purchase?

  (Can we assure that this improvement was caused by the RS?)

Empirical research

- **Characterizing dimensions:**
  - Who is the **subject** that is in the focus of research?
  - What **research method**s are applied?
  - In which **setting** does the research take place?

Subject	Online customers, students, historical online sessions, computers, …
Research method	Experiments, quasi-experiments, non-experimental research
Setting	Lab, real-world scenarios, off-line (data) study

Evaluation settings

- **Off-line evaluation**
  - Based on historical data
  - Aiming to predict hidden part of the data
- **Lab studies**
  - Expressly created for the purpose of the study
  - Extraneous variables can be controlled more easy by selecting study participants
    - Possibility to get more feedback
  - But doubts may exist about participants motivated by money or prizes
  - **Participants should behave as they would in a real-world enviroment**
- **Field studies (On-line, A/B testing)**
  - Conducted in an preexisting real-world enviroment
  - Users are intrinsically motivated to use a system

Varianty Evaluace

Online

- Na běžícím serveru
- Těžko se opakuje
- Náročné (časově i finančně)
- Pouze několik metod
- Skutečné metriky (CTR, konverze…)
- Lze měřit i změny GUI atp.

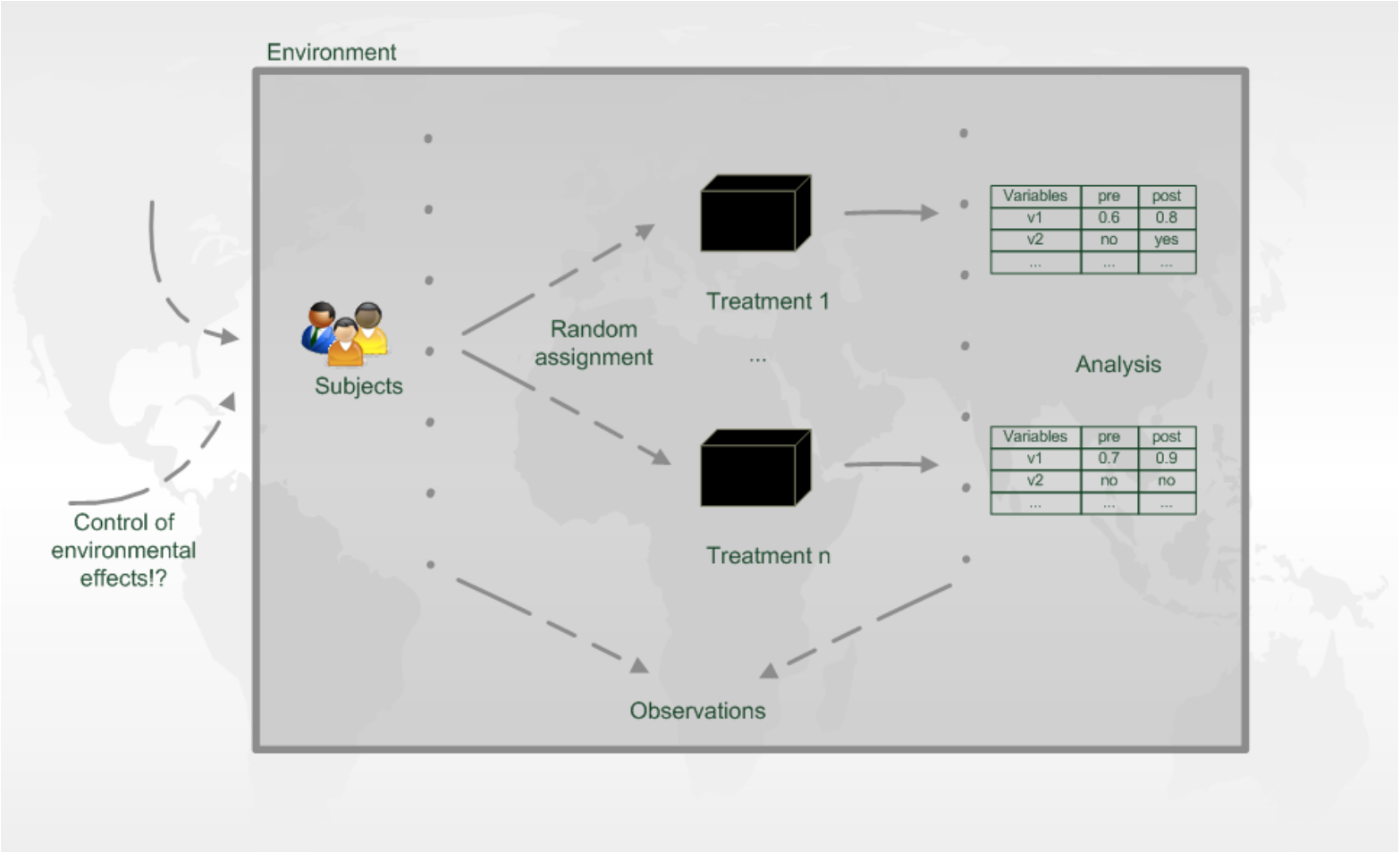
Offline

- Datová simulace
- Snadno se opakuje
- Výsledky (poměrně) rychle
- Umělé metriky (RMSE, nDCG, diversity…)
- Dovedeme pouze porovnávat schopnost predikce minulého chování uživatele

Success in offline do not imply success in online…

…ale pokud metoda neuspěje v offline, obvykle nemá cenu jí zkoušet online.

Experiment designs – Online Evaluation, A/B testing



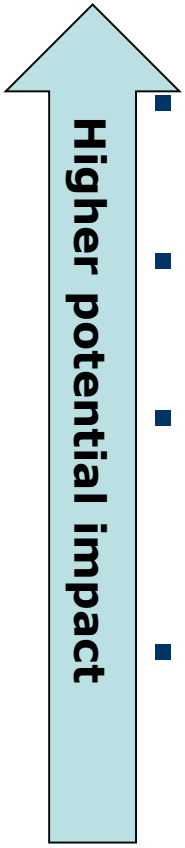
Evaluation of Online studies

- **A/B testing**
  - ***Evaluate metric as close to the actual target variable as possible***
  - Retailer's target variable is profit
    - i.e. Netflix's target variable is monthly subscribes
    - Usually, larger overal consumption increase profit
  - Broadcaster's target variable may be influenceness / total mass of readers
- **The direct effect on target variables 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 /…

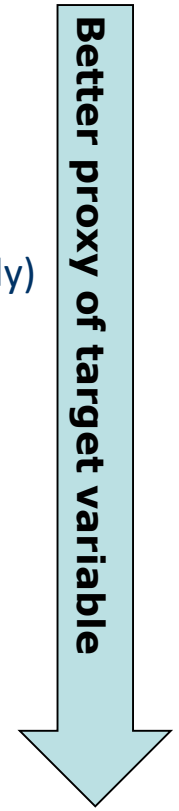
Common on-line evaluation metrics – E-commerce

- ***Always design evaluation metrics with respect to your target variable***
  - However select something, where the effect is measurable
  - Cascade of evaluation metrics
    - From high to low detectability of changes
    - From low to high impact on your true target

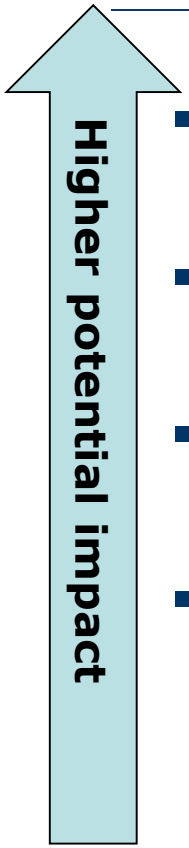
Common on-line evaluation metrics – E-commerce



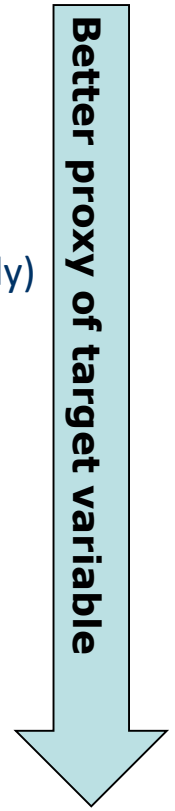
- **Recommending correctness**
  - Visit (once) recommended object (i.e. ignore page layout)
- **Click-through rate**
  - Click on recommended item / Click and do something (do not leave imediatelly)
- **Conversions rate**
  - Buy recommended item / Recommend -> Click -> Purchase
  - Share / follow / like / ask about... recommended item
- **Cross-sale increase**
  - Add to cart -> Recommend -> Add another (recommended) to cart



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



- **Recommending correctness**
  - Visit (once) recommended object (i.e. ignore page layout)
- **Click-through rate**
  - Click on recommended item / Click and do something (do not leave imediatelly)
- **„Conversions“ rate**
  - Share / follow / like / comment... recommended item
- **Value per user**
  - Total time / number of visited objects / displayed adds… per user or per session
  - Returning rate of users / user loyalty



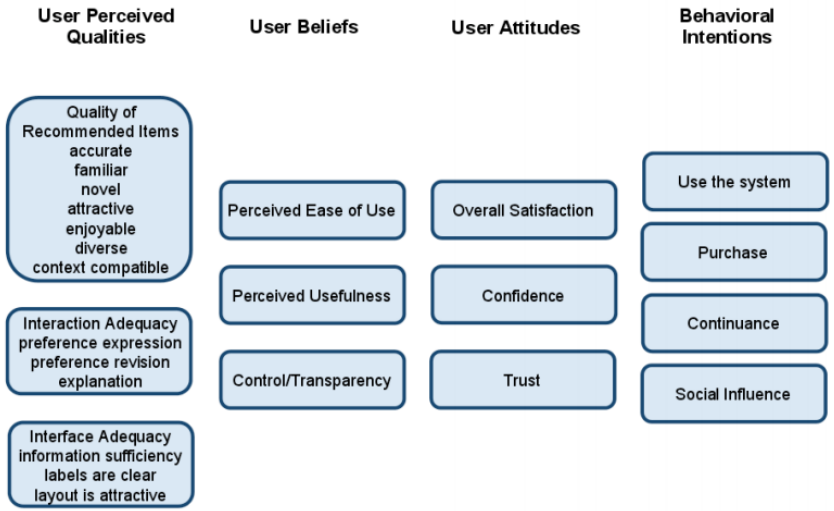
Common on-line evaluation metrics – Technical

- **!!Response time!!**
- **Train / re-train time**
- **Memory / CPU 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?

Evaluation in Lab studies

- **Same as on-line experiments**
- **Questionnaire**
  - Features otherwise harder to detect directly
  - Helpfulness / Ease of use / Relevance
  - Trust
  - Novelty to the user etc.
- **Physiological response**
  - Eye tracking etc.



- ***Key criterion in lab studies is that subjects should well approximate behavior of your real users***
  - This may be harder than it seems

Evaluation of Off-line experiments

- **Simulation on existing dataset**
  - Train / Validation / Test split
    - Random (bootstrap) – only in case of very large datasets
    - Cross-validation variants
    - **Temporal splits** – better than CV for RecSys (causality problems), however lower support in non-recsys audience
    - **Event-based simulation** – the best option from causality perspective, most expensive
- **Prediction of „correct“ objects**
  - According to some metic / metrics

Evaluation of Off-line experiments

- **Simulation on existing dataset**

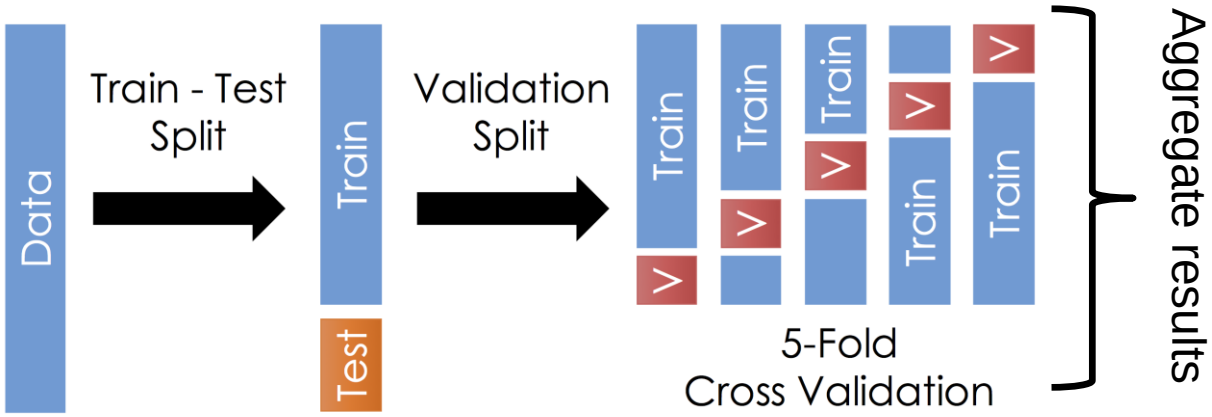
  Correct evaluation protocol:

  - For each method and set of parameters:
    - Learn model on TRAIN set
    - Evaluate prediction on VALIDATION set
  - Select best parameters for each method
  - For each method:
    - Learn model on TRAIN + VALIDATION set
    - Evaluate prediction on TEST set
  - Compare results

- **Never use any knowledge of the test set data**
  - E.g. For mean ratings, object similarities etc.

Evaluation of Off-line experiments

- **Simulation on existing dataset: cross-validation**



- **Istead of Train – Test split, you may use additional „outer“ cross-validation**
  - Get results from all parts of the dataset
- **Never use any knowledge of the test set data**
  - E.g. For mean ratings, object similarities etc

Off-line Evaluation Metrics

- **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**
  - User rated some objects, how large is the prediction error on those?
  - MAE, RMSE,…
- **Novelty**
  - Does the user already know / visited recommended objects?
  - This may be both positive and negative depending on task
    - However it is always trivial
    - No need of complex system to recommend previously visited objects
- **Diversity**
  - Are all the recommendations similar to each other?
  - Relevance vs. Diversity tradeoff
  - **Intra-List Diversity**

Off-line Evaluation Metrics

- **Novelty**
  - Items known (has feedback) by the user
  - Well known items (blockbusters in movies/books), based on overal consumption
  - Items that are new (have been added recently)

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

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

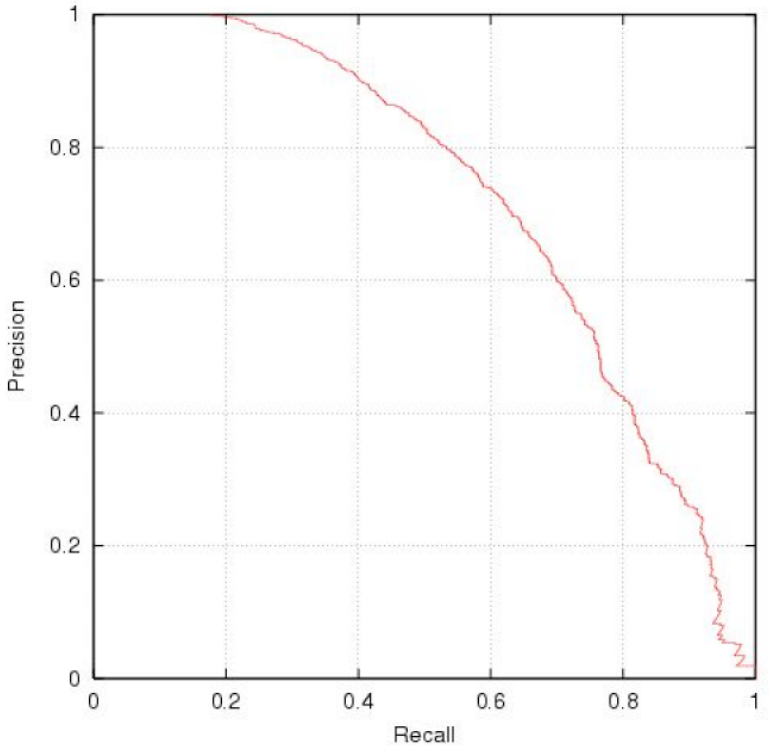
$$Precision = \frac{tp}{tp + fp} = \frac{|good\ movies\ recommended|}{|\text{all recommendations}|}$$

- **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)



F_1 Metric

- **The F_1 Metric attempts to combine Precision and Recall into a single value for comparison purposes.**
  - May be used to gain a more balanced view of performance

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

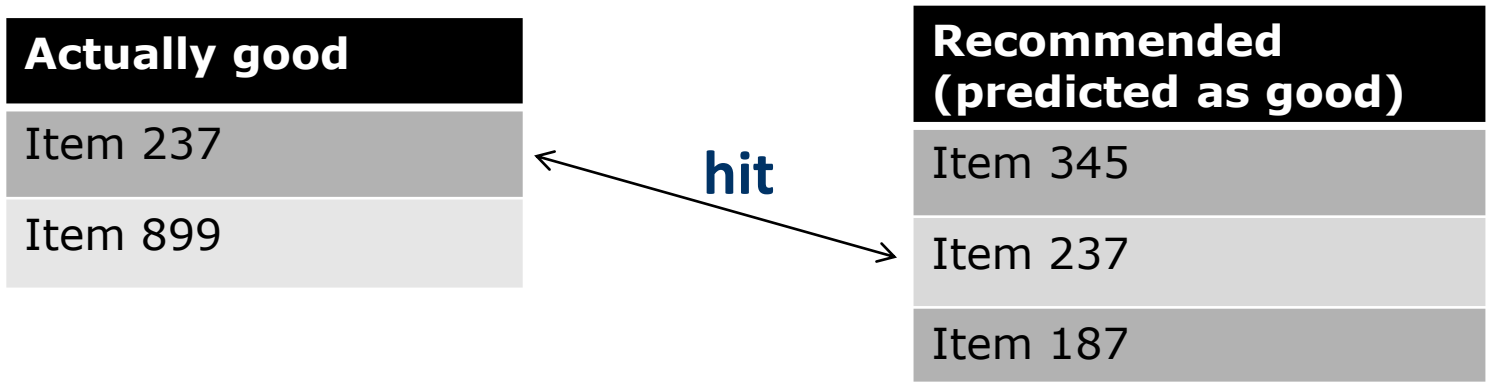
- **The F_1 Metric gives equal weight to precision and recall**
  - Other F_β metrics weight recall with a factor of β.

Precision and Recall in Recommender Systems

- **Limit on top-k**
  - **Precision@top-k**
  - Recall@top-k
- **Position within top-k does not matter**
  - The list is short enough that user observe it all
  - With increasing k, this becames less applicable

Metrics: Rank position matters

For a user:



- **Rank metrics extend recall and precision to take the positions of correct items in a ranked list into account**
  - Relevant items are more useful when they appear earlier in the recommendation list
  - Particularly important in recommender systems as lower ranked items may be overlooked by users

Metrics: Rank Score

- **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

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

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

  - Linear reduction factor

- **Liftindex:**

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

» *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)** $nDCG_{pos} \frac{DCG_{pos}}{IDCG_{pos}}$
  - Normalized to the interval [0..1]

Example

Rank	Hit?
1	
2	X
3	X
4	X
5	

- **Assumptions:**
  - |T| = 3
  - Ranking half life (alpha) = 2

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

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

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

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

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

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

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

Example cont.

- **Reducing the ranking half life (alpha) = 1**

Rank	Hit?
1	
2	X
3	X
4	X
5	

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

$$rankscore_p = \frac{1}{2^{\frac{2-1}{1}}} + \frac{1}{2^{\frac{3-1}{1}}} + \frac{1}{2^{\frac{4-1}{1}}} = 0.875$$

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

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	

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

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

Rank	Hit?
1	X
2	
3	
4	X
5	X

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:

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

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

Evaluation in RS – rating based

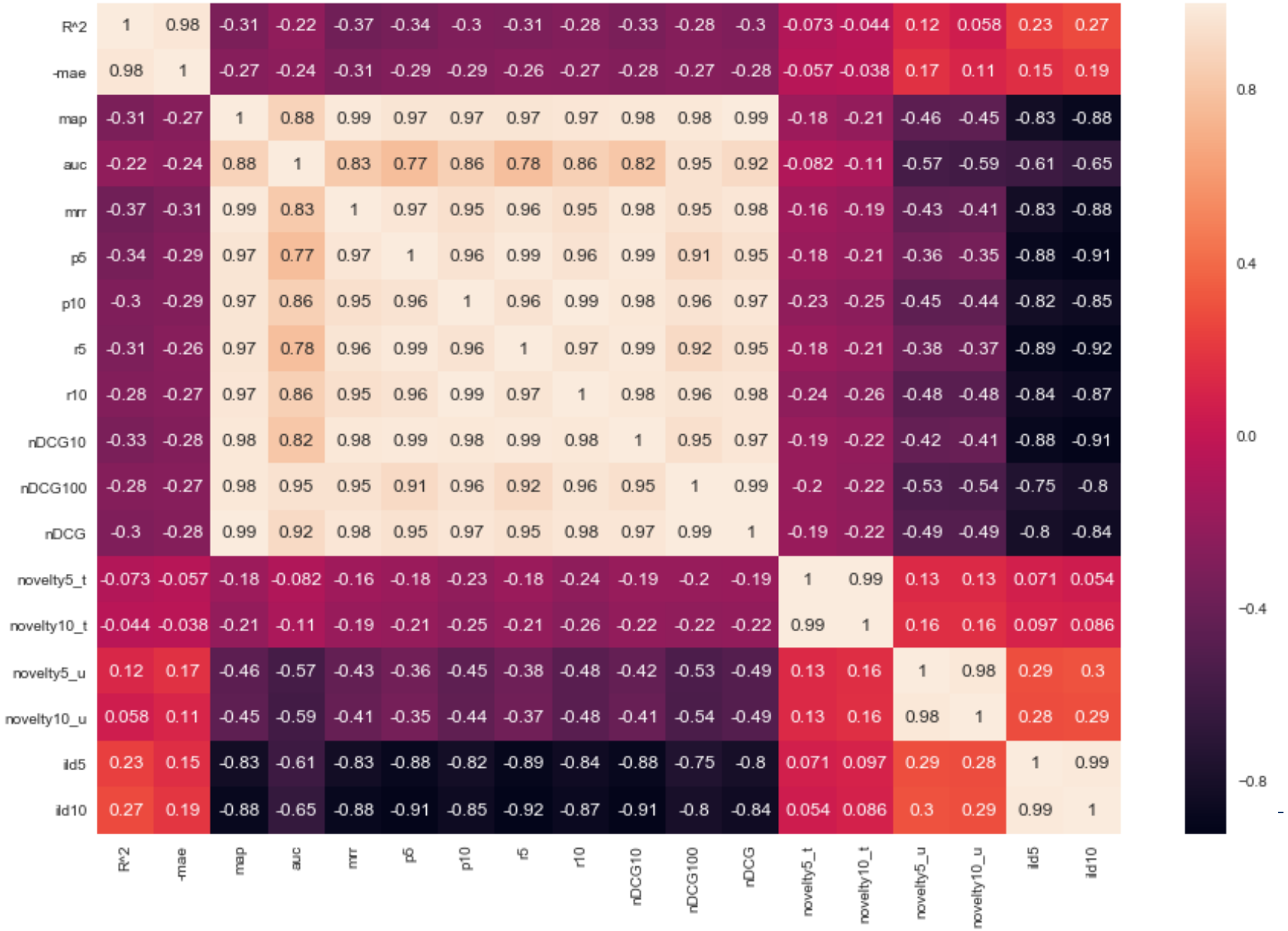
- **Datasets with items rated by users**
  - MovieLens datasets 100K-10M ratings
  - Netflix 100M ratings
- **Historic user ratings constitute ground truth**
- **Metrics measure error rate**
  - Mean Absolute Error (*MAE*) computes the deviation between predicted ratings and actual ratings

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

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

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

Metrics: Comparison



Dilemma of establishing ground truth

- **IR measures are frequently applied, however**:

Offline experimentation	Online experimentation
Ratings, transactions	Ratings, feedback
Historic session (not all recommended items are rated)	Live interaction (all recommended items are rated)
Ratings of unrated items unknown, but interpreted as "bad" (default assumption, user tend to rate only good items)	"Good/bad" ratings of not recommended items are unknown
If default assumption does not hold: True positives may be too small False negatives may be too small	False/true negatives cannot be determined
Precision may increase Recall may vary	Precision ok Recall questionable

Results from offline experimentation have limited predictive power for online user behavior.

Offline experimentation

- **Netflix competition 2004 – 2007?**
  - Web-based movie rental
  - Prize of $1,000,000 for accuracy improvement (RMSE) of 10% compared to own Cinematch system.

- **Historical dataset**
  - ~480K users rated ~18K movies on a scale of 1 to 5
  - ~100M ratings
  - Last 9 ratings/user withheld
    - Probe set – for teams for evaluation
    - Quiz set – evaluates teams' submissions for leaderboard
    - Test set – used by Netflix to determine winner

Methodology

- **Setting to ensure internal validity:**
  - One randomly selected share of known ratings (**training set**) used as input to train the algorithm and build the model
  - Model allows the system to compute recommendations at runtime
  - Remaining share of withheld ratings (**testing set**) required as ground truth to evaluate the model's quality
  - To ensure the reliability of measurements the random split, model building and evaluation steps are repeated several times
- **N-fold cross validation is a stratified random selection procedure**
  - N disjunct fractions of known ratings with equal size (1/N) are determined
  - N repetitions of the model building and evaluation steps, where each fraction is used exactly once as a testing set while the other fractions are used for training
  - Setting N to 5 or 10 is popular

Analysis of results

- **Are observed differences statistically meaningful or due to chance?**
  - Standard procedure for testing the statistical significance of two deviating metrics is the pairwise analysis of variance (ANOVA)
  - Null hypothesis H_0: observed differences have been due to chance
  - If outcome of test statistics rejects H_0, significance of findings can be reported

- **Practical importance of differences?**
  - Size of the effect and its practical impact
  - External validity or generalizability of the observed effects

Online experimentation

- Effectiveness of different algorithms for recommending cell phone games
  [Jannach, Hegelich 09]
- Involved 150,000 users on a commercial mobile internet portal
- Comparison of recommender methods
- Random assignment of users to a specific method

Experimental Design

- **A representative sample 155,000 customers were extracted from visitors to site during the evaluation period**
  - These were split into 6 groups of approximately 22,300 customers
  - Care was taken to ensure that customer profiles contained enough information (ratings) for all variants to make a recommendation
  - Groups were chosen to represent similar customer segments
- **A catalog of 1,000 games was offered**
- **A five-point ratings scale ranging from -2 to +2 was used to rate items**
  - Due to the low number of explicit ratings, a click on the “details” link for a game was interpreted as an implicit “0” rating and a purchase as a “1” rating
- **Hypotheses on personalized vs. non-personalized recommendation techniques and their potential to**
  - Increase conversion rate (i.e. the share of users who become buyers)
  - Stimulate additional purchases (i.e. increase the average shopping basket size)

Non-experimental research

- **Quasi-experiments**
  - Lack random assignments of units to different treatments
- **Non-experimental / observational research**
  - Surveys / Questionnaires
  - Longitudinal research
    - Observations over long period of time
    - E.g. customer life-time value, returning customers
  - Case studies
    - Focus on answering research questions about how and why
    - E.g. answer questions like: *How recommendation technology contributed to Amazon.com's becomes the world's largest book retailer?*
  - Focus group
    - Interviews
    - Think aloud protocols

Data sparsity

- **Natural datasets include historical interaction records of real users**
  - Explicit user ratings
  - Datasets extracted from web server logs (implicit user feedback)

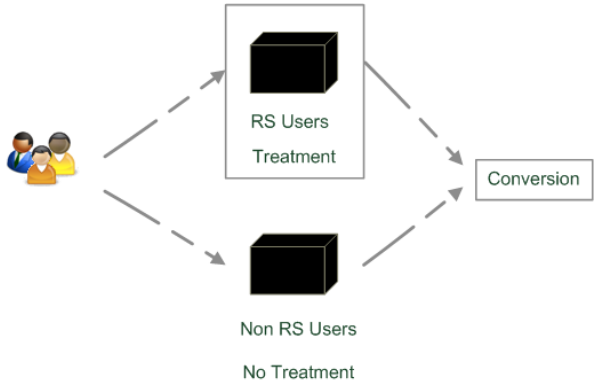
- **Sparsity of a dataset is derived from ratio of empty and total entries in the user-item matrix:**
  - Sparsity = $1 - |R|/|I| \cdot |U|$
  - R = ratings
  - I = items
  - U = users

Quasi-experimental

- **SkiMatcher Resort Finder introduced by Ski-Europe.com to provide users with recommendations based on their preferences**

- **Conversational RS**
  - question and answer dialog
  - matching of user preferences with knowledge base

- **Delgado and Davidson evaluated the effectiveness of the recommender over a 4 month period in 2001**
  - Classified as a quasi-experiment as users decide for themselves if they want to use the recommender or not



SkiMatcher Results

	July	August	September	October
Unique Visitors	10,714	15,560	18,317	24,416
• SkiMatcher Users	1,027	1,673	1,878	2,558
• Non-SkiMatcher Users	9,687	13,887	16,439	21,858
Requests for Proposals	272	506	445	641
• SkiMatcher Users	75	143	161	229
• Non-SkiMatcher Users	197	363	284	412
Conversion	2.54%	3.25%	2.43%	2.63%
• SkiMatcher Users	7.30%	8.55%	8.57%	8.95%
• Non-SkiMatcher Users	2.03%	2.61%	1.73%	1.88%
Increase in Conversion	359%	327%	496%	475%

[Delgado and Davidson, ENTER 2002]

Interpreting the Results

- **The nature of this research design means that questions of causality cannot be answered (lack of random assignments), such as**
  - Are users of the recommender systems more likely convert?
  - Does the recommender system itself cause users to convert?

  Some hidden exogenous variable might influence the choice of using RS as well as conversion.

- **However, significant correlation between using the recommender system and making a request for a proposal**

- **Size of effect has been replicated in other domains**
  - Tourism
  - Electronic consumer products

What is popular?

- **Evaluations on historical datasets measuring accuracy**
- **Most popular datasets**
  - Movies (MovieLens, EachMovie, Netflix)
  - Web 2.0 platforms (tags, music, papers, …)
- **Most popular measures for accuracy**
  - Precision/Recall
    - Items are classified as good or bad
  - MAE (Mean Absolute Error), RMSE (Root Mean Squared Error)
    - Items are rated on a given scale
- **Availability of data heavily biases what is done**
  - Tenor at RecSys conferences to foster live experiments
  - Public infrastructures to enable A/B tests

What is popular? cont.

- **Quantitative survey in the literature**
  - High ranked journal on IS and IR
  - ACM Transactions on Information Systems

- **Evaluation designs ACM TOIS 2004-2010**
  - In total 15 articles on RS
  - Nearly 50% movie domain
  - 80% offline experimentation
  - 2 user experiments under lab conditions
  - 1 qualitative research

Discussion & summary

- General principles of empirical research an current state of practice in evaluating recommendation techniques were presented
- Focus on how to perform empirical evaluations on historical datasets
- Discussion about different methodologies and metrics for measuring the accuracy or coverage of recommendations.
- Overview of which research designs are commonly used in practice.
- From a technical point of view, measuring the accuracy of predictions is a well accepted evaluation goal
  - but other aspects that may potentially impact the overall effectiveness of a recommendation system remain largely under developed.