Learning from User-generated Data
Summer Term 2022

Evaluating Recommender Systems

Markus Schedl
markus.schedl@jku.at

http://www.cp.jku.at

JOHANNES KEPLER UNIVERSITY LINZ
INSTITUTE OF COMPUTATIONAL PERCEPTION
MULTIMEDIA MINING AND SEARCH GROUP

Categories of Evaluation Experiments

- **Offline Testing:**
    - Based on static (sometimes even synthetic) datasets
    - Relying solely on historic data => no real users ever see the recommendations of the system under evaluation
    - Common experimental setups: cross-fold validation, random percentage of data for training/validation/test sets, temporal split
- **Online Testing:**
    - Testing “in the wild” with real users using the system
    - Predominantly A/B testing (different groups of users are exposed to recommendations created by two or more systems to compare)
    - Often done by industry => access to large user base
- **User Studies:**
    - Commonly, adopting questionnaires (quantitative or qualitative)
    - Either designed from scratch or based on existing evaluation frameworks

Evaluating Recommender Systems

- Recommendation can be seen as a special case of a **retrieval task**:
  - "Query" is implicitly given (e.g., user's listening history)
  - Retrieved documents are recommended items
  - Analogously to retrieval, we have (predicted) scores for each item
    → can build a ranked document/item list
  - Exact value of predicted scores is ignored (only ranking matters)
  - Full armory of performance measures used in IR is available

- Recommendation as a **classification task**:
  - Predicting ratings for unknown items, based on known user ratings
  - Some additional evaluation (error) metrics are possible

- Recommendation as a **user-centric task** aimed at satisfying the user
  - "Beyond-accuracy metrics" (coverage, novelty, diversity, etc.)
  - Sometimes, user studies

Evaluation Under Retrieval Aspects

- Compare predicted and known user-item-interactions
- Predicted item is relevant if the user actually consumed/rated it
- Offline testing

Performance Measures:

- Recall and Precision
- F-measure
- Precision a k documents (also Precision@k or P@k)
- Average Precision (AP)
- R-precision
- Reciprocal Rank (RR)
- Mean Average Precision (MAP)
- Discounted Cumulative Gain (DCG)

- Rank Correlation (compare differences in rankings created by 2 algorithms)

Recall and Precision

- Result to a seed item is an *unordered* set of documents.

$$Recall = \frac{|Rel \cap Ret|}{|Rel|}$$

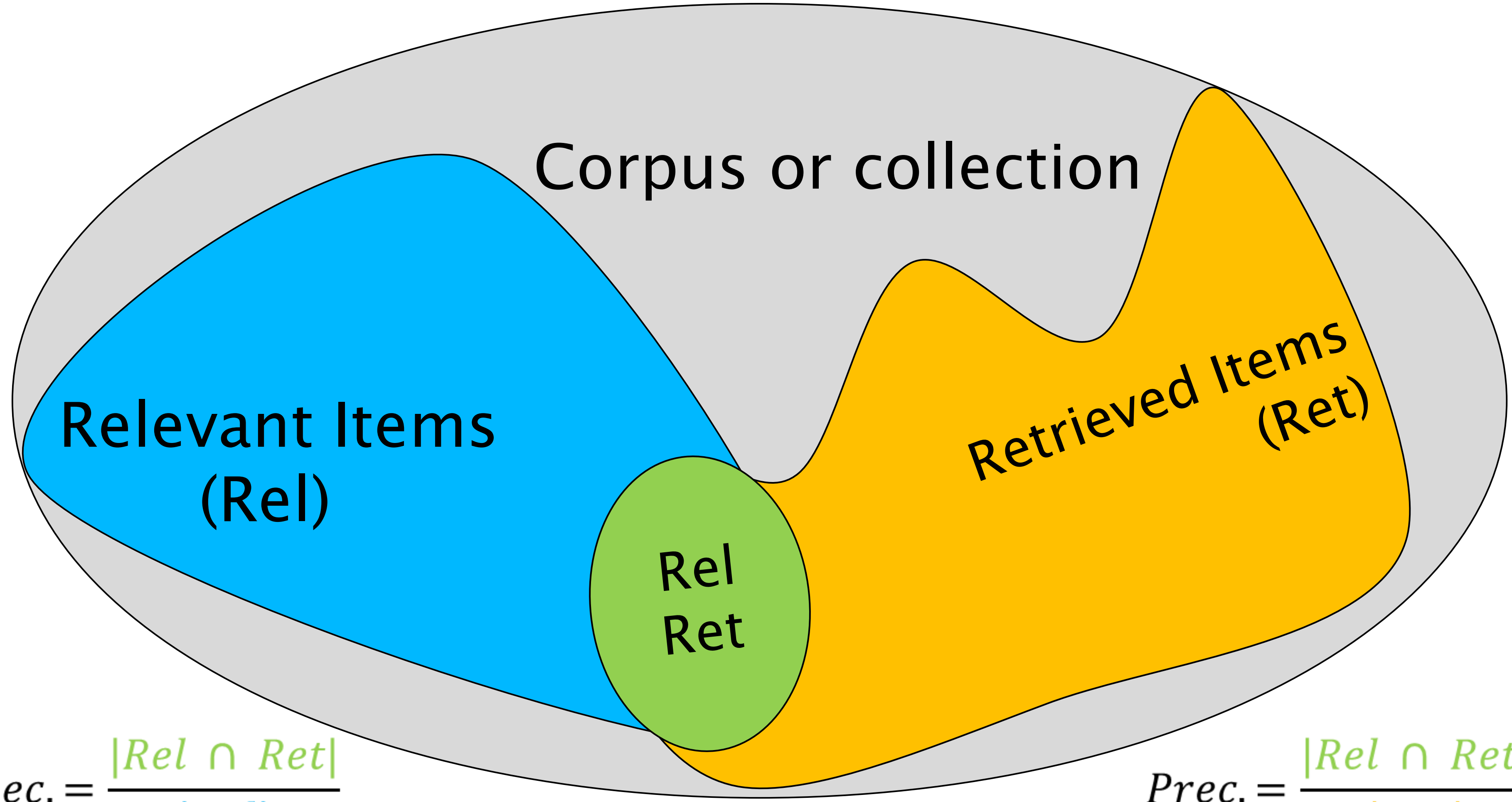
- Recall models how exhaustively the search results satisfy the user's information/entertainment need.

$$Precision = \frac{|Rel \cap Ret|}{|Ret|}$$

- Fraction of relevant items among recommended items.

Q: What do you think is more important in a recommender system: high precision or high recall?

Recall and Precision



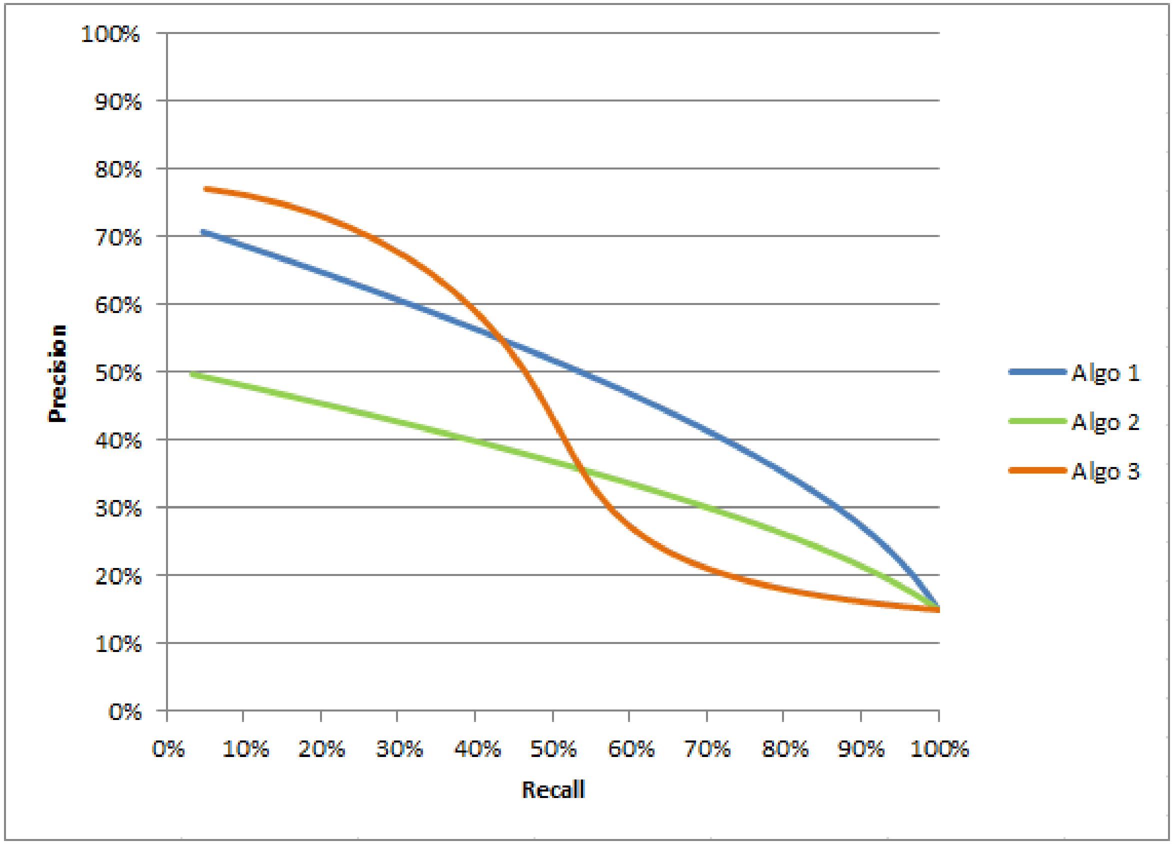
$$Rec. = \frac{|Rel \cap Ret|}{|Rel|}$$

$$Prec. = \frac{|Rel \cap Ret|}{|Ret|}$$

Problem: *Rel* is usually only partially known (cf. “weak labeling”).

Recall and Precision

- Recall and precision varies, dependent on the number of retrieved items (usually, inverse relationship)
  → plots showing "precision at 11 standard recall levels"



http://blog.cluster-text.com/tag/precision-and-recall/

F-measure

- Sometimes also referred to as F_1 *score* or *F-score*

- Harmonic mean of precision and recall:

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad F@k = 2 \times \frac{Precision@k \times Recall@k}{Precision@k + Recall@k}$$

- Aggregate measure, taking into account both precision and recall
  → facilitates easy comparison between different algorithms

- Between the values of recall and precision, usually closer to the smaller one
  → high *F-measures* are only possible if precision and recall high

Precision@k (P@k)

- Assumption: user is in general not interested in all items the system can recommend, but only looks at a number of *k* highest ranked items
- *P@k* assumes that user inspects the *k* items in an *arbitrary order*, and the user inspects *all of them*.

$$P@k = \frac{|Rel \cap Ret[1 \ldots k]|}{k}$$

Ret[1…k] is the top *k* items returned

Average Precision

- Problem of *P@k*: what should be taken as value of *k*? 10? 50? 100?
- Solution: a measure that combines precision values at all possible recall levels
- For every relevant item *d* in recommendation list, compute precision at the rank of *d*:

$$AP = \frac{1}{|Rel|} \times \sum_{i=1}^{|Ret|} relevant(i) \times P@i$$

relevant(i) = 1 iff the i^{th} retrieved item is relevant, 0 otherwise

- If a relevant item does not appear in *Ret*, its precision is 0.
- Implicitly models recall, because accounts for relevant items not in result list.

R-precision

- Assume that there exist exactly R relevant items for a user
- Precision at R^{th} position in the results ranking ($P@R$)
- Predicting exactly the **number of** items relevant for user u in the test set (i.e., the number of items that are known to be liked by the user)
- $\rightarrow$ R is smallest K for which the recommender system can achieve a recall of 1

Q: How do recall and precision relate at R^{th} position in the ranking (where R is the number of relevant items)?

Reciprocal Rank (RR)

- So far, we assumed that all relevant items are equally useful.
- Experiments showed that most tasks do not require high recall, i.e., a user is usually satisfied when presented with a few highly relevant/liked items.
- Assumption: user is satisfied after having encountered the *first relevant item* and this item is recommended at a high rank
- Inverse of the highest rank of the first relevant item:

$$RR = \frac{1}{\min_k \{Ret[k] \in Rel\}}$$

Mean Average Precision (MAP)

- So far, performance measures were defined on a single user.

- In practice, when evaluating recommendation algorithms, we are interested in how well they perform for a variety of different users

$$MAP = \frac{\sum_{i=1}^{|I|} AP(i)}{|I|}$$

I is the set of items, *AP(i)* is the average precision for user *i*

- Also MRR, etc.

Discounted Cumulative Gain (DCG)

- Output of a recommender system is an *ordered* set of items.

- Assumptions:
  - Users prefer highly liked items in the top of the result list.
  - Highly liked items at the end of the result list are less valuable.
  - User assigns different levels of liking (utility) to different items (e.g., ratings from 0-4)

- *Cumulative Gain* (CG): graded relevance up to position *k* in ranking ("gain" for the user):

$$CG@k = \sum_{i=1}^{k} relevance(i)$$

relevance(i) is "likedness" score (rating) the user assigns to item suggested at position *i*

Discounted Cumulative Gain (DCG)

- Example:
  recommended items : i_1 (3), i_2 (2), i_3 (3), i_4 (0), i_5 (1), i_6 (2), i_7 (4), i_8 (0)

$$CG@6 = \sum_{i=1}^{6} relevance(i) = 3 + 2 + 3 + 0 + 1 + 2 = 11$$

- *CG* does not account for ordering of results → *DCG* does

$$DCG@k = relevance(1) + \sum_{i=2}^{k} \frac{relevance(i)}{log_2(i)}$$

$$DCG@6 = 3 + (2 + 1.892 + 0 + 0.431 + 0.774) = 8.10$$

- not normalized: hence, ***NDCG*** = *DCG* / ***IDCG*** (Ideal DCG)

Evaluation under Classification Aspects (Error Metrics)

- Predict ratings for unknown data items
- Offline testing
- Error metrics

Performance Measures:

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)

Mean Absolute Error (MAE)

- RS predict **ratings** for unknown data items (e.g., on 5-point Likert scale)
- Measure how close predicted ratings are to true ratings

$$MAE = \frac{1}{|T|} \cdot \sum_{(u,i) \in T} |r_{u,i}' - r_{u,i}|$$

T … test set
u … user
i … item
$r_{u,i}'$ … predicted rating
$r_{u,i}$ true rating

Small and large prediction errors of an item are similarly treated!

Root Mean Squared Error (RMSE)

- De-facto standard in evaluating rating-based RS
- In contrast to MAE, RMSE disproportionally *penalizes large prediction errors* (squared!)

$$RMSE = \sqrt{\frac{1}{|T|} \cdot \sum_{(u,i) \in T} \left(r'_{u,i} - r_{u,i}\right)^2}$$

T … test set
u … user
i … item
$r_{u,i}{}'$ … predicted rating
$r_{u,i}$ true rating

- Sometimes normalized to range of ratings ($r_{max} - r_{min}$); ranking remains the same

User-centric Evaluation

- Problem with all quantitative effectiveness measures used in offline testing:
  - Do they really assess if the recommended items satisfy the user?
  - What does “satisfy” mean? (e.g., fulfilling an intent, same genre, suited for a specific situation, … => depends on the (dynamic) needs of user)
  - They barely consider the user experience with the system

- More detailed investigation of user experience and satisfaction:
  - Beyond-accuracy metrics
  - Questionnaires (e.g., based on existing UX evaluation frameworks)

Beyond-Accuracy Metrics

- ***Diversity*** (Rationale: recommended items should not be too similar/boring)
  - Intra-list diversity (ILD): average pairwise distance between all items in the recommendation list (requires some meaningful similarity metric, commonly based on some content descriptors)
  - Entropy: (normalized) Shannon entropy based on frequencies of descriptors present in recommendation list (e.g., genres or tags)

- ***Novelty*** (Rationale: user wants to discover new items)
  - System can reach high accuracy just by making “easy” predictions (e.g., recommend always popular songs), but these may be useless for the user
  - Can be defined on a *global* level, e.g., inverse of overall item popularity
  - Can be defined on an *individual* level, e.g., fraction of unseen items in recommendation list (in time window); novelty can refer to different levels, e.g., artist, album, song in the music domain; if task is artist recommendation then an unseen item by a known user is not novel

Beyond-Accuracy Metrics

- ***(Items and User) Coverage*** (Rationale: system should be able to serve all users and give each item a chance to be recommended)
  - Percentage of items that appear in at least one recommendation list
  - Percentage of users for whom recommendations can be made

- ***Serendipity*** (Rationale: user wants to discover something exciting, unexpected); e.g., interesting item from another genre that the user usually does not like; hard to measure though metrics do exist

- ***Explainability*** (Rationale: recommender system should explain *why* an item was recommended => increase trust, credibility, etc.); e.g.:
  - List similar users and their tastes ("...because you friends like it.")
  - Provide contextual explanations ("...because you listen to this kind of music at night.")
  - Content-based explanations ("…because this movie features your favorite actor.")

Questionnaires

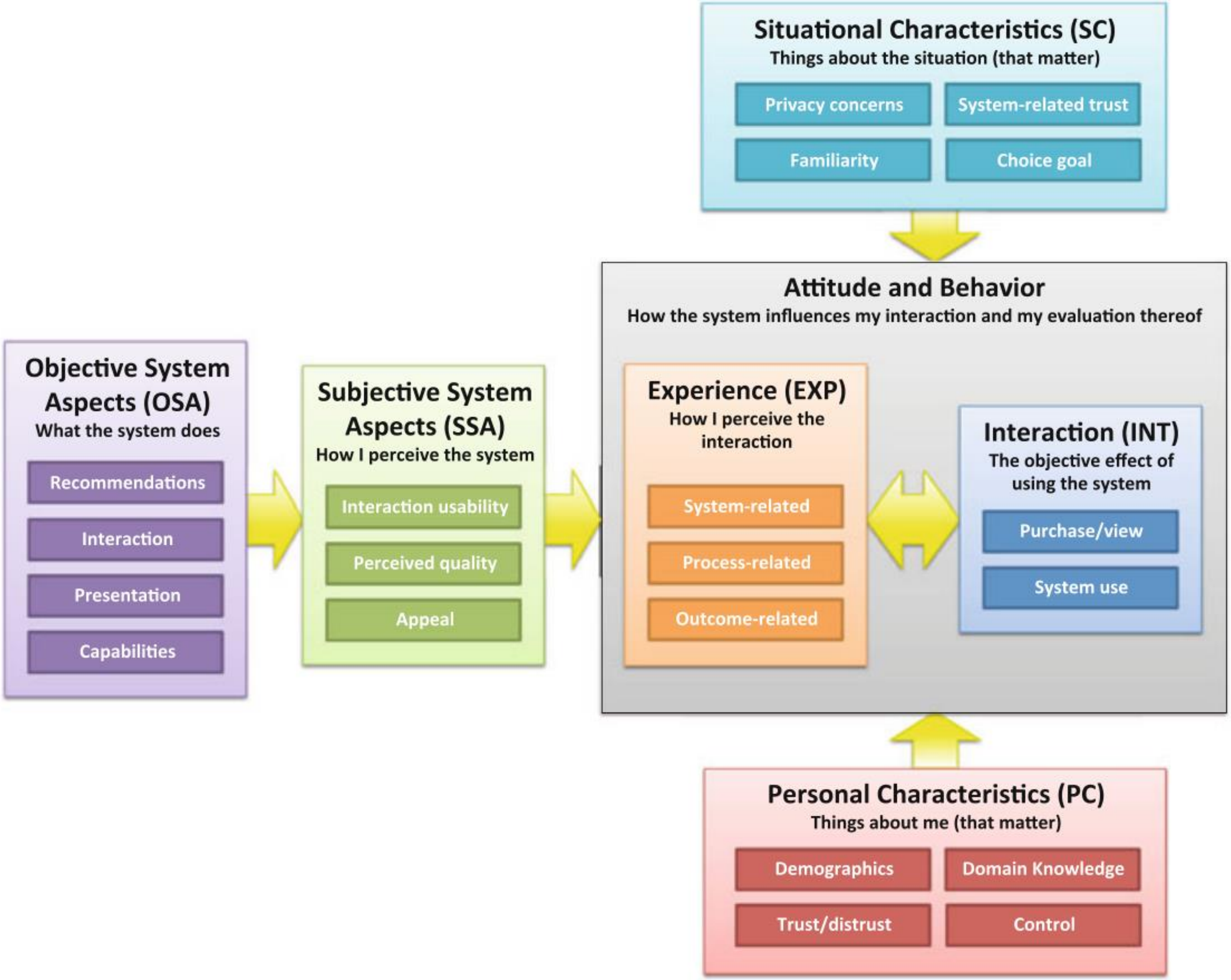
- **Quantitative methods:**
  Likert-style ratings, manual accuracy (or beyond-accuracy) feedback for recommended items, (analyze interaction logs)

- **Qualitative methods:**
  Open-question surveys, structured interviews, diary studies;
  observe user behavior, explicitly ask users about their experiences with the RS

- **UX evaluation frameworks** for RS evaluation:
  - [Pu et al., 2011]: Recommender systems' Quality of user experience (ResQue)
  - [Knijnenburg et al., 2012]: comprehensive framework incl. questionnaires
  - [Pu et al., 2012]: Survey on user-centric evaluation of RS

UX Evaluation Framework

[Knijnenburg et al., 2012]



Example Questions

Perceived recommendation quality

- I liked the items recommended by the system.
- The recommended items fitted my preference.
- The recommended items were relevant.
- The system recommended too many bad items.
- I didn’t like any of the recommended items.
- The items I selected were “the best among the worst”.

Effort to use the system

- The system is convenient.
- I have to invest a lot of effort in the system.
- It takes many mouse-clicks to use the system.

Perceived system effectiveness and fun

- I have fun when I’m using the system.
- I would recommend the system to others.
- Using the system is a pleasant experience.
- The system is useless.
- The system makes me more aware of my choice options.
- I can find better items using the recommender system.

Example Questions

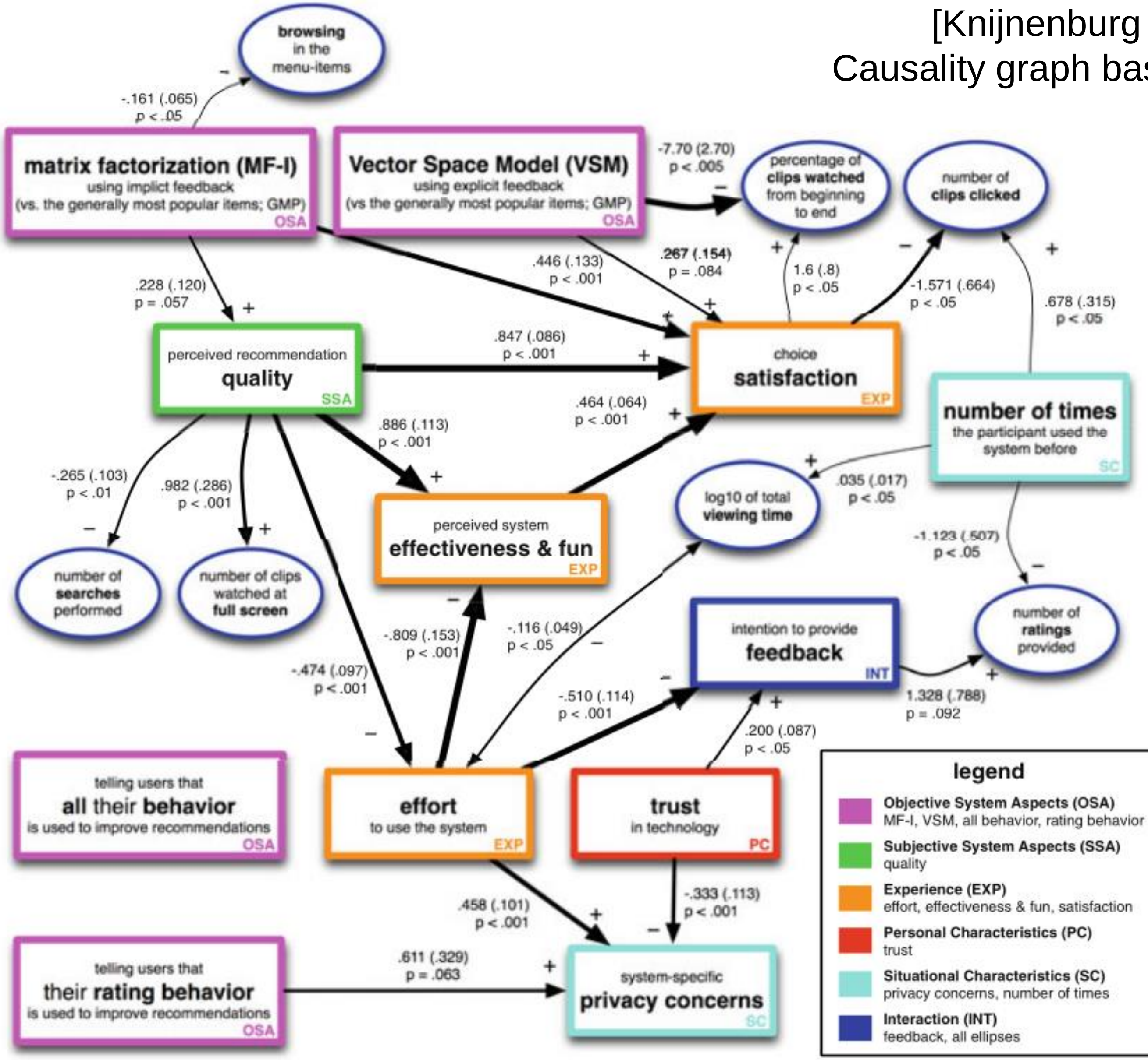
Perceived recommendation variety

- The recommendations contained a lot of variety.
- The recommendations covered many programme genres.
- All the recommended programmes were similar to each other.
- Most programmes were from the same genre.

Choice satisfaction

- I like the items I've chosen.
- I was excited about my chosen items.
- I enjoyed watching my chosen items.
- The items I watched were a waste of my time.
- The chosen items fit my preference.

[Knijnenburg et al., 2012]
Causality graph based on SEM



Summary

- Main flavors of RS evaluation:
  - Offline testing
  - Online testing (A/B testing)
  - User studies
- Different perspectives:
  - Information retrieval (IR)
  - Machine Learning (ML) => rating prediction (classification)
  - User-centric
- Quantitative versus qualitative methods
- Beyond-accuracy metrics
- UX evaluation frameworks

References

[Gunawardana and Shani, 2015]: Evaluating Recommender Systems. Recommender Systems Handbook, 2nd edition, Francesco Ricci, Lior Rokach, Bracha Shapira (eds.), 265–308 (2015).

[Knijnenburg et al., 2012]: Explaining the user experience of recommender systems. User Model User-Adap Inter 22, 441–504 (2012). https://doi.org/10.1007/s11257-011-9118-4

[Pu et al., 2012]: Evaluating recommender systems from the user's perspective: survey of the state of the art. *User Model User-Adap Inter* 22, 317–355 (2012). https://doi.org/10.1007/s11257-011-9115-7

[Pu et al., 2011]: A user-centric evaluation framework for recommender systems. Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011): 157–164. https://doi.org/10.1145/2043932.2043962