Evaluating RecSys

Davide Frey, WIDE Team, Inria Rennes

Source: Recommender Handbook Chapter 8

A variety of Algorithm

Content Based

User-Based Collaborative Filtering

Item-Based Collaborative Filtering

Matrix-Factorization CF

Inria

Variables Affecting RecSys

- Application Domain
- Reliability of Data (ratings)
- User Population
- Allowable CPU and Memory
- Static vs Streaming Item Set
- User goals: discover new stuff vs match their tastes

Evaluation Strategies

Online

- User Studies
- Online Experiments
  (A/B) testing
- Real-world metrics:
  - Click rate
  - Revenue

Offline

- Dataset Driven
- Inherently biased
- A number of custom metrics

Inria

Experimental Setting

- Evaluating Recommender is Experimental Science
- Typical Experimental Guidelines
  - Hypothesis establishment:
    - Algorithm A is more accurate than Algorithm B
  - Controlling Variables:
    - Vary one variable at a time: other variables stay fixed
      - Algorithm
      - Algorithm parameters
      - Data
      - ….
  - Generalization Power
    - We want high generalization power
    - Not Trivial
    - Need diverse settings: datasets, applications
    - The more diverse the data, the better the generalization power

Inria

Offline Experiments

- Simulate the behavior of users based on pre-collected datasets
- Attractive because
  - No need for a deployed system
  - No need for user interaction
  - Easy to compare a variety of algorithms
- Not ideal because
  - Can only answer a limited set of questions: prediction power
  - Results may be biased by the dataset
    - how was the dataset collected?

Inria

Limiting Bias in Offline Evaluation

- Prefiltering data should be done with care:
  - Removing items/users with low counts to reduce experimental cost:
    - Often done in papers but introduces bias
    - Random sampling may be better but
      - Lower scores (not good for papers)
      - May favor algorithms that work better with sparse data
- Dataset itself may be biased
  - Users only rate items they've been exposed to
  - Users only rate items on which they have strong opinions
  - Some users rate more items than others
  - Dataset may have been collected on a system with a running recommender

Inria

Simulating User Behavior

- Typically done by hiding some data in dataset
  - Training set / testing set
- If timestamps are available: time-based split
  - Step through user ratings in temporal order
  - Select random instant for each user and hide everything after that and predict
    - May be costly
  - Sample users, then single test time, hide items after test time for each user
- If time not available or unimportant
  - Sample Users, and sample items to hide

Inria

Given n / all but n

- Common protocol used in papers
  - Use a fixed number of items
  - Exclude a fixed number of items
- Good for identifying when algorithms work best
- May not be representative of real applications

More Complex User Models

- ML based models simulating user behavior
- Vast research field
- But difficult task
- If model is inaccurate, results may be completely off

Inria

User Studies

- Some systems rely on interaction with users
  - Difficult to conduct offline evaluation
- User study
  - Recruit set of subjects
  - Have them interact with recommender
  - Observe and record their behavior
    - Quantitative metrics: completed tasks, time to complete task, accuracy of user task
    - Qualitative metrics: ask questions before, during, or after task

Example of User Study

- Evaluate effect of recommender of browsing behavior on news website
  - Ask users to read a set of stories that are interesting to them
  - Check whether users click on recommendation
  - Do users read more stories when recommendations are available?
  - Ask qualitative questions
  - Track eye movement

Inria

Challenges in User Studies

- Expensive to conduct
- Sometimes require pilot studies
  - Fix bugs
  - Improve recommender
  - Such data should not be used to draw conclusions
- May introduce bias:
  - Users know they're in a study
  - Users may not represent true population
  - Payment of users may lead to bias (e.g subsidy of items they select)

Inria

Between vs Within Subjects

- Compare candidate methods/algorithms
  - between subjects:

    Each subject is assigned a given method and experiments with it
  - Within subjects

    Each subject tests a variety of candidate methods
- Within subjects typically more informative but users are more conscious of experiments
- Between subjects (aka A-B testing)

Inria

More sources of bias

- Displaying items
  - Display items sequentially or all together
  - Order in which items are displayed
  - Randomization may mitigate effects
- Questionnaires
  - Critical to ask neutral questions

  Do not suggest correct answer
  - Do not ask “private” or uncomfortable questions

Inria

Online Evaluation

- Goal of recommender system is to influence user behavior
- Some variables can only be measured on real setting (running system)
- But process is costly and risky
  - Bad recommendation may discourage users
  - Need to evaluate one aspect at a time:
    - Interface
    - Algorithm
    - This may sometimes be difficult

Inria

Recommender System Properties

- Prediction Accuracy
- Coverage
- Confidence
- Trust
- Novelty
- Serendipity
- Diversity
- Utility
- Risk
- Robustness
- Privacy
- Adaptivity
- Scalability

Inria

Measuring Accuracy: RMSE

- Root Mean Squared Error
  - Measures error in predicting rating
  - System generates rating for a set of user-item pairs whose real ratings are known

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} (\hat{r}_{ui} - r_{ui})^2}$$

Inria

Measuring Accuracy: MAE

- Mean Absolute Error
  - Similar to RMSE

$$\text{MAE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} |\hat{r}_{ui} - r_{ui}|}$$

  - RMSE disproportionally penalizes large errors
    - Error 2 on three ratings and 0 on 1 vs. <- preferred by RMSE
    - Error 3 on one rating and 0 on 3 <- preferred by MAE

Inria

Measuring Accuracy: NRMSE, NMAE, avg RMSE, avg MAE

- Normalized RMSE and Normalized MAE
  - Normalized by the range of ratings
    - divided by (rmax – rmin)
  - Simply scaled versions so ranking stays the same
- Averaging to account for unbalanced item/user distribution
  - If item distribution is unbalance
    - Compute per-item RMSE/MAE and then average
  - If user distibution is unbalance
    - Compute per-user RMSE/MAE and then average
- In some cases: recommending bad item is worse than not recommending good one
  - Distortion measure: score that weighs some errors worse than others

Inria

Measuring Accuracy: beyond ratings

- Ratings are not always what we want to predict
  - If goal is to predict whether a user will watch a movie, buy a product
  - Then it may be better to measure accuracy in these terms

	Recommended	Not recommended
Used	True-Positive (tp)	False-Negative (fn)
Not used	False-Positive (fp)	True-Negative (tn)

Inria

Measuring Accuracy: Precision, Recall, FPR

$$\textbf{Precision} = \frac{\#tp}{\#tp + \#fp}$$

$$\textbf{Recall (True Positive Rate)} = \frac{\#tp}{\#tp + \#fn}$$

$$\textbf{False Positive Rate} (1 - \textbf{Specificity}) = \frac{\#fp}{\#fp + \#tn}$$

- Tradeoff: longer vs shorter recommendation lists
- Good to compare over range of list lengths, or at fixed list length (precision at n)
  - Precision-recall curve: compare recall and precision
  - ROC curve: compare TPR and FPR (receiver operating characteristic)

Inria

Measuring Accuracy: Precision vs FPR

- Precision measure proportion of recommendations that were useful for user
  - More relevant for: video rental (we do not care about FPR)
- FPR measures unsuitable recommendation vs population of unsuitable items
  - More relevant if false positives have a cost: e.g. mail item to user who returns it if not happy.

Measuring Accuracy: F-Measure

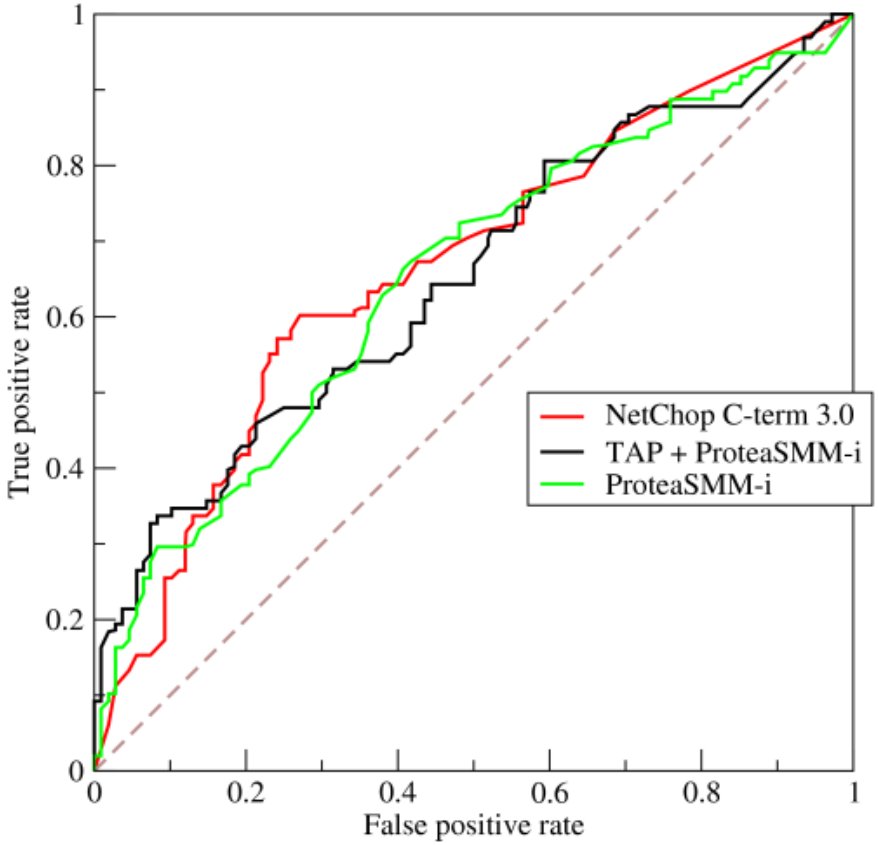
- F-Measure or F1-Score

Harmonic mean of precision and recall

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{2\mathrm{tp}}{2\mathrm{tp} + \mathrm{fp} + \mathrm{fn}}.$$

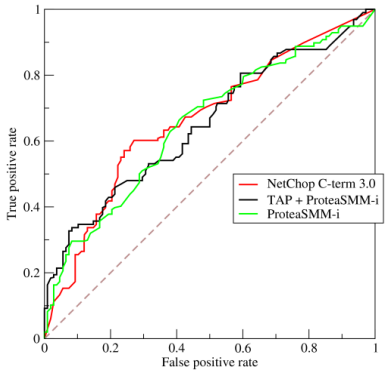
Inria

Measuring Accuracy:
AUC: Area Under the ROC Curve



Measuring Accuracy: F-Measure and AUC

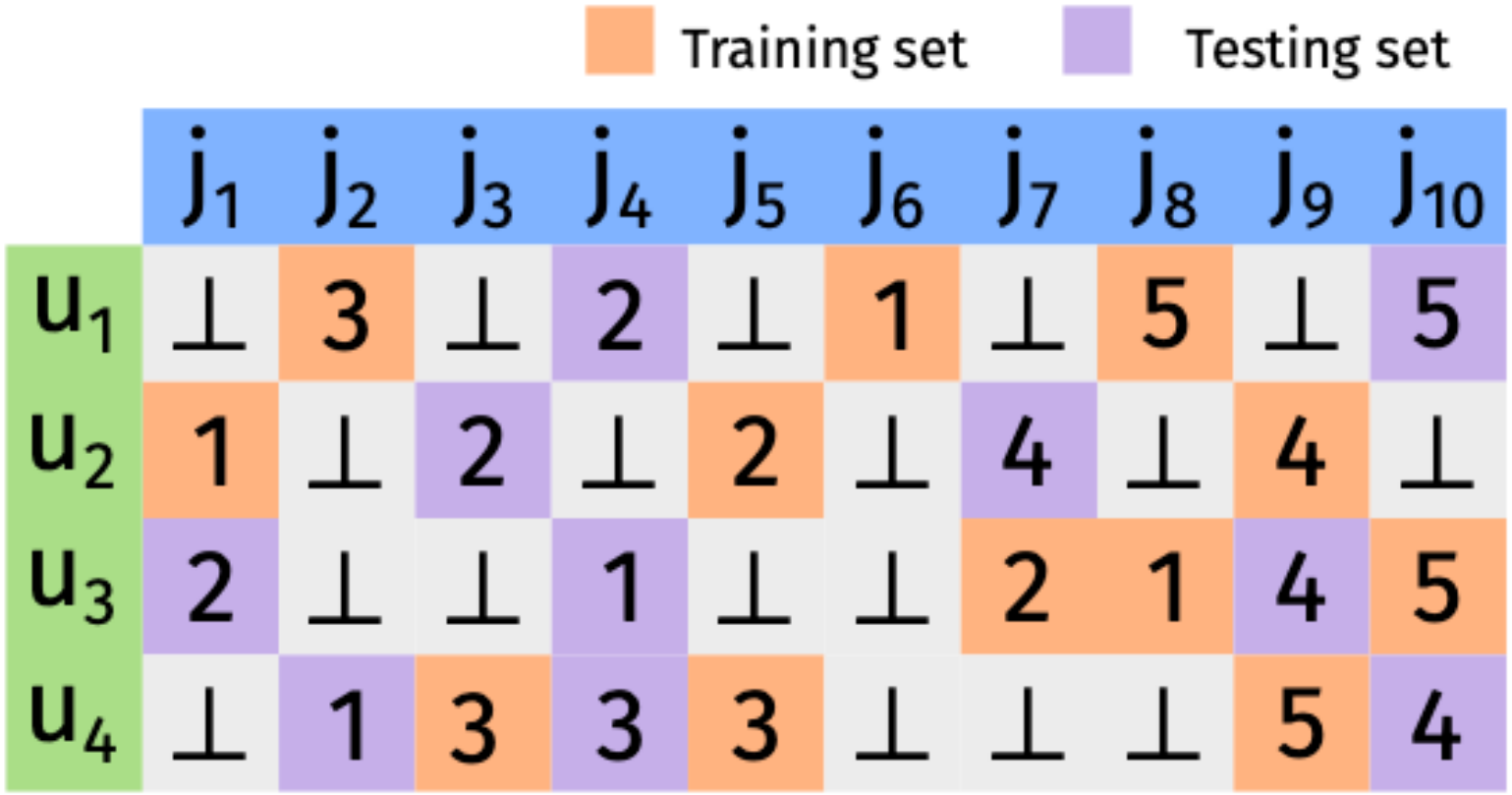
$$F_1 = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$



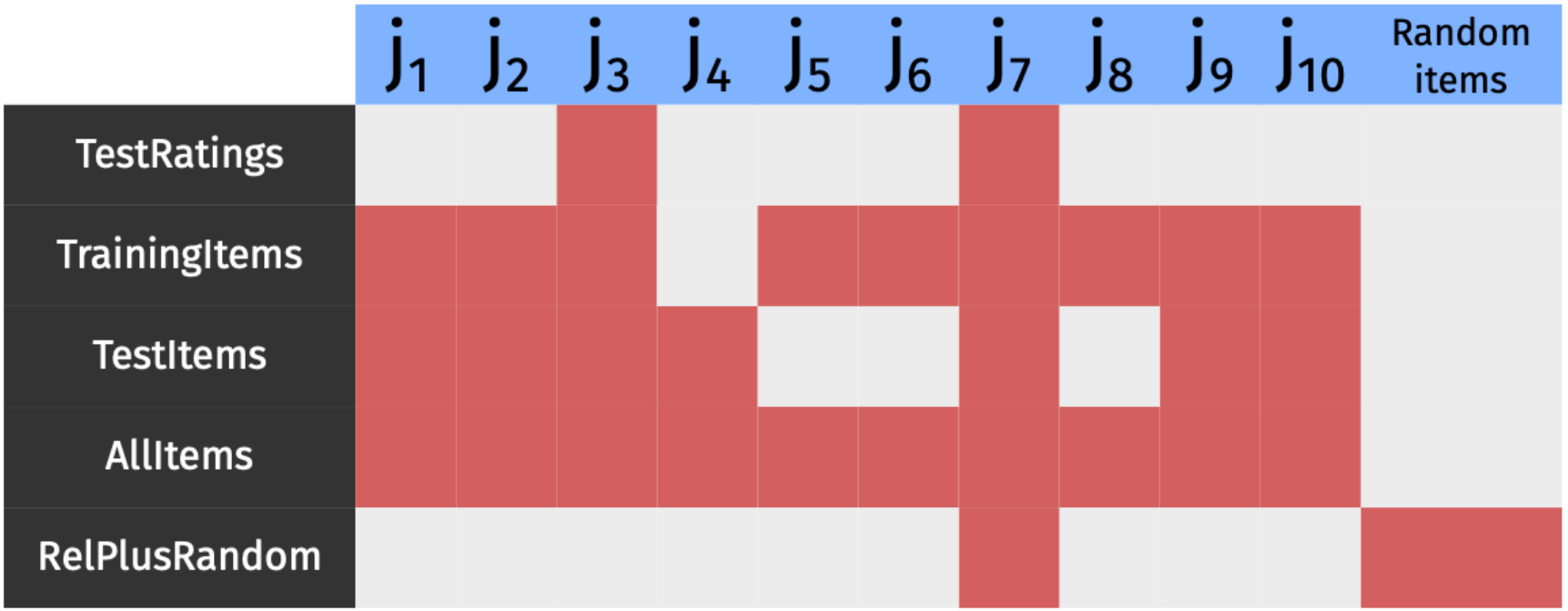
- Useful for comparing algorithms in general
- But when selecting an algorithm for a task
  - Better to choose based on specific need

Inria

Several Ways to Compute Accuracy



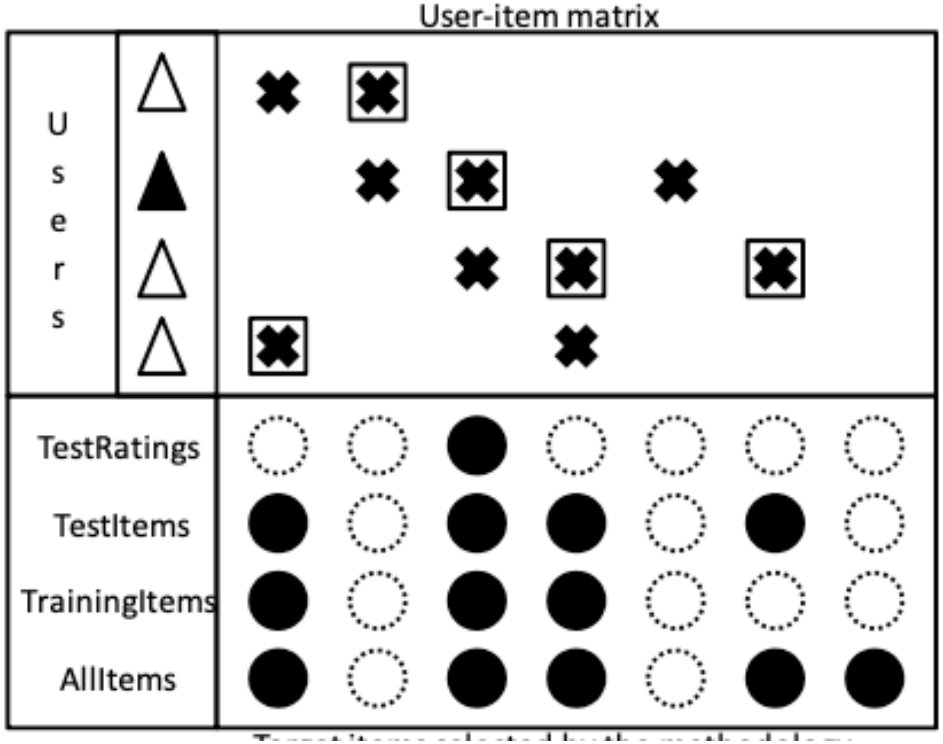
Several Ways to Compute Accuracy



(b) Candidate items for user u_2 $(th = 3)$

Several Ways to Compute Accuracy

Figure 1. Graphical representation of the four methodologies. Target user is represented with a solid triangle, crosses represent the users' ratings (unboxed ratings denote the training set). Black circles represent items included in L_u.



Several Ways to Compute Accuracy

- Depending on set of target items that recommender should rank
  - Test rating: items rated by u in the test set
  - Test items: items that have a test rating by some user and no training rating by target user
  - Training items: items that have a training rating by some user and no training rating by target user
  - AllItems: all items except those rated by target user in the training set
  - OnePlusRandom: a set of highly relevant items is selected among those contained in the test set. Then, a set of non-relevant items is created by randomly selecting N additional items

Inria

Ranking Measures

- Sometimes it's important to rank items into a list
  - Need reference ranking
    - May be based on ratings but not necessarily
  - Need to handle ties in rankings
  - Normalized Distance-based Performance Measure
    - NDPM

Ranking Measures: NDPM

$$C^{+} = \sum_{ij} \operatorname{sgn}(r_{ui} - r_{uj}) \operatorname{sgn}(\hat{r}_{ui} - \hat{r}_{uj})$$ Rankings agree

$$C^{-} = \sum_{ij} \operatorname{sgn}(r_{ui} - r_{uj}) \operatorname{sgn}(\hat{r}_{uj} - \hat{r}_{ui})$$ Rankings disagree

$$C^{u} = \sum_{ij} \operatorname{sgn}^2(r_{ui} - r_{uj})$$ Not tied by reference ranking

$$C^{s} = \sum_{ij} \operatorname{sgn}^2(\hat{r}_{ui} - \hat{r}_{uj})$$ Not tied by system

$$C^{u0} = C^{u} - (C^{+} + C^{-})$$ Not tied by reference ranking, but tied by system

Inria

Ranking Measures: NDPM

$$\text{NDPM} = \frac{C^{-} + 0.5C^{u0}}{C^{u}}$$

- Perfect score of 0 if all preference relations are predicted
- Worst score of 1 if all preference relations are contradicted
- Not predicting a reference preference only penalized by half
- Predicting preference not ordered by reference is not penalized

Inria

Spearman's and Kendall's

- In some cases we know user's true preferences: so tied means actually indifferent
- May use rank correlation
  - Spearman's

$$\rho = \frac{1}{n_u} \frac{\sum_i (r_{i,u} - \bar{r})(\hat{r}_{i,u} - \bar{\hat{r}})}{\sigma(r)\sigma(\hat{r})}$$

  - Kendall's

$$\tau = \frac{C^+ - C^-}{\sqrt{C^u}\sqrt{C^s}}$$

Inria

Item-space Coverage

- Some algorithms may provide high-quality recommendations for a small subset of items only
- Item Space coverage
  - Proportion of items that can be recommended
- Sales Diversity : how unequally items are chosen by a user with a particular system
  - Gini Index
  - Shannon Entropy

Inria

User-Space Coverage

- Measures distributional inequality with respect to users
- Proportion of users for which the system can provide recommendations
  - E.g. Users that rate few items may not be able to receive high-quality recommendations

Cold-Start

- Subproblem of Coverage
  - Cold-start items
  - Cold-start users
- May measure how large those sets are
- Some systems may be better at recommending cold-start items than others

Inria

Confidence

- System's trust in its recommendations
- In some cases, it may be good to display confidence to the user
- Measure of confidence:
  - estimated probability that the predicted value is true
  - Complete distribution
  - Confidence intervals

Trust

- Trust of the user in the recommendations
- Ways to increase trust
  - Recommending items that the user
  - Explaining recommendations
- Measuring Trust
  - Measure number of recommendations that were followed in online test
  - Questionnaires in user studies

Inria

Novelty

- Novel if user did not know about item
- Can be measured in user study
- But can also be estimated offline:

Serendipity

- Measure How surprising a recommendation is
- Deviation from natural prediction
  - Books by authors that the user is less familiar with

Diversity

- How diverse are recommended items
  - Can be measured as opposite to item-item similarity
  - Diversity may come at cost of accuracy or other properties

Utility

- How useful are recommendations
  - E-commerce: revenue
  - With ratings: may use ratings as utility measure

Risk

- Some application may involve risk:
  - Recommending stock items
  - Different recommendations based on user profiles:
    - Risk-averse
    - Risk-seeking
  - Measure not only expected utility but utility variance

Robustness

- Stability of predictions in the presence of fake information
- Typically fake information results from attacks
  - Owner of a hotel boosting ratings
- Hard to evaluate it on real systems
- Mostly evaluated on offline experiments
- Possibly replaying logs to identify attacks

Inria

Privacy

- A recommendation can expose user-profile information
- Say you get a recommendation for “How to leave your partner” and your partner sees it.

Inria

Adaptivity

- How well can the system adapt to
  - User tastes that change
  - News items that become outdated
  - Sudden events: catastrophes

Scalability

- How large a set of users/items can the system operate with
- How long does it take to compute a recommendation
- Particularly important for streaming recommendations (e.g. news items)