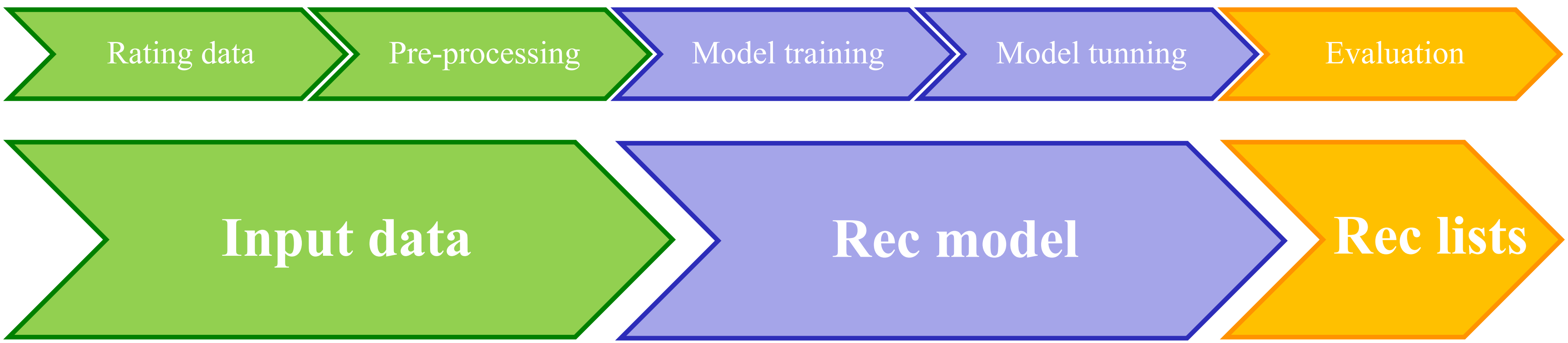
Evaluating Recommender Systems

Masoud Mansoury
AMLab, University of Amsterdam
Discovery Lab, Elsevier

Where we are …



Ratings
Item contents
User attributes
User requirements
User targets (cases)
…

Collaborative Filtering
Content-based
Knowledge-based
Demographic
Hybrid
…

How to measure the quality of the recommendations?
How to evaluate the effectiveness of the recommendation model?

Evaluating Recommender Systems

- **Evaluation is crucial to obtain an understanding of the effectiveness of various recommendation algorithms.**
  - often multifaceted, and a single criterion cannot capture the goals of designer.
  - An incorrect design of the experimental evaluation can lead to either underestimation or overestimation of the true accuracy of a model.

Evaluation Paradigms

- **User studies**
- **Online evaluation**
- **Offline evaluation with historical datasets**

(1) User studies

- **Users are actively recruited, and asked to interact with the recommender system to perform specific tasks.**
- **Feedback can be collected from the user interaction, and the system also collects information about their interaction with the recommender system.**
- **These data are then used to make inferences about the likes or dislikes of the user.**
  - For example, users could be asked to interact with the recommendations at a product site and give their feedback about the quality of the recommendations.
  - Such an approach could then be used to judge the effectiveness of the underlying algorithms.
  - Alternatively, users could be asked to listen to several songs, and then provide their feedback on these songs in the form of ratings.

(1) User studies

+ **It allows for the collection of information about the user interaction with the system.**
  - Various scenarios can be tested about the effect of changing the recommender system on the user interaction, such as the effect of changing a particular algorithm or user-interface.

– **the active awareness of the user about the testing of the recommender system can often bias her choices and actions.**

– **It is also difficult and expensive to recruit large cohorts of users for evaluation purposes.**

– **In many cases, the recruited users are not representative of the general population because the recruitment process is itself a bias-centric filter, which cannot be fully controlled.**

– **Therefore, the results from user evaluations cannot be fully trusted.**

(2) Online Evaluation

- **Online evaluations also leverage user studies except that the users are often real users in a fully deployed or commercial system.**

- **This approach is sometimes less susceptible to bias from the recruitment process, because the users are often directly using the system in the natural course of affairs.**

- **Such systems can often be used to evaluate the comparative performance of various algorithms.**

(2) Online Evaluation

- **Typically, users can be sampled randomly, and the various algorithms can be tested with each sample of users.**

- **A typical example of a metric, which is used to measure the effectiveness of the recommender system on the users, is the conversion rate.**
  - The conversion rate measures the frequency with which a user selects a recommended item.
  - For example, in a news recommender system, one might compute the fraction of times that a user selects a recommended article.
  - These methods are also referred to as A/B testing, and they measure the direct impact of the recommender system on the end user.

(2) Online Evaluation

- **The basic idea in these methods is to compare two algorithms as follows:**
  - Segment the users into two groups A and B.
  - Use one algorithm for group A and another algorithm for group B for a period of time, while keeping all other conditions (e.g., selection process of users) across the two groups as similar as possible.
  - At the end of the process, compare the conversion rate (or other payoff metric) of the two groups.

(2) Online Evaluation

- **The main disadvantage is that such systems cannot be realistically deployed unless a large number of users are already enrolled.**
  - Therefore, it is hard to use this method during the start up phase.
- **Furthermore, such systems are usually not openly accessible, and they are only accessible to the owner of the specific commercial system at hand.**
  - Therefore, such tests can be performed only by the commercial entity, and for the limited number of scenarios handled by their system.
  - This means that the tests are often not generalizable to system-independent benchmarking by scientists and practitioners.

(3) Offline Evaluation with Historical Datasets

- **Offline methods are among the most popular techniques for testing recommendation algorithms, because standardized frameworks and evaluation measures have been developed for such cases.**

- **In offline testing, historical data, such as ratings, are used.**

- **In some cases, temporal information may also be associated with the ratings, such as the time-stamp at which each user has rated the item.**
  - A well known example of a historical data set is the Netflix Prize data set.
  - This data set was originally released in the context of an online contest, and has since been used as a standardized benchmark for testing many algorithms.

(3) Offline Evaluation with Historical Datasets

+ **This approach do not require access to a large user base.**
  - Once a data set has been collected, it can be used as a standardized benchmark to compare various algorithms across a variety of settings.

+ **Multiple data sets from various domains (e.g., music, movies, news) can be used to test the generalizability of the recommender system.**

− **The main disadvantage of offline evaluations is that they do not measure the actual propensity of the user to react to the recommender system in the future.**
  - For example, the data might evolve over time, and the current predictions may not reflect the most appropriate predictions for the future.

− **Furthermore, measures such as accuracy do not capture important characteristics of recommendations, such as serendipity and novelty.**

Offline Recommender Evaluation

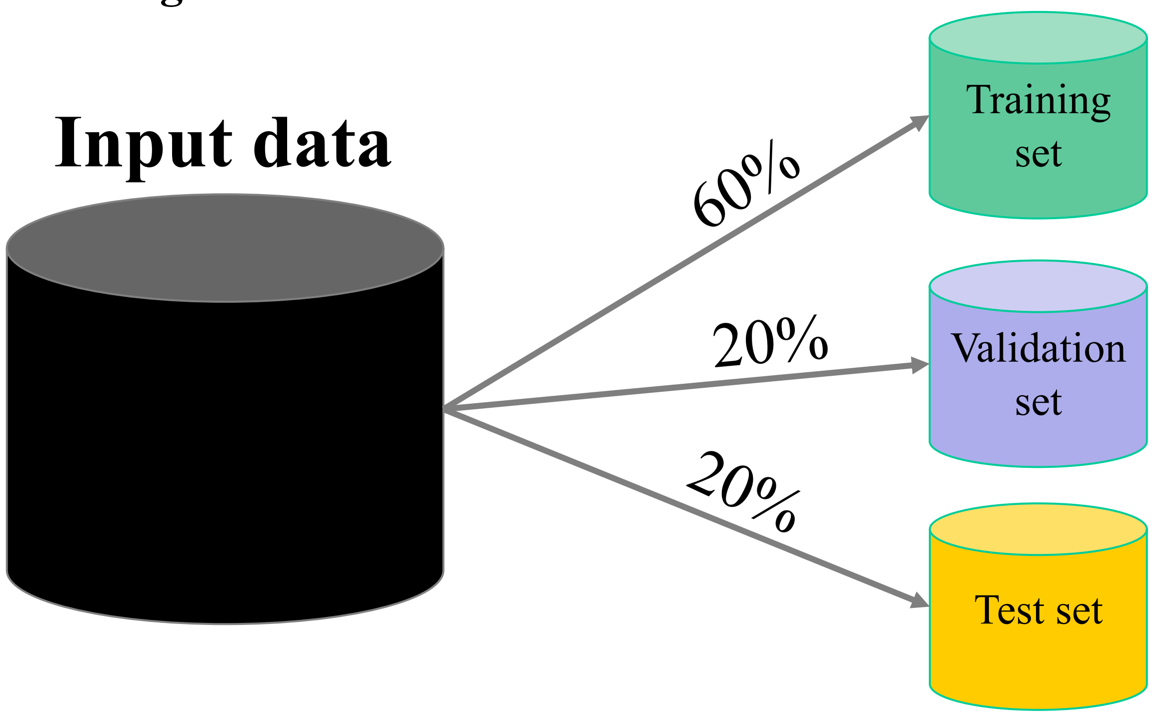
Offline Recommender Evaluation

- **It is crucial to design recommender systems in such a way that the accuracy is not grossly overestimated or underestimated.**
  - For example, one cannot use the same set of specified ratings for both *training* and *evaluation*.
  - Doing so would grossly *overestimate* the accuracy of the underlying algorithm.

- **Only a part of the data is used for training, and the remainder is often used for testing.**

Offline Recommender Evaluation

- To avoid overestimation or underestimation, the input data is divided into:
  - Training data
  - Validation data
  - Testing data



Offline Recommender Evaluation

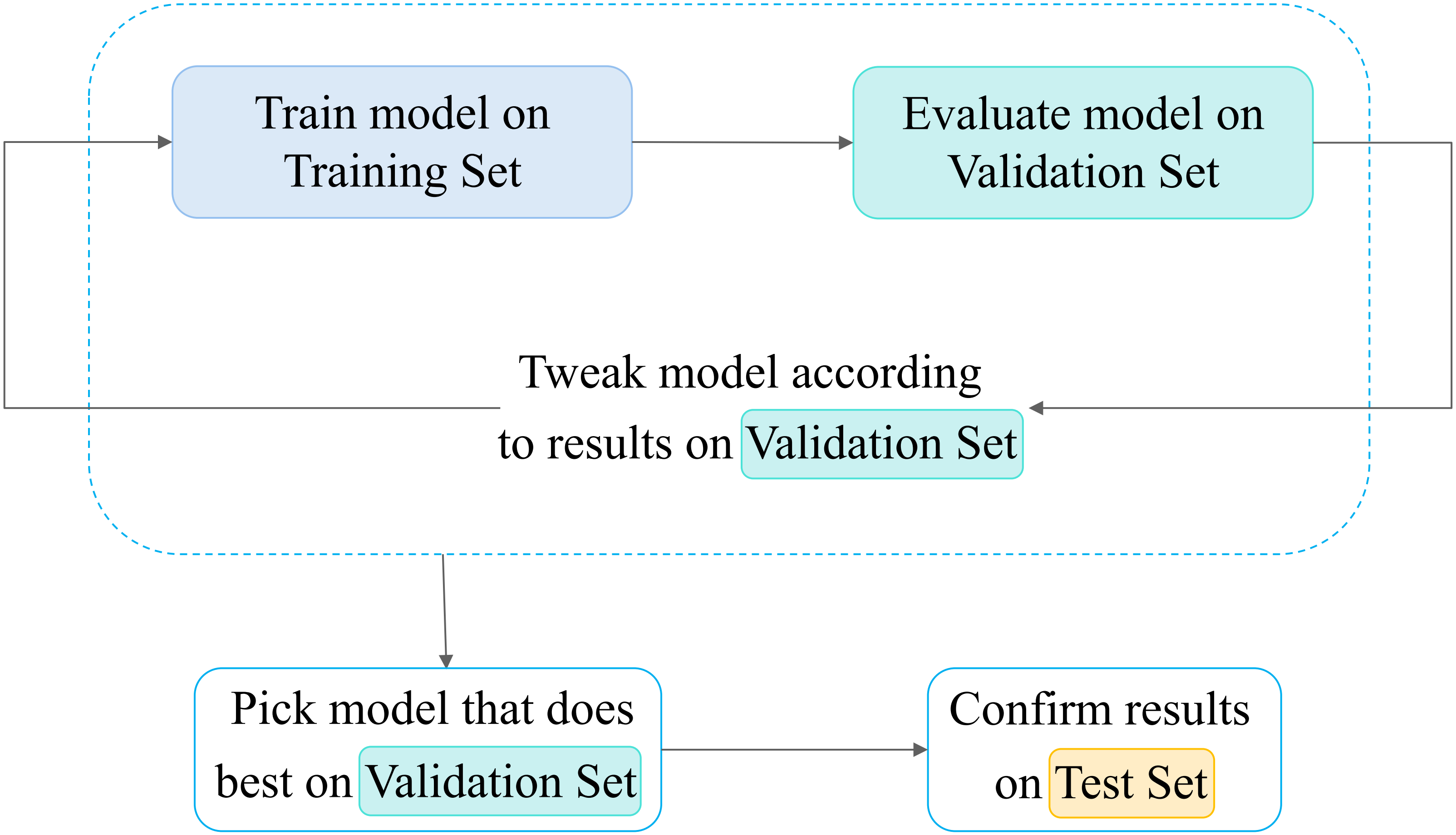
- **Training data**
  - This part of the data is used to build the training model.
  - For example, in a neighborhood model, this part of the data is used to create the similarity matrix from the ratings matrix.
- **Validation data**
  - This part of the data is used for model selection and parameter tuning.
  - For example, parameter k in a neighborhood model may be determined by testing the accuracy over the validation data.
  - Multiple models are built from the training data, the validation data are used to determine the accuracy of each model and select the best one.
- **Testing data**
  - This part of the data is used to test the accuracy of the final (tuned) model.
  - It is important that the testing data are not even looked at during the process of parameter tuning and model selection to prevent overfitting.

Offline Recommender Evaluation

50%	25%	25%
MODEL-BUILDING	VALIDATION (TUNING, MODEL-SELECTION)	TESTING DATA

USED FOR BUILDING
TUNED MODEL

Offline Recommender Evaluation



Segmenting input data

- **In practice, real data sets are not pre-partitioned into training, validation, and test data sets.**

- **Dividing the data is based on users' profile**
  - E.g., 50% as training, 25% as validation, and 25% as test.

Segmenting input data

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u1	*i4*	4
u1	*i5*	5
u1	*i6*	4
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u2	*i5*	3
u2	*i6*	4
u3	*i2*	3
u3	*i3*	3
u3	*i4*	1
u3	*i5*	1
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2
u4	*i4*	3
u4	*i5*	3
u4	*i6*	4
u5	*i1*	1
u5	*i3*	1
u5	*i4*	2
u5	*i5*	3
u5	*i6*	3

Segmenting input data

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u1	*i4*	4
u1	*i5*	5
u1	*i6*	4
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u2	*i5*	3
u2	*i6*	4
u3	*i2*	3
u3	*i3*	3
u3	*i4*	1
u3	*i5*	1
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2
u4	*i4*	3
u4	*i5*	3
u4	*i6*	4
u5	*i1*	1
u5	*i3*	1
u5	*i4*	2
u5	*i5*	3
u5	*i6*	3

$u1$'s profile

$u2$'s profile

$u3$'s profile

$u4$'s profile

$u5$'s profile

Segmenting input data

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u1	*i4*	4
u1	*i5*	5
u1	*i6*	4
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u2	*i5*	3
u2	*i6*	4
u3	*i2*	3
u3	*i3*	3
u3	*i4*	1
u3	*i5*	1
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2
u4	*i4*	3
u4	*i5*	3
u4	*i6*	4
u5	*i1*	1
u5	*i3*	1
u5	*i4*	2
u5	*i5*	3
u5	*i6*	3

***u1*'s profile**

***u2*'s profile**

***u3*'s profile**

***u4*'s profile**

***u5*'s profile**

Training Set

UserID	MovieID	Rating

Validation Set

UserID	MovieID	Rating

Test Set

UserID	MovieID	Rating

Segmenting input data

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u1	*i4*	4
u1	*i5*	5
u1	*i6*	4
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u2	*i5*	3
u2	*i6*	4
u3	*i2*	3
u3	*i3*	3
u3	*i4*	1
u3	*i5*	1
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2
u4	*i4*	3
u4	*i5*	3
u4	*i6*	4
u5	*i1*	1
u5	*i3*	1
u5	*i4*	2
u5	*i5*	3
u5	*i6*	3

$u1$'s profile

$u2$'s profile

$u3$'s profile

$u4$'s profile

$u5$'s profile

Dividing $u1$'s profile

Training Set

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7

Validation Set

UserID	MovieID	Rating
u1	*i4*	4
u1	*i5*	5

Test Set

UserID	MovieID	Rating
u1	*i6*	4

Segmenting input data

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u1	*i4*	4
u1	*i5*	5
u1	*i6*	4
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u2	*i5*	3
u2	*i6*	4
u3	*i2*	3
u3	*i3*	3
u3	*i4*	1
u3	*i5*	1
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2
u4	*i4*	3
u4	*i5*	3
u4	*i6*	4
u5	*i1*	1
u5	*i3*	1
u5	*i4*	2
u5	*i5*	3
u5	*i6*	3

***u1*'s profile**

***u2*'s profile**

***u3*'s profile**

***u4*'s profile**

***u5*'s profile**

Dividing *u2*'s profile

Training Set

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4

Validation Set

UserID	MovieID	Rating
u1	*i4*	4
u1	*i5*	5
u2	*i5*	3

Test Set

UserID	MovieID	Rating
u1	*i6*	4
u2	*i6*	4

Segmenting input data

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u1	*i4*	4
u1	*i5*	5
u1	*i6*	4
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u2	*i5*	3
u2	*i6*	4
u3	*i2*	3
u3	*i3*	3
u3	*i4*	1
u3	*i5*	1
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2
u4	*i4*	3
u4	*i5*	3
u4	*i6*	4
u5	*i1*	1
u5	*i3*	1
u5	*i4*	2
u5	*i5*	3
u5	*i6*	3

***u1*'s profile**

***u2*'s profile**

***u3*'s profile**

***u4*'s profile**

***u5*'s profile**

Dividing *u3*'s profile

Training Set

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u3	*i2*	3
u3	*i3*	3

Validation Set

UserID	MovieID	Rating
u1	*i4*	4
u1	*i5*	5
u2	*i5*	3
u3	*i4*	1

Test Set

UserID	MovieID	Rating
u1	*i6*	4
u2	*i6*	4
u3	*i5*	1

Segmenting input data

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u1	*i4*	4
u1	*i5*	5
u1	*i6*	4
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u2	*i5*	3
u2	*i6*	4
u3	*i2*	3
u3	*i3*	3
u3	*i4*	1
u3	*i5*	1
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2
u4	*i4*	3
u4	*i5*	3
u4	*i6*	4
u5	*i1*	1
u5	*i3*	1
u5	*i4*	2
u5	*i5*	3
u5	*i6*	3

***u1*'s profile**

***u2*'s profile**

***u3*'s profile**

***u4*'s profile**

***u5*'s profile**

Dividing *u4*'s profile

Training Set

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u3	*i2*	3
u3	*i3*	3
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2

Validation Set

UserID	MovieID	Rating
u1	*i4*	4
u1	*i5*	5
u2	*i5*	3
u3	*i4*	1
u4	*i4*	3
u4	*i5*	3

Test Set

UserID	MovieID	Rating
u1	*i6*	4
u2	*i6*	4
u3	*i5*	1
u4	*i6*	4

Segmenting input data

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u1	*i4*	4
u1	*i5*	5
u1	*i6*	4
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u2	*i5*	3
u2	*i6*	4
u3	*i2*	3
u3	*i3*	3
u3	*i4*	1
u3	*i5*	1
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2
u4	*i4*	3
u4	*i5*	3
u4	*i6*	4
u5	*i1*	1
u5	*i3*	1
u5	*i4*	2
u5	*i5*	3
u5	*i6*	3

$u1$'s profile

$u2$'s profile

$u3$'s profile

$u4$'s profile

$u5$'s profile

Dividing $u5$'s profile

Training Set

UserID	MovieID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u3	*i2*	3
u3	*i3*	3
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2
u5	*i1*	1
u5	*i3*	1
u5	*i4*	2

Validation Set

UserID	MovieID	Rating
u1	*i4*	4
u1	*i5*	5
u2	*i5*	3
u3	*i4*	1
u4	*i4*	3
u4	*i5*	3
u5	*i5*	3

Test Set

UserID	MovieID	Rating
u1	*i6*	4
u2	*i6*	4
u3	*i5*	1
u4	*i6*	4
u5	*i6*	3

Accuracy Metrics in Offline Evaluation

Recommendation Problems

- **Prediction version of problem**
  - Predict the rating value for a user-item combination.
  - It is assumed that training data is available, indicating user preferences for items.
  - The missing (or unobserved) values are predicted using this training model.
  - Also referred to as the *matrix completion problem*.
- **Ranking version of problem**
  - Not necessary to predict the ratings of users for specific items in order to make recommendations to users.
  - Rather, recommend the *top-k* items for a particular user.
  - Also referred to as the *top-k recommendation problem*.

Recommendation Problems

- **Prediction version of problem**

	i1	*i2*	*i3*	*i4*	*i5*	*i6*
u1	7	?	7	?	?	4
u2	6	7	?	4	3	?
u3	?	3	?	1	1	?
u4	1	?	2	3	?	4
u5	1	?	1	?	3	3

Recommendation Model

	i1	*i2*	*i3*	*i4*	*i5*	*i6*
u1	7	6	7	4	5	4
u2	6	7	4	4	3	4
u3	3	3	3	1	1	2
u4	1	2	2	3	3	4
u5	1	1	1	2	3	3

- **Ranking version of problem**

	i1	*i2*	*i3*	*i4*	*i5*	*i6*
u1	7	?	7	?	?	4
u2	6	7	?	4	3	?
u3	?	3	?	1	1	?
u4	1	?	2	3	?	4
u5	1	?	1	2	3	3

Recommendation Model

Top-2 recommendation lists

u1	*i2*
u1	*i5*
u2	*i3*
u2	*i6*
u3	*i1*
u3	*i3*
u4	*i5*
u4	*i2*
u5	*i4*
u5	*i2*

Accuracy Metrics for Rating Prediction Task

- **The goal is to understand how close the recommendation model predicts the ratings to actual ratings.**

- **One example is *Mean Squared Error (MSE)*:**
  - r_{uj} is the actual rating provided by user u on item j.
  - $\hat{r}_{uj}$ is the predicted rating for user u on item j.

$$\boldsymbol{MSE = \frac{\sum_{(u,j) \in TestSet} (\hat{r}_{uj} - r_{uj})^2}{\#TestSet}}$$

  - Clearly, smaller values of the MSE are indicative of superior performance.

Accuracy Metrics for Rating Prediction Task

Training Set

User ID	Movie ID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u3	*i2*	3
u3	*i3*	3
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2

Accuracy Metrics for Rating Prediction Task

Training Set

User ID	Movie ID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u3	*i2*	3
u3	*i3*	3
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2

Recommendation model

Accuracy Metrics for Rating Prediction Task

Training Set

User ID	Movie ID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u3	*i2*	3
u3	*i3*	3
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2

→ **Recommendation model** →

Test Set

User ID	Movie ID	Rating r_{uj}	Predicted Rating $\hat{r}_{uj}$
u1	*i6*	4	?
u2	*i6*	4	?
u3	*i5*	1	?
u4	*i6*	4	?

Accuracy Metrics for Rating Prediction Task

Training Set

User ID	Movie ID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u3	*i2*	3
u3	*i3*	3
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2

Recommendation model

Test Set

User ID	Movie ID	Rating r_{uj}	Predicted Rating $\hat{r}_{uj}$
u1	*i6*	4	**3.8**
u2	*i6*	4	**4.3**
u3	*i5*	1	**2.9**
u4	*i6*	4	**4.9**

Accuracy Metrics for Rating Prediction Task

Training Set

User ID	Movie ID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u3	*i2*	3
u3	*i3*	3
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2

Recommendation model

Test Set

User ID	Movie ID	Rating r_{uj}	Predicted Rating $\hat{r}_{uj}$	$\hat{r}_{uj} - r_{uj}$
u1	*i6*	4	**3.8**	**-0.2**
u2	*i6*	4	**4.3**	**0.3**
u3	*i5*	1	**2.9**	**1.9**
u4	*i6*	4	**4.9**	**0.9**

Accuracy Metrics for Rating Prediction Task

Training Set

User ID	Movie ID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u3	*i2*	3
u3	*i3*	3
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2

Recommendation model

Test Set

User ID	Movie ID	Rating r_{uj}	Predicted Rating $\hat{r}_{uj}$	$\hat{r}_{uj} - r_{uj}$
u1	*i6*	4	**3.8**	**-0.2**
u2	*i6*	4	**4.3**	**0.3**
u3	*i5*	1	**2.9**	**1.9**
u4	*i6*	4	**4.9**	**0.9**

$$MSE = \frac{\sum_{(u,j)\in TestSet}(\hat{r}_{uj} - r_{uj})^2}{\#TestSet} = \frac{(-0.2)^2 + 0.3^2 + 1.9^2 + 0.9^2}{4} = 1.138$$

Accuracy Metrics for Rating Prediction Task

- **Root Mean Square Error (RMSE)**
  - The square-root of MSE

$$\boldsymbol{RMSE = \sqrt{MSE}}$$

  - Often used instead of MSE.
  - Standard metric used for Netflix Prize contest.

Accuracy Metrics for Rating Prediction Task

Training Set

User ID	Movie ID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u3	*i2*	3
u3	*i3*	3
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2

→ **Recommendation model** →

Test Set

User ID	Movie ID	Rating r_{uj}	Predicted Rating $\hat{r}_{uj}$	$\hat{r}_{uj} - r_{uj}$
u1	*i6*	4	**3.8**	**-0.2**
u2	*i6*	4	**4.3**	**0.3**
u3	*i5*	1	**2.9**	**1.9**
u4	*i6*	4	**4.9**	**0.9**

$$MSE = \frac{\sum_{(u,j)\in TestSet}(\hat{r}_{uj} - r_{uj})^2}{\#TestSet} = \frac{(-0.2)^2 + 0.3^2 + 1.9^2 + 0.9^2}{4} = 1.138$$

$$RMSE = \sqrt{MSE} = 1.067$$

Accuracy Metrics for Rating Prediction Task

- **Mean Absolute Error (MAE)**

$$\boldsymbol{MAE = \frac{\sum_{(u,j)\in TestSet}\left|\hat{r}_{uj} - r_{uj}\right|}{\#TestSet}}$$

Accuracy Metrics for Rating Prediction Task

Training Set

User ID	Movie ID	Rating
u1	*i1*	7
u1	*i2*	6
u1	*i3*	7
u2	*i1*	6
u2	*i2*	7
u2	*i4*	4
u3	*i2*	3
u3	*i3*	3
u4	*i1*	1
u4	*i2*	2
u4	*i3*	2

Recommendation model

Test Set

User ID	Movie ID	Rating r_{uj}	Predicted Rating $\hat{r}_{uj}$	$\hat{r}_{uj} - r_{uj}$
u1	*i6*	4	**3.8**	**-0.2**
u2	*i6*	4	**4.3**	**0.3**
u3	*i5*	1	**2.9**	**1.9**
u4	*i6*	4	**4.9**	**0.9**

$$MSE = \frac{\sum_{(u,j)\in TestSet}(\hat{r}_{uj} - r_{uj})^2}{\#TestSet} = \frac{(-0.2)^2 + 0.3^2 + 1.9^2 + 0.9^2}{4} = 1.138$$

$$RMSE = \sqrt{MSE} = 1.067$$

$$MAE = \frac{\sum_{(u,j)\in TestSet}\left|\hat{r}_{uj} - r_{uj}\right|}{\#TestSet} = \frac{|-0.2| + |0.3| + |1.9| + |0.9|}{4} = 0.825$$

RMSE versus MAE

- **Is RMSE or MAE better as an evaluation measure?**
  - There is no clear answer to this question, as this depends on the application at hand.

- **RMSE is more significantly affected by large error values or outliers.**
  - A few badly predicted ratings can significantly ruin the RMSE.

RMSE versus MAE

- **In applications where robustness of prediction across various ratings is very important, the RMSE may be a more appropriate measure.**

- **MAE is a better reflection of the accuracy when the importance of outliers in the evaluation is limited.**

- **The main problem with RMSE is that it is not a true reflection of the average error, and it can sometimes lead to misleading results.**

Accuracy Metrics for Ranking Task

- **The goal is to understand how *relevant* or *useful* the recommendations are for the customer.**

- **The test data is used as users' actual consumed items.**
  - Also referred to as *ground-truth data* or *true positives*
  - In the case of *unary data*, all 1's are consumed items
  - In the case of *interval-based rating data*, high ratings are converted into consumed items

Accuracy Metrics for Ranking Task

- **The goal is to understand how *relevant* or *useful* the recommendations are for the customer.**

- **The test data is used as users' actual consumed items.**
  - Also referred to as *ground-truth data* or *true positives*
  - In the case of *unary data*, all 1's are consumed items
  - In the case of *interval-based rating data*, high ratings are converted into consumed items

	i1	*i2*	*i3*	*i4*	*i5*	*i6*
u1	7		7			4
u2	6	7		4	3	
u3		3		1	1	
u4	1		2	6		4
u5	6		1	2	3	7

Assuming 6 and 7 to be liked items

Accuracy Metrics for Ranking Task

- **The goal is to understand how *relevant* or *useful* the recommendations are for the customer.**

- **The test data is used as users' actual consumed items.**
  - Also referred to as *ground-truth data* or *true positives*
  - In the case of *unary data*, all 1's are consumed items
  - In the case of *interval-based rating data*, high ratings are converted into consumed items

	i1	*i2*	*i3*	*i4*	*i5*	*i6*
u1	7		7			4
u2	6	7		4	3	
u3		3		1	1	
u4	1		2	6		4
u5	6		1	2	3	7

Assuming 6 and 7 to be liked items

Accuracy Metrics for Ranking Task

- **The goal is to understand how *relevant* or *useful* the recommendations are for the customer.**

- **The test data is used as users' actual consumed items.**
  - Also referred to as *ground-truth data* or *true positives*
  - In the case of *unary data*, all 1's are consumed items
  - In the case of *interval-based rating data*, high ratings are converted into consumed items

	i1	*i2*	*i3*	*i4*	*i5*	*i6*
u1	7		7			4
u2	6	7		4	3	
u3		3		1	1	
u4	1		2	6		4
u5	6		1	2	3	7

Assuming 6 and 7 to be liked items

u1	*i1*
	i3

Accuracy Metrics for Ranking Task

- **The goal is to understand how *relevant* or *useful* the recommendations are for the customer.**

- **The test data is used as users' actual consumed items.**
  - Also referred to as *ground-truth data* or *true positives*
  - In the case of *unary data*, all 1's are consumed items
  - In the case of *interval-based rating data*, high ratings are converted into consumed items

	i1	*i2*	*i3*	*i4*	*i5*	*i6*
u1	7		7			4
u2	6	7		4	3	
u3		3		1	1	
u4	1		2	6		4
u5	6		1	2	3	7

Assuming 6 and 7 to be liked items

u1	*i1*
	i3
u2	*i1*
	i2

Accuracy Metrics for Ranking Task

- **The goal is to understand how *relevant* or *useful* the recommendations are for the customer.**

- **The test data is used as users' actual consumed items.**
  - Also referred to as *ground-truth data* or *true positives*
  - In the case of *unary data*, all 1's are consumed items
  - In the case of *interval-based rating data*, high ratings are converted into consumed items

	i1	*i2*	*i3*	*i4*	*i5*	*i6*
u1	7		7			4
u2	6	7		4	3	
u3		3		1	1	
u4	1		2	6		4
u5	6		1	2	3	7

Assuming 6 and 7 to be liked items

u1	*i1*
	i3
u2	*i1*
	i2
u3	-

Accuracy Metrics for Ranking Task

- **The goal is to understand how *relevant* or *useful* the recommendations are for the customer.**

- **The test data is used as users' actual consumed items.**
  - Also referred to as *ground-truth data* or *true positives*
  - In the case of *unary data*, all 1's are consumed items
  - In the case of *interval-based rating data*, high ratings are converted into consumed items

	i1	*i2*	*i3*	*i4*	*i5*	*i6*
u1	7		7			4
u2	6	7		4	3	
u3		3		1	1	
u4	1		2	6		4
u5	6		1	2	3	7

Assuming 6 and 7 to be liked items

u1	*i1*
	i3
u2	*i1*
	i2
u3	-
u4	*i4*

Accuracy Metrics for Ranking Task

- **The goal is to understand how *relevant* or *useful* the recommendations are for the customer.**

- **The test data is used as users' actual consumed items.**
  - Also referred to as *ground-truth data* or *true positives*
  - In the case of *unary data*, all 1's are consumed items
  - In the case of *interval-based rating data*, high ratings are converted into consumed items

	i1	*i2*	*i3*	*i4*	*i5*	*i6*
u1	7		7			4
u2	6	7		4	3	
u3		3		1	1	
u4	1		2	6		4
u5	6		1	2	3	7

Assuming 6 and 7 to be liked items

u1	*i1*
	i3
u2	*i1*
	i2
u3	-
u4	*i4*
u5	*i1*
	i6

Accuracy Metrics for Ranking Task

- **Recommendation lists are compared with the ground-truth data and the accuracy is measured as the degree to which they are matched.**

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

Metric: Precision

- **A measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved.**
  - What percentage of the recommended items are relevant?
  - The percentage of the recommended items that also exist in the test data

- **Precision for each user u**

Recommended items

All rated items rated by user u in test set

$$\boldsymbol{precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{n}}$$

Size of the recommendation list

  - The overall precision is the average precision across all users

Metric: Precision

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{n}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

Metric: Precision

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{n}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

Metric: Precision

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{n}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

Metric: Precision

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{n}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	
u2	*i2*	
	i3	
	i4	
	i5	
u3	*i1*	
	i2	
	i5	
	i7	
u4	*i3*	
	i4	
	i5	
	i6	

Metric: Precision

$$\boldsymbol{precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{n}}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	
	i3	
	i4	
	i5	
u3	*i1*	
	i2	
	i5	
	i7	
u4	*i3*	
	i4	
	i5	
	i6	

Metric: Precision

$$\boldsymbol{precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{n}}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

Metric: Precision

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{n}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$precision = 100 \times \frac{1}{4} = 25\%$$

Metric: Precision

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{n}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	✓
	i3	✓
	i4	✗
	i5	✓
u3	*i1*	
	i2	
	i5	
	i7	
u4	*i3*	
	i4	
	i5	
	i6	

$$precision = 100 \times \frac{1}{4} = 25\%$$

Metric: Precision

$$precision(n) = 100\times\frac{|\mathcal{R}(n) \cap G|}{n}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$precision = 100\times\frac{1}{4} = 25\%$$

$$precision = 100\times\frac{3}{4} = 75\%$$

Metric: Precision

$$precision(n) = 100\times\frac{|\mathcal{R}(n)\cap G|}{n}$$

Ground-truth data

$u1$	$i2$
	$i3$
$u2$	$i1$
	$i2$
	$i3$
	$i5$
	$i7$
$u3$	$i3$
	$i4$
	$i6$
$u4$	$i1$
	$i2$
	$i3$
	$i4$
	$i5$
	$i6$

Top-4 recommendation list

$u1$	$i1$	✗
	$i3$	✓
	$i4$	✗
	$i6$	✗
$u2$	$i2$	✓
	$i3$	✓
	$i4$	✗
	$i5$	✓
$u3$	$i1$	✗
	$i2$	✗
	$i5$	✗
	$i7$	✗
$u4$	$i3$	
	$i4$	
	$i5$	
	$i6$	

$$precision = 100\times\frac{1}{4} = 25\%$$

$$precision = 100\times\frac{3}{4} = 75\%$$

Metric: Precision

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{n}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$precision = 100 \times \frac{1}{4} = 25\%$$

$$precision = 100 \times \frac{3}{4} = 75\%$$

$$precision = 100 \times \frac{0}{4} = 0\%$$

Metric: Precision

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{n}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	✓
	i3	✓
	i4	✗
	i5	✓
u3	*i1*	✗
	i2	✗
	i5	✗
	i7	✗
u4	*i3*	✓
	i4	✓
	i5	✓
	i6	✓

$$precision = 100 \times \frac{1}{4} = 25\%$$

$$precision = 100 \times \frac{3}{4} = 75\%$$

$$precision = 100 \times \frac{0}{4} = 0\%$$

Metric: Precision

$$precision(n) = 100\times\frac{|\mathcal{R}(n)\cap G|}{n}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	✓
	i3	✓
	i4	✗
	i5	✓
u3	*i1*	✗
	i2	✗
	i5	✗
	i7	✗
u4	*i3*	✓
	i4	✓
	i5	✓
	i6	✓

$$precision = 100\times\frac{1}{4} = 25\%$$

$$precision = 100\times\frac{3}{4} = 75\%$$

$$precision = 100\times\frac{0}{4} = 0\%$$

$$precision = 100\times\frac{4}{4} = 100\%$$

Metric: Precision

$$precision(n) = 100\times\frac{|\mathcal{R}(n) \cap G|}{n}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	✓
	i3	✓
	i4	✗
	i5	✓
u3	*i1*	✗
	i2	✗
	i5	✗
	i7	✗
u4	*i3*	✓
	i4	✓
	i5	✓
	i6	✓

$$precision = 100\times\frac{1}{4} = 25\%$$

$$precision = 100\times\frac{3}{4} = 75\%$$

$$precision = 100\times\frac{0}{4} = 0\%$$

$$precision = 100\times\frac{4}{4} = 100\%$$

$$\overline{precision} = 50\%$$

Metric: Precision

- **Is precision enough to show the quality of recommendations?**

- **The precision value heavily depends on the number of items rated by a user in test set.**
  - Precision would be higher for users with more rated items in test set.
  - When user profile in test set is larger, there is higher chance that those items appear in recommendation list.

- **A complementary metric is recall.**

Metric: Recall

- **A measure of completeness, determines the fraction of relevant items retrieved out of all relevant items.**
  - What percentage of the relevant items are recommended?
  - The percentage of the relevant items in test data that also appeared in the recommendation lists.

- **Recall for each user u**

Recommended items

All rated items rated by user u in test set

$$\boldsymbol{recall(n)} = \mathbf{100} \times \frac{|\mathcal{R}(\boldsymbol{n}) \cap \boldsymbol{G}|}{|\boldsymbol{G}|}$$

Number of the rated items in test set

  - The overall recall is the average recall across all users

Metric: Recall

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{|G|}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

Metric: Recall

$$\boldsymbol{precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{|G|}}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

Metric: Recall

$$\boldsymbol{precision}(\boldsymbol{n}) = \mathbf{100} \times \frac{|\mathcal{R}(\boldsymbol{n}) \cap \boldsymbol{G}|}{|\boldsymbol{G}|}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

Metric: Recall

$$\boldsymbol{precision(n) = 100\times\frac{|\mathcal{R}(n)\cap G|}{|G|}}$$

Ground-truth data

User	Item
u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

User	Item
u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$\boldsymbol{recall = 100\times\frac{1}{2} = 50\%}$$

Metric: Recall

$$\boldsymbol{precision(n) = 100\times\frac{|\mathcal{R}(n)\cap G|}{|G|}}$$

Ground-truth data

u1	*i2*	✗
	i3	✓
u2	*i1*	✗
	i2	✓
	i3	✓
	i5	✓
	i7	✗
u3	*i3*	
	i4	
	i6	
u4	*i1*	
	i2	
	i3	
	i4	
	i5	
	i6	

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$\boldsymbol{recall = 100\times\frac{1}{2} = 50\%}$$

Metric: Recall

$$precision(n) = 100\times\frac{|\mathcal{R}(n)\cap G|}{|G|}$$

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$recall = 100\times\frac{1}{2} = 50\%$$

$$recall = 100\times\frac{3}{5} = 60\%$$

Metric: Recall

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{|G|}$$

Ground-truth data

u1	*i2*	✗
	i3	✓
u2	*i1*	✗
	i2	✓
	i3	✓
	i5	✓
	i7	✗
u3	*i3*	✗
	i4	✗
	i6	✗
u4	*i1*	
	i2	
	i3	
	i4	
	i5	
	i6	

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$recall = 100 \times \frac{1}{2} = 50\%$$

$$recall = 100 \times \frac{3}{5} = 60\%$$

Metric: Recall

$$\boldsymbol{precision(n)} = \mathbf{100}\times\frac{|\mathcal{R}(\boldsymbol{n}) \cap \boldsymbol{G}|}{|\boldsymbol{G}|}$$

Ground-truth data

u1	*i2*	✗
	i3	✓
u2	*i1*	✗
	i2	✓
	i3	✓
	i5	✓
	i7	✗
u3	*i3*	✗
	i4	✗
	i6	✗
u4	*i1*	
	i2	
	i3	
	i4	
	i5	
	i6	

?

Top-4 recommendation list

u1	*i1*	$\boldsymbol{recall} = \mathbf{100}\times\frac{\mathbf{1}}{\mathbf{2}} = \mathbf{50\%}$
	i3	
	i4	
	i6	
u2	*i2*	$\boldsymbol{recall} = \mathbf{100}\times\frac{\mathbf{3}}{\mathbf{5}} = \mathbf{60\%}$
	i3	
	i4	
	i5	
u3	*i1*	$\boldsymbol{recall} = \mathbf{100}\times\frac{\mathbf{0}}{\mathbf{3}} = \mathbf{0\%}$
	i2	
	i5	
	i7	
u4	*i3*	
	i4	
	i5	
	i6	

Metric: Recall

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{|G|}$$

Ground-truth data

u1	*i2*	✗
	i3	✓
u2	*i1*	✗
	i2	✓
	i3	✓
	i5	✓
	i7	✗
u3	*i3*	✗
	i4	✗
	i6	✗
u4	*i1*	✗
	i2	✗
	i3	✓
	i4	✓
	i5	✓
	i6	✓

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$recall = 100 \times \frac{1}{2} = 50\%$$

$$recall = 100 \times \frac{3}{5} = 60\%$$

$$recall = 100 \times \frac{0}{3} = 0\%$$

Metric: Recall

$$\boldsymbol{precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{|G|}}$$

Ground-truth data

User	Item	
$u1$	$i2$	✗
	$i3$	✓
$u2$	$i1$	✗
	$i2$	✓
	$i3$	✓
	$i5$	✓
	$i7$	✗
$u3$	$i3$	✗
	$i4$	✗
	$i6$	✗
$u4$	$i1$	✗
	$i2$	✗
	$i3$	✓
	$i4$	✓
	$i5$	✓
	$i6$	✓

?

Top-4 recommendation list

$u1$	$i1$, $i3$, $i4$, $i6$	$\boldsymbol{recall = 100 \times \frac{1}{2} = 50\%}$
$u2$	$i2$, $i3$, $i4$, $i5$	$\boldsymbol{recall = 100 \times \frac{3}{5} = 60\%}$
$u3$	$i1$, $i2$, $i5$, $i7$	$\boldsymbol{recall = 100 \times \frac{0}{3} = 0\%}$
$u4$	$i3$, $i4$, $i5$, $i6$	$\boldsymbol{recall = 100 \times \frac{4}{6} = 67\%}$

Metric: Recall

$$precision(n) = 100 \times \frac{|\mathcal{R}(n) \cap G|}{|G|}$$

Ground-truth data

u1	*i2*	✗
	i3	✓
u2	*i1*	✗
	i2	✓
	i3	✓
	i5	✓
	i7	✗
u3	*i3*	✗
	i4	✗
	i6	✗
u4	*i1*	✗
	i2	✗
	i3	✓
	i4	✓
	i5	✓
	i6	✓

?

Top-4 recommendation list

u1	*i1*	$recall = 100\times\frac{1}{2} = 50\%$
	i3	
	i4	
	i6	
u2	*i2*	$recall = 100\times\frac{3}{5} = 60\%$
	i3	
	i4	
	i5	
u3	*i1*	$recall = 100\times\frac{0}{3} = 0\%$
	i2	
	i5	
	i7	
u4	*i3*	$recall = 100\times\frac{4}{6} = 67\%$
	i4	
	i5	
	i6	

$$\overline{recall} \approx 44\%$$

Metric: Recall

- **Is recall enough to show the quality of recommendations?**
- **Recall value depends on the size of the recommendations**
  - Higher recall can be achieved by increasing the size of the recommendations
- **Consider the situation that the size of the recommendations is set to the number of items in the systems.**
  - This way, the recall would always be 100%.

- **Assume there are 7 items in the system and the recommendation system shows a recommendation list of size 7 to each user.**

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
	i2
	i5
	i7
u2	*i2*
	i3
	i4
	i5
	i1
	i6
	i7
u3	*i1*
	i2
	i5
	i7
	i3
	i4
	i6
u4	*i3*
	i4
	i5
	i6
	i1
	i2
	i7

- **Assume there are 7 items in the system and the recommendation system shows a recommendation list of size 7 to each user.**

Ground-truth data

u1	*i2*	✓
	i3	✓
u2	*i1*	✓
	i2	✓
	i3	✓
	i5	✓
	i7	✓
u3	*i3*	✓
	i4	✓
	i6	✓
u4	*i1*	✓
	i2	✓
	i3	✓
	i4	✓
	i5	✓
	i6	✓

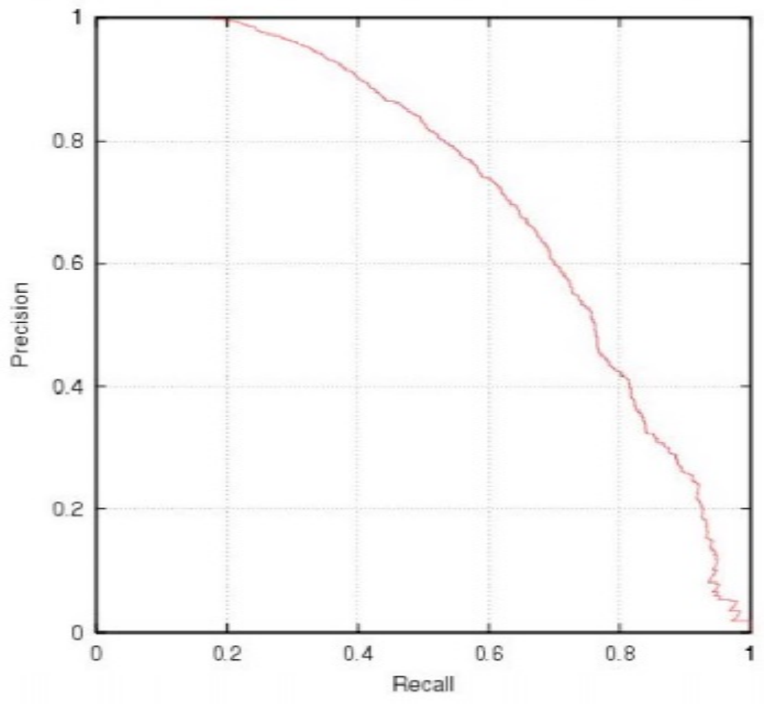
Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
	i2
	i5
	i7
u2	*i2*
	i3
	i4
	i5
	i1
	i6
	i7
u3	*i1*
	i2
	i5
	i7
	i3
	i4
	i6
u4	*i3*
	i4
	i5
	i6
	i1
	i2
	i7

$$\overline{recall} = 100\%$$

Precision vs. Recall

- **E.g., typically when a recommender system is tuned to increase precision, recall decreases as a result (or vice versa)**



Metric: F_1

- **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

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

- **It is a harmonic mean between precision and recall.**
- **It gives equal weight to precision and recall.**

Rank position matters

- Precision, Recall, and F_1, while showing the quality of recommendation list, does not show the ranking quality of list.

- For a user

Actually good
Item 10
Item 16

relevant

Recommended (predicted as good)
Item 13
Item 10
Item 18

Rank position matters

- **Rank metrics extend precision and recall 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

Metric: Normalized Discounted Cumulative Gain

- **Normalized Discounted Cumulative Gain (NDCG) is a metric of ranking quality or the relevance of the top listed products.**

- **The principle of NDCG is that the more relevant products must be ranked better than the irrelevant products.**

- **The higher NDCG indicates that the relevant products are ranked higher.**

Metric: Normalized Discounted Cumulative Gain

- **Normalized Discounted Cumulative Gain (NDCG)**

$$NDCG_K = \frac{DCG_K}{IDCG_K}$$

Metric: Normalized Discounted Cumulative Gain

- **Normalized Discounted Cumulative Gain (NDCG)**

$$NDCG_K = \frac{DCG_K}{IDCG_K}$$

- **Discounted Cumulative Gain (DCG)**

Size of the recommendation list

$$DCG_K = \sum_{i=1}^{K} \frac{relevance_i}{\log_2(i+1)}$$

Relevance of recommendation at position i

Metric: Normalized Discounted Cumulative Gain

- **Normalized Discounted Cumulative Gain (NDCG)**

$$NDCG_K = \frac{DCG_K}{IDCG_K}$$

- **Discounted Cumulative Gain (DCG)**

Size of the recommendation list

$$DCG_K = \sum_{i=1}^{K} \frac{relevance_i}{\log_2(i+1)}$$

Relevance of recommendation at position i

- **Idealized Discounted Cumulative Gain (IDCG)**
  - Assumption that items are ordered by decreasing relevance

$$IDCG_K = \sum_{i=1}^{min\{K,|G|\}} \frac{relevance_i}{\log_2(i+1)}$$

Metric: Normalized Discounted Cumulative Gain

- **Both DCG_K and $IDCG_K$ are multiplication of two terms:**
  - $relevance_i$: 1 if recommended item also is in test set, 0 otherwise
  - $\frac{1}{\log_2(i+1)}$: assigns weight to each position in the list, higher weight to the top positions

Position	$\log_2(i+1)$	$\frac{1}{\log_2(i+1)}$
1	1	1
2	1.585	0.63
3	2	0.5
4	2.322	0.43
5	2.585	0.38

Metric: Normalized Discounted Cumulative Gain

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	
	i3	
	i4	
	i5	
u3	*i1*	
	i2	
	i5	
	i7	
u4	*i3*	
	i4	
	i5	
	i6	

Metric: Normalized Discounted Cumulative Gain

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$DCG_4 = \frac{0}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} + \frac{0}{\log_2 5} = 0.63$$

Metric: Normalized Discounted Cumulative Gain

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$DCG_4 = \frac{0}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} + \frac{0}{\log_2 5} = 0.63$$

What is the ideal ranking for $u1$?

$u1$ liked two items ($i2$ and $i3$):

Metric: Normalized Discounted Cumulative Gain

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	
	i3	
	i4	
	i5	
u3	*i1*	
	i2	
	i5	
	i7	
u4	*i3*	
	i4	
	i5	
	i6	

$$DCG_4 = \frac{0}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} + \frac{0}{\log_2 5} = 0.63$$

What is the ideal ranking for $u1$?

$u1$ liked two items ($i2$ and $i3$):

$$IDCG_4 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} = 1.63$$

Metric: Normalized Discounted Cumulative Gain

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$DCG_4 = \frac{0}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} + \frac{0}{\log_2 5} = 0.63$$

What is the ideal ranking for $u1$?

$u1$ liked two items ($i2$ and $i3$):

$$IDCG_4 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} = 1.63$$

$$NDCG_4 = \frac{0.63}{1.63} \approx 0.39$$

Metric: Normalized Discounted Cumulative Gain

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	✓
	i3	✓
	i4	✗
	i5	✓
u3	*i1*	
	i2	
	i5	
	i7	
u4	*i3*	
	i4	
	i5	
	i6	

$$NDCG_4 = \frac{0.63}{1.63} \approx 0.39$$

Metric: Normalized Discounted Cumulative Gain

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	✓
	i3	✓
	i4	✗
	i5	✓
u3	*i1*	
	i2	
	i5	
	i7	
u4	*i3*	
	i4	
	i5	
	i6	

$$NDCG_4 = \frac{0.63}{1.63} \approx 0.39$$

$$DCG_4 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} + \frac{1}{\log_2 5} = 2.06$$

Metric: Normalized Discounted Cumulative Gain

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	✓
	i3	✓
	i4	✗
	i5	✓
u3	*i1*	
	i2	
	i5	
	i7	
u4	*i3*	
	i4	
	i5	
	i6	

$$NDCG_4 = \frac{0.63}{1.63} \approx 0.39$$

$$DCG_4 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} + \frac{1}{\log_2 5} = 2.06$$

What is the ideal ranking for $u2$?

Metric: Normalized Discounted Cumulative Gain

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	✓
	i3	✓
	i4	✗
	i5	✓
u3	*i1*	
	i2	
	i5	
	i7	
u4	*i3*	
	i4	
	i5	
	i6	

$$\boxed{NDCG_4 = \frac{0.63}{1.63} \approx 0.39}$$

$$DCG_4 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} + \frac{1}{\log_2 5} = 2.06$$

What is the ideal ranking for $u2$?

$u2$ liked five items:

Metric: Normalized Discounted Cumulative Gain

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*	✗
	i3	✓
	i4	✗
	i6	✗
u2	*i2*	✓
	i3	✓
	i4	✗
	i5	✓
u3	*i1*	
	i2	
	i5	
	i7	
u4	*i3*	
	i4	
	i5	
	i6	

$$NDCG_4 = \frac{0.63}{1.63} \approx 0.39$$

$$DCG_4 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} + \frac{1}{\log_2 5} = 2.06$$

What is the ideal ranking for $u2$?

$u2$ liked five items:

$$IDCG_4 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} + \frac{1}{\log_2 5} = 2.56$$

Metric: Normalized Discounted Cumulative Gain

Ground-truth data

u1	*i2*
	i3
u2	*i1*
	i2
	i3
	i5
	i7
u3	*i3*
	i4
	i6
u4	*i1*
	i2
	i3
	i4
	i5
	i6

?

Top-4 recommendation list

u1	*i1*
	i3
	i4
	i6
u2	*i2*
	i3
	i4
	i5
u3	*i1*
	i2
	i5
	i7
u4	*i3*
	i4
	i5
	i6

$$NDCG_4 = \frac{0.63}{1.63} \approx 0.39$$

$$DCG_4 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} + \frac{1}{\log_2 5} = 2.06$$

What is the ideal ranking for $u2$?

$u2$ liked five items:

$$IDCG_4 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} + \frac{1}{\log_2 5} = 2.56$$

$$NDCG_4 = \frac{2.06}{2.56} \approx 0.8$$

Non-Accuracy Metrics

- **Reference**
  - ***Recommender systems handbook***, Chapter 8.

- **Novelty**
  - Section 8.3.6
- **Serendipity**
  - Section 8.3.7
- **Diversity**
  - Section 8.3.8

Metric: Novelty

- **Novel recommendations are recommendations for items that the user did not know about.**

- **In a user study, it can be easily measured by asking users whether they were already familiar with a recommended item.**
  - But, in offline experiment, it is challenging.

Metric: Novelty

1. **Simulating the items that the user is familiar with, but did not report rating for.**
   - Split the data set on time, i.e. hide all the user ratings that occurred after a specific point in time
   - In addition, we can hide some ratings that occurred prior to that time
   - When recommending, the system is rewarded for each item that was recommended and rated after the split time, but would be punished for each item that was recommended but rated prior to the split time.

2. **Assuming that popular items are less likely to be novel.**
   - Novelty can be taken into account by using an accuracy metric where the system does not get the same credit for correctly predicting popular items as it does when it correctly predicts nonpopular items

Metric: Serendipity

- **Serendipity is a measure of how surprising the successful recommendations are.**
  - E.g., if the user has rated positively many movies where a certain star actor appears, recommending the new movie of that actor may be novel, because the user may not know of it, but is hardly surprising.

Metric: Serendipity

1. **Serendipity can be measured as the amount of relevant information that is new to the user in a recommendation.**
   - For example, if following a successful movie recommendation the user learns of a new actor that she likes, this can be considered as serendipitous.
   - One way for simulating this measurement is to manually label pairs of items as redundant.
   - Then, a recommendation is considered as serendipitous if it does not contain redundant items.

2. **To avoid human labeling, a distance measurement can be designed between items based on content.**
   - The successfulness of recommendation is scored by its distance from a set of previously rated items in a collaborative filtering system, or from the user profile in a content-based recommender.
   - Thus, the recommendation far from the user profile would be rewarded more.

Metric: Diversity

- **Diversity is generally defined as the opposite of similarity.**

- **When recommendations are not diverse, it may take longer to explore the range of items.**

- **Example: recommendation for a vacation**
  - Presenting a list with 5 recommendations, all for the same location, varying only on the choice of hotel, or the selection of attraction, may not be as useful as suggesting 5 different locations.
  - The user can view the various recommended locations and request more details on a subset of the locations that are appropriate to her.

Metric: Diversity

- **The most explored method for measuring diversity uses item-item similarity, typically based on item content.**
  - We could measure the diversity of a list based on the sum, average, min, or max distance between item pairs
  - Or measure the value of adding each item to the recommendation list as the new item's diversity from the items already in the list.

- **The item-item similarity measurement used in evaluation can be different from the similarity measurement used by the algorithm that computes the recommendation lists.**
  - For example, we can use for evaluation a costly metric that produces more accurate results than fast approximate methods that are more suitable for online computations.

Evaluating Recommender Systems

Masoud Mansoury
AMLab, University of Amsterdam
Discovery Lab, Elsevier