Recommender systems

EVALUATION METRICS

Nguyen Ngoc Thao
nnthao@fit.hcmus.edu.vn

Evaluating recommender systems

- Evaluations can be offline or online.

- The target algorithm is trained on a majority of ratings.
- It predicts the ratings for the rest of data (unseen).
- One or several metrics are used to assess the quality of recommendations.

- Users are recommended items in a live environment.
- Their satisfactions are measured using feedback or acts implicitly tracked (e.g., click-through rate).

Evaluating recommender systems

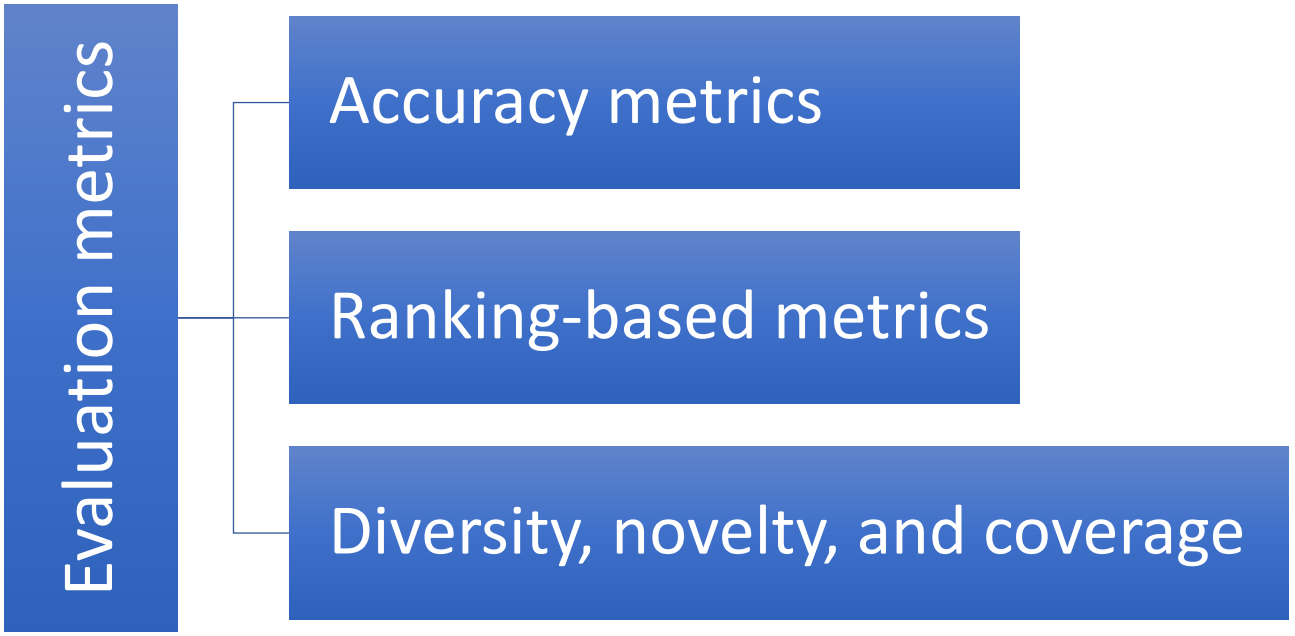
- Evaluations can be offline or online.

- Advantage: quick
- Drawbacks: unable to measure the true satisfaction of users regarding the recommended items

- Advantage: able to measure the true satisfaction
- Drawbacks: expensive and often impossible

Evaluating recommender systems

- There are different perspectives on evaluation metrics.
- Some are based on the recommendation list itself.
  - E.g., accuracy, coverage, diversity and novelty, etc.
- Some are based on the point of views of the system or the users, which is independent of the recommender.
  - E.g., confidence, robustness, adaptivity, scalability, trust, risk and privacy, etc.
- These metrics may not well reflect the users' satisfaction.
- Besides, the users' satisfaction may not be the ultimate goal of RS in some cases.



Evaluation scenario

- Each algorithm first produces the predicted ratings.
- The results are then sorted, and for each user, the top-N items with the highest predicted ratings are recommended.
- The metrics evaluate different properties of these top-N items.
- *The only available information is the rating histories.*

Accuracy metrics

Predictive accuracy metrics

- These metrics are mostly useful where the predicted ratings are shown to the users of system.
- They have easy implementation due to their simplicity

- However, these metrics consider the ratings' space to be uniform, which is not the case in real systems.
- All ratings are treated the same, regardless of their position in the recommendation list.
  - A one-star error for an item on the top gives the same penalty as a one-star error for an item at the end.

Predictive accuracy metrics: MAE ↓

- Mean absolute error (MAE) computes the average absolute deviation of the predicted ratings from the real ratings.

$$\boldsymbol{MAE} = \frac{\mathbf{1}}{|\boldsymbol{\mathcal{T}}|} \sum_{(\boldsymbol{u},\boldsymbol{i}) \in \boldsymbol{\mathcal{T}}} |\tilde{\boldsymbol{r}}_{\boldsymbol{ui}} - \boldsymbol{r}_{\boldsymbol{ui}}|$$

  - $\tilde{r}_{ui}$ and r_{ui} are the predicated rating of the system and the real rating for user u and item i, respectively. $|\mathcal{T}|$ is the size of the test set $\mathcal{T}$.

- The normalized MAE ($\in [0,1]$) can compare RS systems in different rating scales.

$$\boldsymbol{NMAE} = \frac{\boldsymbol{MAE}}{\boldsymbol{r}_{\boldsymbol{max}} - \boldsymbol{r}_{\boldsymbol{min}}}$$

  - r_{max} and r_{min} are the maximum and minimum ratings of all users.

Predictive accuracy metrics: RSME ↓

- Root mean square error (RMSE) is a variation of MAE which puts more emphasis on large errors.

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

- RMSE is more sensitive to observations that are further from the mean.
- The normalized RMSE ($\in [0,1]$) is defined as

$$\boldsymbol{NRMSE} = \frac{\boldsymbol{RMSE}}{\boldsymbol{r}_{\boldsymbol{max}} - \boldsymbol{r}_{\boldsymbol{min}}}$$

MAE and RMSE: An example

- The following table shows the predicted ratings from the RS model (top) and the actual ratings that a user has given to a list of items (below).
- Note that the scores are in range [1, 100].

Items	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Predicted	14	15	18	19	25	18	12	12	15	22
Actual	12	15	20	16	20	19	16	20	16	**76**

- MAE: 8
- RMSE: 16.4356

- Notice that the RMSE increases much more than the MAE since the last actual rating is a clear outlier.

Rank accuracy metrics: Pearson ↑

- Pearson correlation ($\in [-1,1]$) finds the linear relationship between two lists of predicted ratings and real ones of a user.

$$PC_u = \frac{\sum_{i=1}^{N}(r_{ui} - \overline{r_u})(\tilde{r}_{ui} - \overline{\tilde{r}_u})}{\sqrt{\sum_{i=1}^{N}(r_{ui} - \overline{r_u})^2}\sqrt{\sum_{i=1}^{N}(\tilde{r}_{ui} - \overline{\tilde{r}_u})^2}}$$

  - $\overline{r_u}$ and $\overline{\tilde{r}_u}$ are the average actual rating and average predicted rating of user u, respectively.
  - N is the number of items.

PC values	
	• $PC = 1$: a perfect positive linear relationship
	• $PC = 0$: no linear relationship
	• $PC = -1$: a perfect negative linear relationship

Rank accuracy metrics: Spearman ↑

- Spearman's rank correlation ($\in [-1,1]$) finds the statistical dependence between the predicted and real lists of a user.

$$SC_u = \frac{\sum_{i=1}^{N}(R_{ui} - \overline{R_u})\left(\tilde{R}_{ui} - \overline{\tilde{R}_u}\right)}{\sqrt{\sum_{i=1}^{N}(R_{ui} - \overline{R_u})^2}\sqrt{\sum_{i=1}^{N}\left(\tilde{R}_{ui} - \overline{\tilde{R}_u}\right)^2}}$$

  - R_{ui} and $\tilde{R}_{ui}$ are the ranks of items i in the real and predicted lists.

SC values

- $SC = 1$: a perfect monotonic relationship where ranks of both variables are perfectly aligned.
- $SC = 0$: no monotonic relationship.
- $SC = -1$: a perfect monotonic relationship in the opposite direction.

Pearson and Spearman: An example

- The following table shows the predicted ratings from the RS model (top) and the actual ratings that a user has given to a list of item (below).
- Note that the scores are in range [1, 50].

Item	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
Actual	43	42	40	36	33	31	30	29	27	16
Predicted	12	6	17	7	29	50	27	28	20	2

- Pearson correlation: -0.0702
- Spearman correlation: -0.17576

Item	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
Actual rank	1	2	3	4	5	6	7	8	9	10
Predicted rank	7	9	6	8	2	1	4	3	5	10

Spearman correlation: Notes

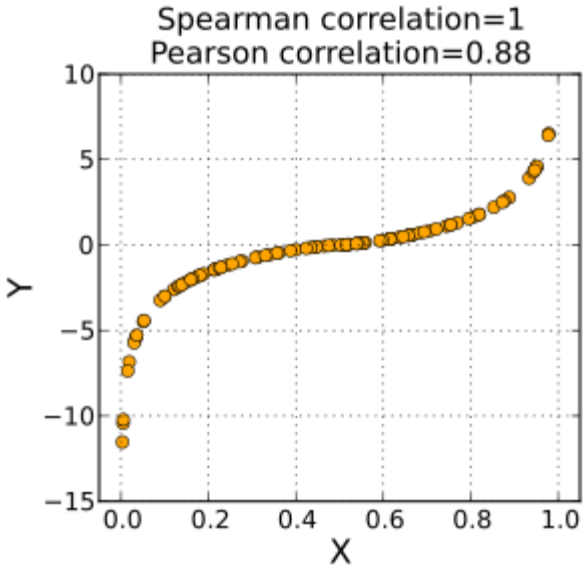
- Only if all n ranks are distinct integers, Spearman rank's correlation can be computed using the popular formula.

$$\boldsymbol{SC_u = 1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N(N^2 - 1)}}$$

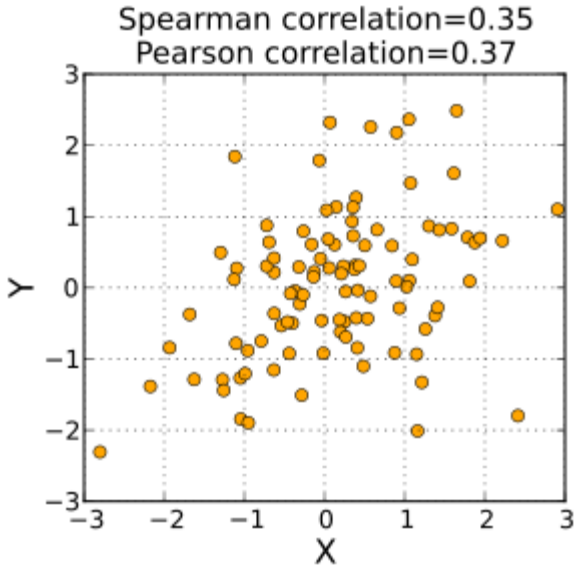
  - $d_i = \tilde{R}_{ui} - R_{ui}$ is the different between two ranks of each item.
  - N is the number of items.

- Spearman's rank correlation is the Pearson correlation of the rank vectors.

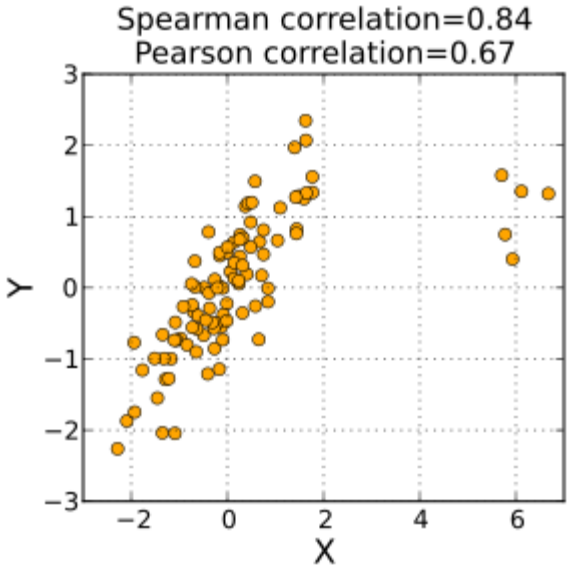
Pearson vs. Spearman correlation



A Spearman correlation of 1 results when the two variables are monotonically related, even if their relationship is not linear. In contrast, this does not give a perfect Pearson correlation.



When the data are roughly elliptically distributed and there are no prominent outliers, the Spearman correlation and Pearson correlation give similar values.



The Spearman correlation is less sensitive than the Pearson correlation to strong outliers that are in the tails of both samples. That is because Spearman's ρ limits the outlier to the value of its rank.

Image credit: Wikipedia

Rank accuracy metrics: Kendall Tau ↑

- Kendall rank correlation ($\in [-1,1]$) finds the similarity of the orderings of two ranked lists.

$$\boldsymbol{\tau = \frac{C - D}{C + D} = \frac{C - D}{N(N+1)/2}}$$

  - C is the number of Concordant pairs—two items for which the recommender predicts in the same order as real rating list.
  - D is the number of Discordant pairs—two items for which the system predicted the wrong order.

τ values	
	• $\tau = 1$ (when $D = 0$): the two lists are completely the same. • $\tau = -1$ (when $C = 0$): the two lists are completely dissimilar.

Kendall Tau: An example

- The following table shows the predicted ranking from the RS model (top) and the actual ranking that a user has given to a list of items (below).

Item	I1	I2	I3	I4	I5	I6
Predicted	3	1	4	2	6	5
Actual	1	2	3	4	5	6

- In the predicted list, there are 11 Concordant pairs and 4 Discordant pairs.
  - Concordant pairs: 3-4, 3-6, 3-5, 1-4, 1-2, 1-6, 1-5, 4-6, 4-6, 2-6, 2-5
  - Discordant pairs: 3-1, 3-2, 4-2, 6-5
- $\tau = \frac{11-4}{11+4} = 0.47$

Variants of Kendall Tau correlation

- Sometimes a user gives several items the same ratings or a recommender predicts the same ratings for several items.
- There is a variant of Kendall Tau correlation for such cases.

$$\boldsymbol{\tau} = \frac{\boldsymbol{C} - \boldsymbol{D}}{\sqrt{(\boldsymbol{C} + \boldsymbol{D} + \boldsymbol{TR})(\boldsymbol{C} + \boldsymbol{D} + \boldsymbol{TP})}}$$

- There can be several groups of ties in each list of ratings.
  - TR is the number of item with the same real ratings.

$$TR = \sum_i tr_i(tr_i - 1)/2$$

    - tr_i is the number of tied values in the i^{th} group in the real rating list.
  - TP is the number of items with the same predicted ratings.

Variants of Kendall Tau correlation

- Kendall Tau correlation does not consider the position of the correct ranked items.
  - E.g., between the first and the second rating and between 50th and 51st ratings.
- One solution is to add more weight to concordant pairs at the top of the list than those towards the end of the list.

Rank accuracy metrics: NDPM ↓

- Normalized distance-based performance measure (NDPM) compares two weakly ordered ranked list.

$$\boldsymbol{NDPM = \frac{2D - CU}{2NP}}$$

  - D is the number of discordant pairs, CU is the number of pairs for which one system gives a tie and the other ranked list does not, and NP is the total number of pairs in the real ranked list minus tied ones.

- NDPM only penalizes the system for tied pairs of predicted list for which one item is strictly preferred in the real list.
  - The modified Kendall Tau correlation penalizes the system even when there are tied pairs in the real ranked list.

Classification accuracy metrics

- These metrics counts how many times the system classifies a relevant item as a good one or an irrelevant item as a bad one.
- An item is relevant to a user, if he has rated it (more than the average rating of the total items he voted).
- The disliked and non-rated items are often grouped together.

Classification accuracy metrics: P–R ↑

- Precision and recall have been among the first series of the metrics used to evaluate recommendation algorithms.
- Consider a confusion matrix that divides items into 4 groups.

		The recommender predicts as	
		Recommended	Not recommended
Actual relevance	Relevant	TP	FN
	Irrelevant	FP	TN

Precision and Recall: General formulas

- Assume that every items are rated by the user.
- Precision calculates the ratio of the relevant items which are recommended to the number of all recommended items.

$$P = \frac{TP}{TP + FP}$$

- Recall is the ratio of the relevant items recommended to the number of all relevant items.

$$R = \frac{TP}{TP + FN}$$

- Both are in range of [0, 1].

Precision and Recall: for N items

- Actual usage of precision and recall relaxes the assumption.
- Let N be the size of the recommendation list.
- Precision and Recall for the limited list can be defined as

$$\boldsymbol{P@N_u = \frac{TP}{N}}$$

$$\boldsymbol{R@N_u = \frac{TP}{|Rel_u|}}$$

  - Rel_u is the set of items relevant to user u.

- Obviously, $P@N_u$ and $R@N_u$ are dependent on the length of the recommendation list.

Classification accuracy metrics: F1 ↑

- Precision and recall are inversely correlated, yet we need to consider both in the evaluation.
- F1 score ($\in [0,1]$) is defined as a combination of precision and recall.

$$\boldsymbol{F1_u = \frac{2 \cdot (P@N_u) \cdot (R@N_u)}{(P@N_u) + (R@N_u)}}$$

Precision and Recall: An example

- The following table show a list of 20 items and their relevancies (+: relevant or –: irrelevant). Assume that we only consider the first 10 ranks.

Rank	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
+/-	**+**	**+**	**+**	**–**	**+**	**–**	**+**	**–**	**+**	**+**	**–**	**–**	**+**	**–**	**–**	**–**	**–**	**–**	**–**	**–**

- N = 10
- $Rel_u = 8$
- P@10 = 7 / 10 = 0.7
- R@10 = 7 / 8 = 0.88
- F1@10 = 2 · 0.7 · 0.88 / (0.7 + 0.88) = 0.8

Rank-based metrics

Rank-based metrics

- These metrics examine the order and position of the items displayed to the user in the recommendation list.
- They do not compare the exact value of the predicted ratings with the real ones.

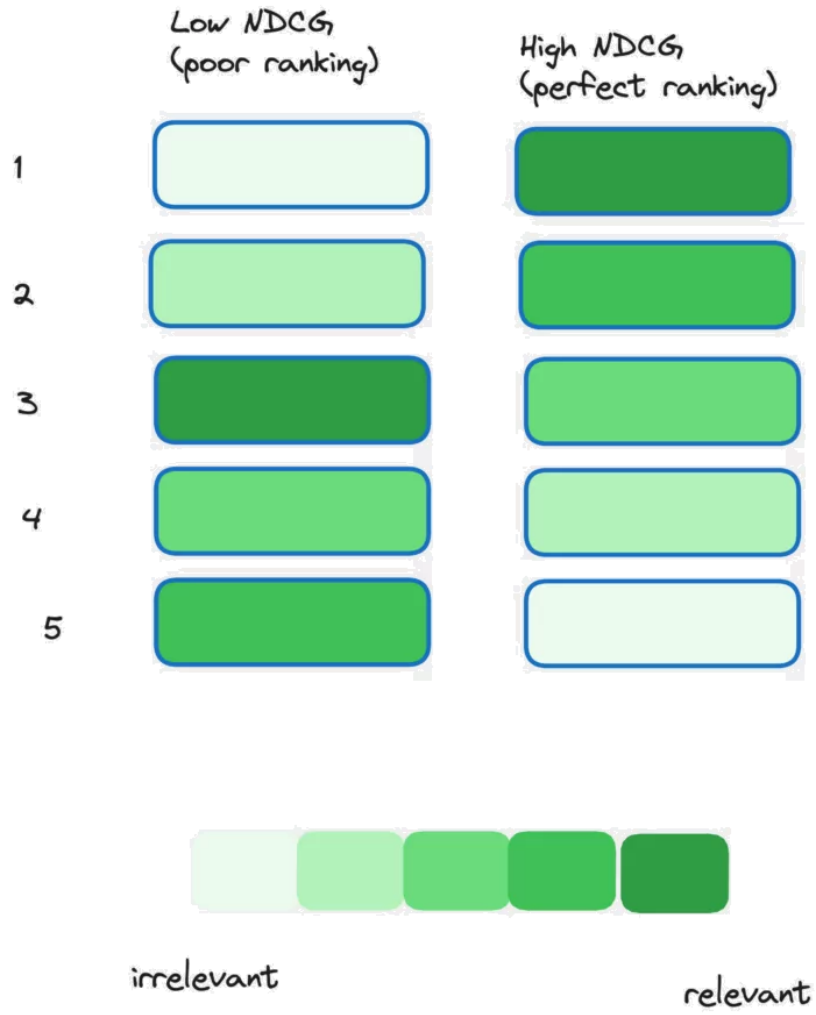


Image credit: aporia

Rank-based metrics: Half-life utility ↑

- **Assumption:** As the rank of the item in the recommendation list decreases, the probability of user's tendency to examine it reduces exponentially.
- Half-life utility is defined as $\boldsymbol{H_u = \sum_{i=1}^{N} \frac{\max(r_{ui} - d, 0)}{2^{(i-1)(h-1)}}}$
  - h is the half-life threshold (empirically set to 5).
    - Users usually tend to pay attention to items at the top of the list.
    - It is the rank of item on the list for which there is a 50-50 chance that user u will examine it.
  - d is the neutral vote (i.e., the user's average rating).

Rank-based metrics: DCG ↑

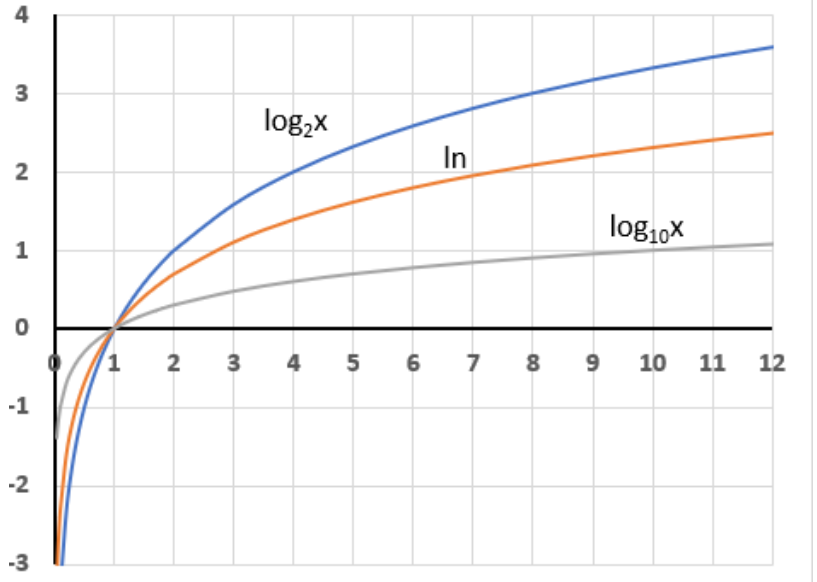
- Discounted Cumulative Gain (DCG) scores the usefulness of an item based on its rank in a recommended list.

$$\boldsymbol{DCG_u^b = \sum_{i=1}^{b-1} Re_i + \sum_{i=b}^{N} \frac{Re_i}{\log_b(R_i + 1)}}$$

  - Re_i is the relevancy of item i (with rank r_i) in the recommendation list
  - E_i is the rank of the item in the list. $R_i = 1$ for the first item at the top of the list.
  - b is the base of the logarithm ($b \geq 2$, empirically set to 2).
- The more relevant items are with higher ranks, the more valuable the recommendation list is for the user.

DCG: Explaining the parameters

- The lower is the rank of the item R_i in the list (i.e., toward the end of the list), its share in the cumulated gain decreases.
  - The discounting function to reduce this share could be log-harmonic.
- The parameter b controls the degree of reduction in items' shares in DCG.
  - The greater the values of b, the slower the shares decrease.



- If the item is totally relevant, $Re_i = 1$. Otherwise, $Re_i = 0$.

Rank-based metrics: NDCG ↑

- Normalized discounted cumulative gain (NGCG) ($\in [0,1]$) normalizes DCG value eliminate the effect of different sizes of the lists.

$$\boldsymbol{NDCG_u = \frac{DCG_u}{DCG_{max}}}$$

  - DCG_{max} is the exact order of items given by the user in real rating list

NDCG values	
	• $NDCG = 1$: The perfect ranking, all relevant items are ranked at the top. • $NDCG = 0$: The worst possible ranking, where there are no relevant items in the top positions.

NDCG: An example

- Each of the following six items is judged on a scale of 0-3 (0: irrelevant, 3: highly relevant). They are ordered by the RS algorithm as follows.

Item	I1	I2	I3	I4	I5	I6
Predicted rank	1	2	3	4	5	6
Relevancy	3	2	3	0	1	2

- Let b = 2.
- DCG = 3 + 2/log2(2+1) + 3/log2(3+1) + 0/log2(4+1) + 1/log2(5+1) + 2/log2(6+1)

  = 3 + 1.262 + 1.5 + 0 + 0.387 + 0.712 = 6.861.
- Sorting all relevant items by their relative relevance gives the DCG_{max} = 7.141.
- NDCG = 6.861 / 7.141 = 0.961.

Rank-based metrics: RBP ↑

- The Rank-biased Precision (RBP) metric is defined as

$$\boldsymbol{RBP_u = (1-p)\sum_{i=1}^{N}\left(Re_i \times p^{R_i-1}\right)}$$

  - R_i and Re_i are defined the same as for DCG.

- RPB is between 0 and 1 since $\sum_{i=1}^{\infty} p^{R_i-1} = 1/(1-p)$.
  - The greater the value, the better the system performs.

An explanation of RBP

- RBP gives more shares to highly ranked relevant items.
- Its discounting function is a geometric sequence instead of the logarithmic one in DCG.
- **Underlying assumption:** Users often examine the first item, and then, with a probability p, they may check the next item.
  - E.g., $p = 0.8$, the user checks the first items, then checks the second item with probability of 0.8, the third item with probability of 0.8^2, etc. and finally the i^{th} item with probability of 0.8^{i-1}.
- Small p: the user only examines the top-ranked items
- Large p : it may also examine the items in lower ranks.

Rank-based metrics: Recovery rate ↓

- Recovery Rate evaluates the performance based on the correct ranking of the items.

$$\boldsymbol{Recovery_u = \frac{1}{|N_R|} \sum_{i \in N_R} \frac{r_i}{C_i}}$$

  - N_R is the set of relevant items in the real rating list of user u.
  - $\boldsymbol{r_i}$ indicates the rank of item i in the predicted list, and $\boldsymbol{C_i}$ is the number of candidate items to recommend to user u.

Diversity, Novelty, and Coverage

Diversity, Novelty, and Coverage

- There is a growing focus on recommenders that balance accuracy with user satisfaction.
- Indeed, in many real applications, the users would like to be recommended diverse and novel items.

- These metrics measure how much a recommender is novel, diverse or covers the items available in the system.

Diversity: Intra-diversity ↓

- Intra-diversity measures how different are the items offered to a user, i.e., the diversity of each recommendation list.

$$\boldsymbol{IntraDiversity_u = \frac{1}{N(N-1)} \sum_{i \neq j} s(I_i, I_j)}$$

  - $s(I_i, I_j)$ is the similarity between items i and j.
    - This can be obtained from the content information of items or using similarity measures (e.g., cosine similarity) for item rating vectors.

- The lower the value, the more diverse items the system recommends to the user.

Diversity: Inter-diversity ↓

- Inter-diversity indicates the extent of difference between the recommendation lists of all users .

$$H_{uu'} = 1 - \left(\frac{Q_{uu'}}{N} \right)$$

  - $Q_{uu'}$ is the similarity between the two recommendation lists of user u and u', which can be estimated using Hamming distance.

- The average metric can be calculated as

$$H_u = \frac{1}{m(m-1)} \sum_{u=1}^{m} \sum_{u \neq u'} H_{uu'}$$

Novelty metrics: Popularity ↓

- Many users would also want to get recommendations for items they have not yet seen.
  - If the novel suggestions get users' attention and they like them, the recommender is successful from this point of view.
- A popular item may be less novel to a user since he more probably experiences it.
- The popularity of an item can be evaluated by its degree.

$$\boldsymbol{Popularity} = \frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}} \boldsymbol{d_i}$$

  - d_i is the degree of item i.

Novelty metrics: SIBN and ESIBN ↑

- Self-Information Based Novelty (SIBN) is known as surprisal or expectedness, considering popular items less novel.
- SIBN for an item i, SIBN, is defined as follows.

$$\boldsymbol{SIBN_i = \log_2\left(\frac{m}{d_i}\right)}$$

- Effective novelty only considers the novelty of items relevant to each user from the top-N recommendation list

$$\boldsymbol{ESIBN_u = \sum_{i=1}^{N} Re_i \cdot SIBN_i}$$

  - It is logical to consider only the novelty of those items which the user would like and consume.

Coverage: Catalog coverage ↑

- A good recommended list not only includes likeable items to users, but also covers a wide variety of items.
- Catalog coverage refers to the percentage of distinct items in the top-N recommendation lists of users.

$$\boldsymbol{C = \frac{I_r}{N}}$$

  - N is the total number of items in the system and I_r indicates the total number of distinct items in the users' top-N lists.
  - We may consider only the items which are relevant to a user.

Coverage: Entropy coverage ↑

- The frequency of offering different items may vary.
  - Some items are frequently recommended to various users (probably popular ones), and some appears less in the lists.
- Entropy coverage considers how many times an item is recommended.

$$EC = -\sum_{i=1}^{m} p_i \log_2 p_i$$

  - p_i is the percentage of the recommendation lists that contains item i.

Unified evaluation metric ↑

- One can choose a few metrics and properly combine them to have a unified evaluation metric.
- For example, we pick diversity, novelty, coverage, precision, and RBP. Then, the formula is defined as follows.

$$\boldsymbol{UM}(\overline{\boldsymbol{H}}, \overline{\boldsymbol{N}}, \overline{\boldsymbol{C}}, \overline{\boldsymbol{P}}, \overline{\boldsymbol{RBP}}) = \frac{\mathbf{5}}{\frac{\mathbf{1}}{\overline{\boldsymbol{H}}} + \frac{\mathbf{1}}{\overline{\boldsymbol{N}}} + \frac{\mathbf{1}}{\overline{\boldsymbol{C}}} + \frac{\mathbf{1}}{\overline{\boldsymbol{P}}} + \frac{\mathbf{1}}{\overline{\boldsymbol{RBP}}}}$$

- $\overline{H}, \overline{N}, \overline{C}, \overline{P}$, and $\overline{RBP}$ are the normalized values for the above metrics.

Q1. MAE vs. RMSE

- The following table shows the predicted ratings from the RS model (top) and the actual ratings that a user has given to a list of items (below).
- Note that the scores are in range [1, 10].

Items	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Predicted	4	7	6	5	7	3	5	7	4	3
Actual	3	6	3	6	7	5	4	5	7	5

- Calculate the errors using the following metrics: MAE, NMAE, RMSE, and NRMSE.

Q2. Pearson vs. Spearman

- The following table shows the predicted ratings from the RS model (top) and the actual ratings that a user has given to a list of items (below).
- Note that the scores are in range [1, 10].

Items	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Predicted	4	7	6	5	7	3	5	7	4	3
Actual	3	6	3	6	7	5	4	5	7	5

- Calculate the values of Pearson correlation coefficient and Spearman rank's correlation.

Q3. Kendall Tau

- The following table shows the predicted ratings from the RS model (top) and the actual ratings that a user has given to a list of items (below).
- Note that the scores are in range [1, 10].

Items	I1	I2	I3	I4	I5	I6
Predicted	4	7	6	5	3	8
Actual	3	6	4	7	5	9

- Calculate the (original) Kendall Tau correlation.

Q4. Precision, Recall, and F1

- The following table show 9 items and their relevancies (+: relevant or –: irrelevant).

Rank	1	2	3	4	5	6	7	8	9
+/-	+	–	+	+	–	–	–	+	–

- Calculate the values of Precision, Recall, and F1 at every rank position.

Q5. NGCG

- The following table shows the predicted ratings from the RS model (top) and the actual ratings that a user has given to a list of items (below).
- Note that the scores are in range [1, 5].

Items	**I1**	**I2**	**I3**	**I4**
Predicted	3.23	2.13	3.12	4.58
Actual	2	1	4	5

- Calculate the NDCG value.

List of references

- Jalili, Mahdi, Sajad Ahmadian, Maliheh Izadi, Parham Moradi, and Mostafa Salehi. "Evaluating collaborative filtering recommender algorithms: a survey." IEEE Access 6 (2018): 74003-74024.


THE END

Q5. NDCG

	i1	i2	i3	i4
$r_{u,i}$	2	1	4	5
$\hat{r}_{u,i}$	3.23	2.13	3.12	4.58

- $p_1 = i_4,\ p_2 = i_1,\ p_3 = i_3,\ p_4 = i_2$
- $r_{u,p_1} = 5,\ r_{u,p_2} = 2,\ r_{u,p_3} = 4,\ r_{u,p_4} = 1$
- $DCG_4 = 5 + \frac{2}{log_2 2} + \frac{4}{log_2 3} + \frac{1}{log_2 4} = 10.0237$
- $IDCG_4 = 5 + \frac{4}{log_2 2} + \frac{2}{log_2 3} + \frac{1}{log_2 4} = 10.7619$
- $NDCG_4 = \frac{DCG_4}{IDCG_4} = 0.9314$