Part 2: Evaluation

Robert Kwieciński

Adam Mickiewicz University

March 27, 2021

Evaluation measures

Based on the context we have to choose correct evaluation measure. Main types are:

- offline metrics:
  - RMSE (Root mean square error),
  - Precision at k,
  - Recall (at k),
  - F-score,
  - NDCG (Normalized discounted cumulative gain),
  - MRR (Mean reciprocal rank),
  - LAUC (Limited area under the curve),
  - HR (Hit rate),
  - Shannon entropy,
  - Gini index
- online metrics (A/B tests):
  - % of users who stopped using product,
  - click-through rate,
  - % of users performed given action (bought/rated/watched),
- user studies - ask users which recommender system seems better to them.

RMSE and MAE

Root mean square error

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(r_{ui} - \hat{r}_{ui})^2}{n}},$$

where we sum over all ratings in the test set.

RMSE and MAE

Root mean square error

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(r_{ui} - \hat{r}_{ui})^2}{n}},$$

where we sum over all ratings in the test set.

Mean absolute error

$$\text{MAE} = \frac{\sum_{i=1}^{n}|r_{ui} - \hat{r}_{ui}|}{n},$$

where we sum over all ratings in the test set.

Toy example

```
Training data:

matrix([[3, 4, 0, 0, 5, 0, 0, 4],
        [0, 1, 2, 3, 0, 0, 0, 0],
        [0, 0, 0, 5, 0, 3, 4, 0]], dtype=int64)

Test data:

matrix([[0, 0, 0, 0, 0, 0, 3, 0],
        [0, 0, 0, 0, 5, 0, 0, 0],
        [5, 0, 4, 0, 0, 0, 0, 2]], dtype=int64)

Recommendations:
```

	0	1	2	3	4	5	6
0	0	30	4.375000	60	4.375000	50	3.375000
1	10	40	4.166667	60	3.166667	70	3.166667
2	20	40	5.333333	70	4.333333	0	3.333333

Figure: Source: Notebook P2. Evaluation - toy example

User 0: 0, 1, 0 - 1 rating in the test set
User 1: 1, 0, 0 - 1 rating in the test set
User 2: 0, 1, 1 - 3 ratings in the test set

Precision at k

Precision at k

The percentage of relevant (present in the test set) items within top k recommendations.

Toy example

User 0: 0, 1, 0 - 1 rating in the test set
User 1: 1, 0, 0 - 1 rating in the test set
User 2: 0, 1, 1 - 3 ratings in the test set

User 0: Precision@3= $\frac{1}{3}$
User 1: Precision@3= $\frac{1}{3}$
User 2: Precision@3= $\frac{2}{3}$

Depending on business purpose we aggregate results from each user to overall score by taking average/weighted average/median.

Most common is taking the average:

Overall precision@3= $\frac{\frac{1}{3}+\frac{1}{3}+\frac{2}{3}}{3} = \frac{4}{9}$.

Recall at k

Recall at k

The percentage of relevant (present in the test set) items covered by top k recommendations.

Toy example

User 0: 0, 1, 0 - 1 rating in the test set
User 1: 1, 0, 0 - 1 rating in the test set
User 2: 0, 1, 1 - 3 ratings in the test set

User 0: Recall@3$= 1$
User 1: Recall@3$= 1$
User 2: Recall@3$= \frac{2}{3}$

Overall recall@3$= \frac{1+1+\frac{2}{3}}{3} = \frac{8}{9}$.

F-score

F-score (at k)

$$F_\beta = \frac{(1+\beta^2)\cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

$$F_1 = \frac{2 \cdot \frac{4}{9} \cdot \frac{8}{9}}{\frac{4}{9} + \frac{8}{9}} = \frac{16}{27}.$$

F-score

F-score (at k)

$$F_\beta = \frac{(1+\beta^2)\cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

$F_1 = \frac{2 \cdot \frac{4}{9} \cdot \frac{8}{9}}{\frac{4}{9} + \frac{8}{9}} = \frac{16}{27}$.

But it makes sense also this way:

User 0: $F_1 = \frac{2 \cdot \frac{1}{3} \cdot 1}{\frac{1}{3} + 1} = \frac{1}{2}$

User 1: $F_1 = \frac{2 \cdot \frac{1}{3} \cdot 1}{\frac{1}{3} + 1} = \frac{1}{2}$

User 2: $F_1 =$

F-score

F-score (at k)

$$F_\beta = \frac{(1+\beta^2)\cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

$F_1 = \frac{2 \cdot \frac{4}{9} \cdot \frac{8}{9}}{\frac{4}{9} + \frac{8}{9}} = \frac{16}{27}$.

But it makes sense also this way:

User 0: $F_1 = \frac{2 \cdot \frac{1}{3} \cdot 1}{\frac{1}{3} + 1} = \frac{1}{2}$

User 1: $F_1 = \frac{2 \cdot \frac{1}{3} \cdot 1}{\frac{1}{3} + 1} = \frac{1}{2}$

User 2: $F_1 = \frac{2 \cdot \frac{2}{3} \cdot \frac{2}{3}}{\frac{2}{3} + \frac{2}{3}} = \frac{2}{3}$

Overall $F_1 = \frac{\frac{1}{2} + \frac{1}{2} + \frac{2}{3}}{3} = \frac{5}{9}$.

F-score

F-score (at k)

$$F_\beta = \frac{(1+\beta^2)\cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

$F_1 = \frac{2 \cdot \frac{4}{9} \cdot \frac{8}{9}}{\frac{4}{9} + \frac{8}{9}} = \frac{16}{27}$.

But it makes sense also this way:

User 0: $F_1 = \frac{2 \cdot \frac{1}{3} \cdot 1}{\frac{1}{3} + 1} = \frac{1}{2}$

User 1: $F_1 = \frac{2 \cdot \frac{1}{3} \cdot 1}{\frac{1}{3} + 1} = \frac{1}{2}$

User 2: $F_1 = \frac{2 \cdot \frac{2}{3} \cdot \frac{2}{3}}{\frac{2}{3} + \frac{2}{3}} = \frac{2}{3}$

Overall $F_1 = \frac{\frac{1}{2} + \frac{1}{2} + \frac{2}{3}}{3} = \frac{5}{9}$.

If β increases then F_β is more like recall or like precision?

F-score

F-score (at k)

$$F_\beta = \frac{(1+\beta^2)\cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

$F_1 = \frac{2 \cdot \frac{4}{9} \cdot \frac{8}{9}}{\frac{4}{9}+\frac{8}{9}} = \frac{16}{27}$.

But it makes sense also this way:

User 0: $F_1 = \frac{2 \cdot \frac{1}{3} \cdot 1}{\frac{1}{3}+1} = \frac{1}{2}$

User 1: $F_1 = \frac{2 \cdot \frac{1}{3} \cdot 1}{\frac{1}{3}+1} = \frac{1}{2}$

User 2: $F_1 = \frac{2 \cdot \frac{2}{3} \cdot \frac{2}{3}}{\frac{2}{3}+\frac{2}{3}} = \frac{2}{3}$

Overall $F_1 = \frac{\frac{1}{2}+\frac{1}{2}+\frac{2}{3}}{3} = \frac{5}{9}$.

If β increases then F_β is more like recall or like precision?

$$\lim_{\beta \to \infty} F_\beta = \frac{(1+\beta^2)\cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} = \frac{(\frac{1}{\beta^2}+1)\cdot \text{precision} \cdot \text{recall}}{\text{precision} + \frac{\text{recall}}{\beta^2}} = \text{recall}$$

$$\lim_{\beta \to 0^+} F_\beta = \text{precision}$$

NDCG - Normalized discounted cumulative gain

NDCG at k

Denote by rel_i relevance of i-th recommended item (in our case 0 or 1).
Then: $\text{NDCG}_k = \frac{\text{DCG}_k}{\text{IDCG}_k}$,
where $\text{DCG}_k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{\log_2(i+1)}$ and IDCG_k is DCG_k of a perfect recommender system.

Toy example

User 0: 0, 1, 0 - 1 rating in the test set
User 1: 1, 0, 0 - 1 rating in the test set
User 2: 0, 1, 1 - 3 ratings in the test set

User 0: $\text{DCG}_3 = \frac{1}{\log_2 3} \approx 0.63$, $\text{IDCG}_3 = 1$, $\text{NDCG}_3 \approx 0.63$.
User 1: $\text{NDCG}_3 = 1$
User 2: $\text{DCG}_3 =$

NDCG - Normalized discounted cumulative gain

NDCG at k

Denote by rel_i relevance of i-th recommended item (in our case 0 or 1).
Then: $\text{NDCG}_k = \frac{\text{DCG}_k}{\text{IDCG}_k}$,
where $\text{DCG}_k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{\log_2(i+1)}$ and IDCG_k is DCG_k of a perfect recommender system.

Toy example

User 0: 0, 1, 0 - 1 rating in the test set
User 1: 1, 0, 0 - 1 rating in the test set
User 2: 0, 1, 1 - 3 ratings in the test set

User 0: $\text{DCG}_3 = \frac{1}{\log_2 3} \approx 0.63$, $\text{IDCG}_3 = 1$, $\text{NDCG}_3 \approx 0.63$.
User 1: $\text{NDCG}_3 = 1$
User 2: $\text{DCG}_3 = \frac{1}{\log_2 3} + \frac{1}{\log_2 4} \approx 1.13$,
$\text{IDCG}_3 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} \approx 2.13$, $\text{NDCG}_3 \approx 0.53$.

Overall $\text{NDCG}_3 \approx \frac{0.63+1+0.53}{3} = 0.72$.

MAP

MAP at k

MAP (mean average precision) is a mean of AP over all users.

$$\mathrm{AP}_k = \frac{1}{\min(k, r)} \sum_{i=1}^{k} rel(i) \cdot \mathrm{prec@}\, i$$

where rel_i is a relevance of the i-th recommended item (0 or 1) and r is a number of given user's ratings in the test set.

Toy example

User 0: 0, 1, 0 - 1 rating in the test set
User 1: 1, 0, 0 - 1 rating in the test set
User 2: 0, 1, 1 - 3 ratings in the test set

User 0: $\mathrm{AP}_3 = \frac{1}{2}$
User 1: $\mathrm{AP}_3 = 1$
User 2: $\mathrm{AP}_3 =$

MAP

MAP at k

MAP (mean average precision) is a mean of AP over all users.

$$\mathrm{AP}_k = \frac{1}{\min(k, r)} \sum_{i=1}^{k} rel(i) \cdot \mathrm{prec@}\, i$$

where rel_i is a relevance of the i-th recommended item (0 or 1) and r is a number of given user's ratings in the test set.

Toy example

User 0: 0, 1, 0 - 1 rating in the test set
User 1: 1, 0, 0 - 1 rating in the test set
User 2: 0, 1, 1 - 3 ratings in the test set

User 0: $\mathrm{AP}_3 = \frac{1}{2}$
User 1: $\mathrm{AP}_3 = 1$
User 2: $\mathrm{AP}_3 = \frac{\frac{1}{2} + \frac{2}{3}}{3} \approx 0.39.$

$\mathrm{MAP}_3 \approx 0.63.$

MRR

MRR at k

MRR (mean reciprocal rank) is a mean of RR over all users :)
Reciprocal rank is the inverse of the position of the first relevant item.

Toy example

User 0: 0, 1, 0 - 1 rating in the test set
User 1: 1, 0, 0 - 1 rating in the test set
User 2: 0, 1, 1 - 3 ratings in the test set

User 0: $\text{MRR}_3 = \frac{1}{2}$
User 1: $\text{MRR}_3 = 1$
User 2: $\text{MRR}_3 =$

MRR

MRR at k

MRR (mean reciprocal rank) is a mean of RR over all users :)
Reciprocal rank is the inverse of the position of the first relevant item.

Toy example

User 0: 0, 1, 0 - 1 rating in the test set
User 1: 1, 0, 0 - 1 rating in the test set
User 2: 0, 1, 1 - 3 ratings in the test set

User 0: $\text{MRR}_3 = \frac{1}{2}$
User 1: $\text{MRR}_3 = 1$
User 2: $\text{MRR}_3 = \frac{1}{2}$

$\text{MRR}_3 \approx 0.67$.

Confusion matrix

	Relevant	Irrelevant	Total
Recommended	tp	fp	$tp + fp$
Not Recommended	fn	tn	$fn + tn$
Total	$tp + fn$	$fp + tn$	N

Figure: Confusion matrix. Source: [1]

$$\text{recall} = tpr = \frac{tp}{tp+fn}$$
$$\text{fallout} = fpr = \frac{fp}{fp+tn}$$
$$\text{precision} = \frac{tp}{tp+fp}$$

Receiver operating characteristic curve

ROC curve is a plot of recall (y-axis) against fallout (x-axis) for various number of recommendations.

Area under the curve

AUC is the area under ROC curve.

Examples AUC

Example	Acc.	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Precision@4	Recall@4	F1@4	MatthewsCorr@4	AUC	LAUC 4	MAP
(a)	↑	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	0,750	1,000	0,857	0,802	1,000	1,000	1,000
(b)	→	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	0,750	1,000	0,857	0,802	0,952	0,952	0,917
(c)	→	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	0,500	0,667	0,571	0,356	0,905	0,786	0,867
(d)	→	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	0,500	0,667	0,571	0,356	0,857	0,738	0,756
(e)	→	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	0,500	0,667	0,571	0,356	0,619	0,738	0,656
(f)	→	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗	0,500	0,667	0,571	0,356	0,810	0,690	0,700
(g)	→	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓	0,500	0,667	0,571	0,356	0,571	0,690	0,600
(h)	→	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	0,250	0,333	0,286	-0,089	0,714	0,524	0,633
(i)	→	✓	✗	✗	✗	✓	✗	✗	✗	✗	✓	0,250	0,333	0,286	-0,089	0,524	0,524	0,567
(j)	→	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	0,250	0,333	0,286	-0,089	0,333	0,524	0,507
(k)	→	✗	✓	✗	✗	✓	✓	✗	✗	✗	✗	0,250	0,333	0,286	-0,089	0,667	0,476	0,467
(l)	→	✗	✗	✓	✗	✓	✓	✗	✗	✗	✗	0,250	0,333	0,286	-0,089	0,619	0,429	0,411
(m)	→	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	0,250	0,333	0,286	-0,089	0,571	0,381	0,383
(n)	→	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	0,000	0,000	0,000	-0,535	0,429	0,214	0,321
(o)	↓	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	0,000	0,000	0,000	-0,535	0,000	0,214	0,216

Figure: Example from [1]

Example c)

Recommending **1 item**:

$tpr = \frac{1}{3}$, $fpr = \frac{0}{7}$

	Relevant	Irrelevant
Recommended	1	0
Not recommended	2	7

Recommending **2 items**:

$tpr = \frac{2}{3}$, $fpr = \frac{0}{7}$

	Relevant	Irrelevant
Recommended	2	0
Not recommended	1	7

Recommending **3 items**:

$tpr = \frac{2}{3}$, $fpr = \frac{1}{7}$

	Relevant	Irrelevant
Recommended	2	1
Not recommended	1	6

Example c)

Recommending **4 items**:

$tpr = \frac{2}{3}$, $fpr = \frac{2}{7}$

	Relevant	Irrelevant
Recommended	2	2
Not recommended	1	5

Recommending **5 items**:

$tpr = \frac{3}{3}$, $fpr = \frac{2}{7}$

	Relevant	Irrelevant
Recommended	3	2
Not recommended	0	5

Recommending **6 items**:

$tpr = \frac{3}{3}$, $fpr = \frac{3}{7}$

	Relevant	Irrelevant
Recommended	3	3
Not recommended	0	4

...

Example c)

Recommending **4 items**:

$tpr = \frac{2}{3}$, $fpr = \frac{2}{7}$

	Relevant	Irrelevant
Recommended	2	2
Not recommended	1	5

Recommending **5 items**:

$tpr = \frac{3}{3}$, $fpr = \frac{2}{7}$

	Relevant	Irrelevant
Recommended	3	2
Not recommended	0	5

Recommending **6 items**:

$tpr = \frac{3}{3}$, $fpr = \frac{3}{7}$

	Relevant	Irrelevant
Recommended	3	3
Not recommended	0	4

...

Make a plot and figure out that it is not needed to compute confusion matrices.

LAUC

In the most of cases we do not store all ranked items for each user, but only top N recommendations. How to compute AUC then?

LAUC

In the most of cases we do not store all ranked items for each user, but only top N recommendations. How to compute AUC then?

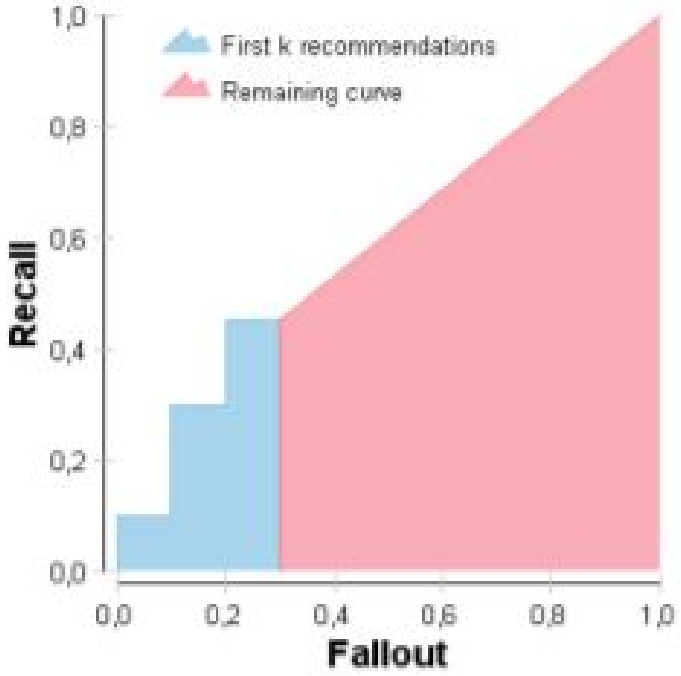

Figure: Limited ROC curve introduced in [1]

LAUC - example c)

In our case for 4 recommendations we had $tpr = \frac{2}{3}$, $fpr = \frac{2}{7}$.

So our area is the sum of 2 parts:
1) Area under the curve from $(0, 0)$ to $(\frac{2}{7}, \frac{2}{3})$.
Note that for computation it is enough to sum up tprs (recalls) at k for all k, where item k is not relevant.

LAUC - example c)

In our case for 4 recommendations we had $tpr = \frac{2}{3}$, $fpr = \frac{2}{7}$.

So our area is the sum of 2 parts:
1) Area under the curve from $(0, 0)$ to $(\frac{2}{7}, \frac{2}{3})$.
Note that for computation it is enough to sum up tprs (recalls) at k for all k, where item k is not relevant.

2) A trapezium which is the area under the line connecting $(\frac{2}{7}, \frac{2}{3})$ with $(1, 1)$.
Note that:
- the height of trapezium is a fraction of not recommended irrelevant items,
- length of the smaller side is a fraction of recommended relevant items,
- length of the second side is 1.

LAUC - example c)

In our case for 4 recommendations we had $tpr = \frac{2}{3}$, $fpr = \frac{2}{7}$.

So our area is the sum of 2 parts:
1) Area under the curve from $(0, 0)$ to $(\frac{2}{7}, \frac{2}{3})$.
Note that for computation it is enough to sum up tprs (recalls) at k for all k, where item k is not relevant.

2) A trapezium which is the area under the line connecting $(\frac{2}{7}, \frac{2}{3})$ with $(1, 1)$.
Note that:
- the height of trapezium is a fraction of not recommended irrelevant items,
- length of the smaller side is a fraction of recommended relevant items,
- length of the second side is 1.

Question: is LAUC at 4 the same for 1,0,0,1,... and 0,1,1,0,...?

LAUC - example c)

In our case for 4 recommendations we had $tpr = \frac{2}{3}$, $fpr = \frac{2}{7}$.

So our area is the sum of 2 parts:
1) Area under the curve from $(0, 0)$ to $(\frac{2}{7}, \frac{2}{3})$.
Note that for computation it is enough to sum up tprs (recalls) at k for all k, where item k is not relevant.

2) A trapezium which is the area under the line connecting $(\frac{2}{7}, \frac{2}{3})$ with $(1, 1)$.
Note that:
- the height of trapezium is a fraction of not recommended irrelevant items,
- length of the smaller side is a fraction of recommended relevant items,
- length of the second side is 1.

Question: is LAUC at 4 the same for 1,0,0,1,... and 0,1,1,0,...? YES!

Hit rate

HR at k

Hit rate at k ($HR@k$) equals one if **at least one** of the recommended items is relevant.
Hit rate of a recommender system is an average of hit rates over all users.

Ambiguity of $HR@k$

There are different definitions of ($HR@k$). Please check carefully before comparing results from different sources.

Toy example

User 0: 0, 1, 0 - 1 rating in the test set
User 1: 1, 0, 0 - 1 rating in the test set
User 2: 0, 1, 1 - 3 ratings the in test set

User 0: $\mathrm{HR}_3 = 1$
User 1: $\mathrm{HR}_3 = 1$
User 2: $\mathrm{HR}_3 =$

Hit rate

HR at k

Hit rate at k ($HR@k$) equals one if **at least one** of the recommended items is relevant.
Hit rate of a recommender system is an average of hit rates over all users.

Ambiguity of $HR@k$

There are different definitions of ($HR@k$). Please check carefully before comparing results from different sources.

Toy example

User 0: 0, 1, 0 - 1 rating in the test set
User 1: 1, 0, 0 - 1 rating in the test set
User 2: 0, 1, 1 - 3 ratings the in test set

User 0: $\mathrm{HR}_3 = 1$
User 1: $\mathrm{HR}_3 = 1$
User 2: $\mathrm{HR}_3 = 1$

$\mathrm{HR}_3 = 1.$

Coverage

Coverage (catalog)

Coverage is a percentage of items which we recommended to at least one user. In our case we restrict to items from the test set.

```
Training data:

matrix([[3, 4, 0, 0, 5, 0, 0, 4],
        [0, 1, 2, 3, 0, 0, 0, 0],
        [0, 0, 0, 5, 0, 3, 4, 0]], dtype=int64)

Test data:

matrix([[0, 0, 0, 0, 0, 0, 3, 0],
        [0, 0, 0, 0, 5, 0, 0, 0],
        [5, 0, 4, 0, 0, 0, 0, 2]], dtype=int64)

Recommendations:
```

	0	1	2	3	4	5	6
0	0	30	4.375000	60	4.375000	50	3.375000
1	10	40	4.166667	60	3.166667	70	3.166667
2	20	40	5.333333	70	4.333333	0	3.333333

Figure: Source: Notebook P2. Evaluation - toy example

In the test set we have 5 different items. Only item 20 were not recommended to any user, so coverage=0.8.

Coverage

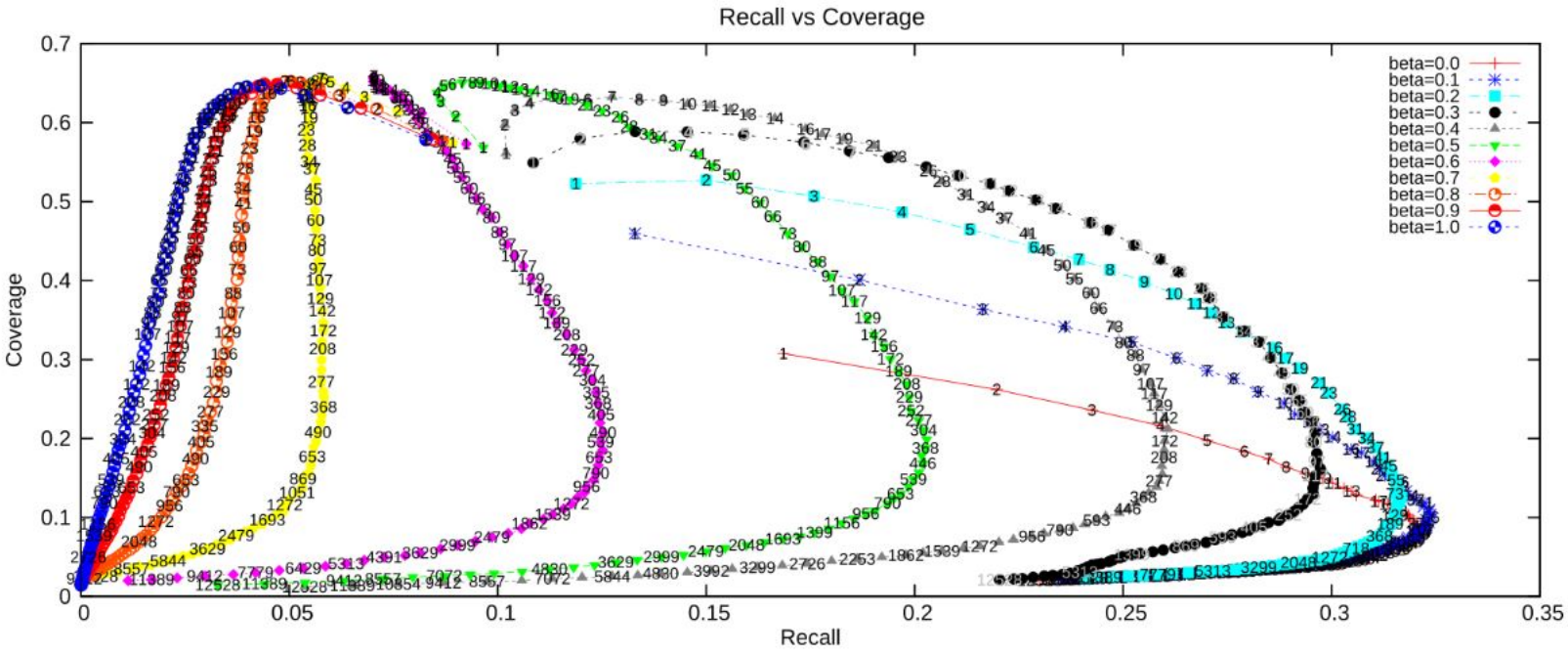

Figure: Compromise between recall and coverage, source: medium.com [2]

Novelty

Novelty

Novelty is a percentage of items which user has not seen before. Usually (especially for implicit feedback datasets) we assume that user seen before only items in the train set.

Novelty

Novelty

Novelty is a percentage of items which user has not seen before. Usually (especially for implicit feedback datasets) we assume that user seen before only items in the train set.

Novelty as an online metric

We can measure novelty based on user studies which is more accurate, but expensive.

Novelty

Novelty

Novelty is a percentage of items which user has not seen before. Usually (especially for implicit feedback datasets) we assume that user seen before only items in the train set.

Novelty as an online metric

We can measure novelty based on user studies which is more accurate, but expensive.

Use case

To measure novelty we need a dataset where the same pair (user, item) can appear multiple times.
For example when we consider clickstream data from 14 days and treat last day as a test set.

In our case we recommend only items which user has not seen (rated) before so novelty always equals 1.

Gini index

Gini index is a measure of dispersion widely used in economy for monitoring wealth distribution.

Gini index in recommender systems [3]

$$G = \frac{1}{n-1} \sum_{j=1}^{n} (2j - n - 1) p(i_j)$$

n is the number of all considered items,
$p(i)$ is a fraction of times item i was recommended,
$i_1, i_2, \ldots, i_n$ is a list of items ordered according to increasing $p(i_j)$

Gini index

Gini index is a measure of dispersion widely used in economy for monitoring wealth distribution.

Gini index in recommender systems [3]

$$G = \frac{1}{n-1} \sum_{j=1}^{n} (2j - n - 1) p(i_j)$$

n is the number of all considered items,
$p(i)$ is a fraction of times item i was recommended,
$i_1, i_2, \ldots, i_n$ is a list of items ordered according to increasing $p(i_j)$

What is a set of possible values of Gini index?
When does Gini index achieve minimal and maximal value?

Gini index - example

$$G = \frac{1}{n-1} \sum_{j=1}^{n} (2j - n - 1) p(i_j)$$

	0	1	2	3	4	5	6
0	0	30	4.375000	60	4.375000	50	3.375000
1	10	40	4.166667	60	3.166667	70	3.166667
2	20	40	5.333333	70	4.333333	0	3.333333

Figure: Source: Notebook P2. Evaluation - toy example

We have items recommended, item 0: 1 time, item 10: 0 times, item 20: 0 times, item 30: 1 time, item 40: 2 times, item 50: 1 time, item 60: 2 times, item 70: 2 times.
So: $n = 8$, $p(i_1) = p(i_2) = 0$, $p(i_3) = p(i_4) = p(i_5) = \frac{1}{9}$, $p(i_6) = p(i_7) = p(i_8) = \frac{2}{9}$.

$$G = \frac{1}{7}((-7 \cdot \frac{0}{9}) + (-5 \cdot \frac{0}{9}) + (-3 \cdot \frac{1}{9}) + \cdots + (5 \cdot \frac{2}{9}) + (7 \cdot \frac{2}{9})) = \frac{3}{7}$$

.

Gini index - based on the test set only

$$G = \frac{1}{n-1}\sum_{j=1}^{n}(2j - n - 1)p(i_j)$$

	0	1	2	3	4	5	6
0	0	30	4.375000	60	4.375000	50	3.375000
1	10	40	4.166667	60	3.166667	70	3.166667
2	20	40	5.333333	70	4.333333	0	3.333333

Figure: Source: Notebook P2. Evaluation - toy example

In Notebook P2. Evaluation we restrict our evaluation to items in the test set so items 10, 30 and 50 should be excluded.
Then we have items recommended, item 0: 1 time, item 20: 0 times, item 40: 2 times, item 60: 2 times, item 70: 2 times.
So: $n = 5$, $p(i_1) = 0$, $p(i_2) = \frac{1}{7}$, $p(i_3) = p(i_4) = p(i_5) = \frac{2}{7}$.

$$G = \frac{1}{4}((-4 \cdot \frac{0}{7}) + (-2 \cdot \frac{1}{7}) + (0 \cdot \frac{2}{7}) + (2 \cdot \frac{2}{7}) + (4 \cdot \frac{2}{7})) = \frac{10}{28} \approx 0.36$$

.

Shannon entropy

Shannon entropy (or Shannon index) is another measure of dispersion. For example in biology it is used to measure diversity of species.

Shannon entropy in recommender systems [3]

$$H = -\sum_{i=1}^{n} p(i) \cdot \log(p(i))$$

n is the number of all considered items
$p(i)$ is a fraction of times item i was recommended
We will use natural logarithm, but other bases are possible.

Shannon entropy

Shannon entropy (or Shannon index) is another measure of dispersion. For example in biology it is used to measure diversity of species.

Shannon entropy in recommender systems [3]

$$H = -\sum_{i=1}^{n} p(i) \cdot \log(p(i))$$

n is the number of all considered items
$p(i)$ is a fraction of times item i was recommended
We will use natural logarithm, but other bases are possible.

Problem with $p(i) = 0$

We can either skip items when $p(i) = 0$ or assume that $0 \cdot \log(0) = 0$ (which has some mathematical sense).

Shannon entropy

Shannon entropy (or Shannon index) is another measure of dispersion.
For example in biology it is used to measure diversity of species.

Shannon entropy in recommender systems [3]

$$H = -\sum_{i=1}^{n} p(i) \cdot \log(p(i))$$

n is the number of all considered items
$p(i)$ is a fraction of times item i was recommended
We will use natural logarithm, but other bases are possible.

Problem with $p(i) = 0$

We can either skip items when $p(i) = 0$ or assume that $0 \cdot \log(0) = 0$ (which has some mathematical sense).

What is a set of possible values of Shannon index?
When does Shannon index achieve minimal and maximal value?

Shannon entropy

$$H = -\sum_{i=1}^{n} p(i) \cdot \log(p(i))$$

	0	1	2	3	4	5	6
0	0	30	4.375000	60	4.375000	50	3.375000
1	10	40	4.166667	60	3.166667	70	3.166667
2	20	40	5.333333	70	4.333333	0	3.333333

Figure: Source: Notebook P2. Evaluation - toy example

In Notebook P2. Evaluation we restrict our evaluation to items in the test set so items 10, 30 and 50 should be excluded.
Then we have items recommended, item 0: 1 time, item 20: 0 times, item 40: 2 times, item 60: 2 times, item 70: 2 times.
So: $n = 5$, $p(i_1) = 0$, $p(i_2) = \frac{1}{7}$, $p(i_3) = p(i_4) = p(i_5) = \frac{2}{7}$.

$$H = -(\frac{1}{7}\ln(\frac{1}{7}) + \frac{2}{7}\ln(\frac{2}{7}) + \frac{2}{7}\ln(\frac{2}{7}) + \frac{2}{7}\ln(\frac{2}{7})) \approx 1.35$$

.

Online evaluation

Offline evaluation is convenient for development purposes, but

- the offline improvement not necessarily implies online improvement on similar measures
- offline evaluation metrics usually can not cover business logic
- the impact of recommendations affect many different aspects of user's behaviour

Finding offline evaluation metric the most correlated with online metric is very important. In [4] authors from Indeed suggested using precision@k in case of their job recommendations task.

A/B testing

A/B test - what is it?

A/B testing is the methodology using to compare variants A and B of a given functionality by randomly dividing samples into 2 groups and targeting them with different variants.

Example

We can check whether our new recommender system is better by sending recommendations generated by old recommender system to randomly selected 10k users (A group) and recommendations generated by new recommender system to other randomly selected 10k (B group).
Then we can compare how many clicks each group generated.

Python implementation

To do (especially for absent students):

- Go through - *P2. Evaluation* notebook to:
  - prepare data for evaluation
  - understand implementation of the metrics described
  - check sample recommendations - are they good?
  - **project task 2: implement some other evaluation measure** - it may be your idea, modification of what we have already implemented (for example Hit2 rate which would count as a success users who received at least 2 relevant recommendations) or something well-known

References I

[1] G. Schröder, M. Thiele, and W. Lehner, "Setting goals and choosing metrics for recommender system evaluations", vol. 811, Jan. 2011.

[2] T. Řehořek, "Evaluating recommender systems: Choosing the best one for your business", https://medium.com/recombee-blog/evaluating-recommender-systems-choosing-the-best-one-for-your-business-c688ab781a35.

[3] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in. Jan. 2011, vol. 12, pp. 257–297. DOI: 10.1007/978-0-387-85820-3_8.

[4] A. Mogenet, T. Pham, M. Kazama, and J. Kong, "Predicting online performance of job recommender systems with offline evaluation", Sep. 2019, pp. 477–480, ISBN: 978-1-4503-6243-6. DOI: 10.1145/3298689.3347032.