

Recommender Systems

Marina Ananyeva, M.S.

ananyeva.me@gmail.com, telegram: @ananyevame

Academic year 2021/2022.

Course outline & sources

Useful links & grading

1. Wiki with the course syllabus
http://wiki.cs.hse.ru/RecSys_2021_2022
2. Github repo
https://github.com/anamarina/RecSys_course
3. Grades table
4. List of recommended articles
5. Drive.google for your abstracts and presentations of selected articles

Grade (RecSys part) = $0.3 * (\text{Home assignments}) + 0.15 * (\text{article abstract/presentations}) +$
+ $0.15 * (\text{practical class activity}) + 0.4 * (\text{oral exam (RecSys case-cracking)})$

Grade for Research seminar (semester 1) = $0.5 * \text{RecSys part} + 0.5 * \text{RL part}$

Overview

- **Week 1 [lecture].**

Introduction. Ranking task, functions and metrics. Overview of RecSys approaches and research directions. Overview of top problems in RecSys.

- **Week 2 [lecture+practice+2 abstracts]**

General approaches [MF, FM, CF (ALS), user/item knn-based similarity]. Microsoft recommenders, implicit libraries.

- **Week 3 [lecture+practice+2 abstracts]**

Content-based & hybrid models. LightFM model and library. Two-level models. Gradient boosting for ranking task. Fails in preprocessing and feature engineering.

- **Week 4 [lecture+practice+2 abstracts]**

Context-based and sequential models. RecBole library.

Overview

- **Week 5 [lecture+practice+2 abstracts]**

Interpretability and explainability in RecSys. Simultaneous learning of interpretations and recommendations. Explanation graph-based algorithms. Attention-based models.

- **Week 6 [lecture+practice+2 abstracts]**

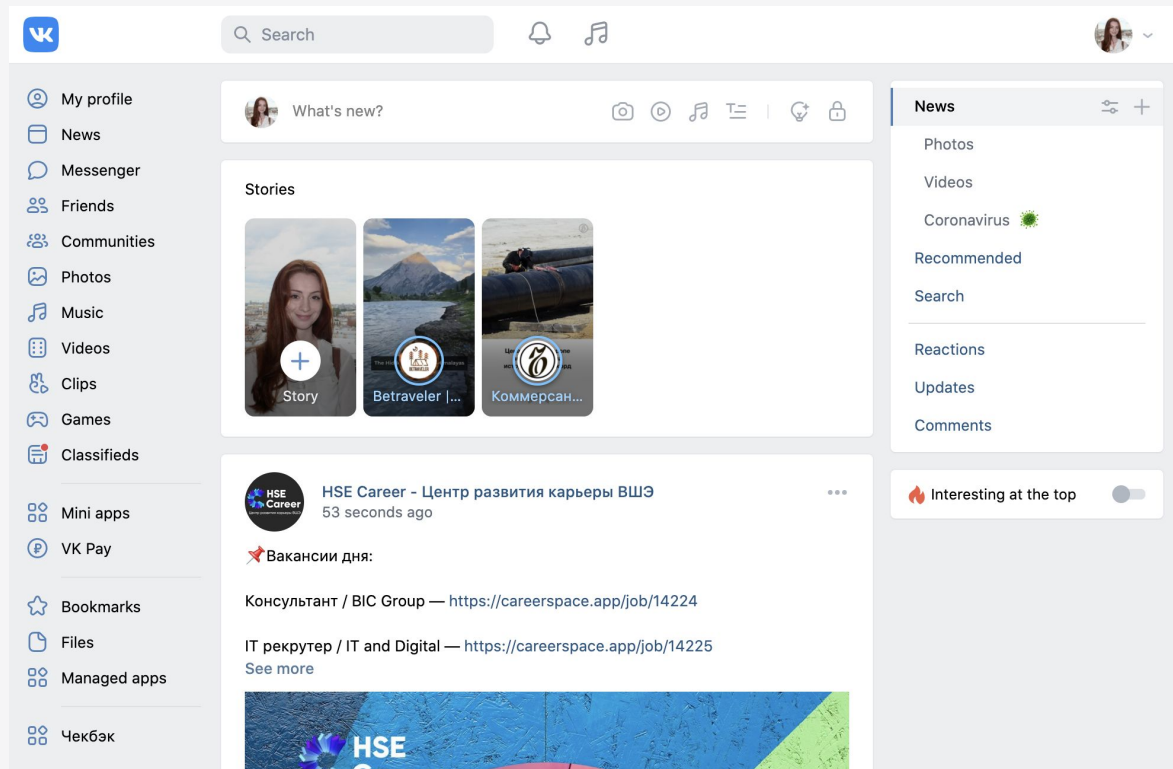
Offline and online metrics. A/B tests: design, testing, monitoring, results. Proxy metrics, datasets, targets. Bias problem and debiasing techniques. Bandits. RL in RecSys.

- **Week 7 [lecture + 2 abstracts]**

ML design in RecSys. Approaches and architectures for production systems. Large scale environment. API services. DB (PostgresQL, Redis, GraphQL). Batch training and inference vs. real-time systems.

Week 1. Intro

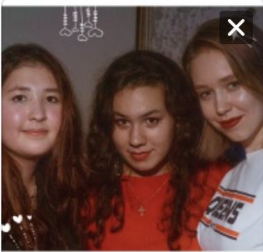
Social media. Personal Feed



Social media. Suggestions


People you may know

Show all >



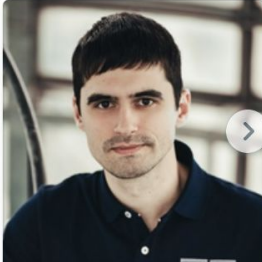
3 mutual friends

+ Add



24 years old
2 mutual friends

+ Add




ВКонтакте
4 mutual friends

+ Add


Recommended communities

Hide



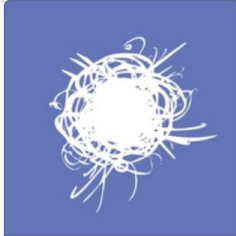
Машинное обучен...
Programming

Follow



QWERTY - лучший...
Education

Follow



Хабр Карьера
Classifieds

Follow

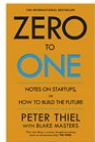
Show all communities >

E-commerce. Amazon



Customers who bought this item also bought

Page 1 of 13



Zero to One: Notes on Start Ups, or How to Build the Future
› Blake Masters
★★★★★ 265
Paperback
£6.49 ✓prime



The Startup Way: How Entrepreneurial Management...
› Eric Ries
★★★★★ 8
Paperback
38 offers from £5.00



The Hard Thing About Hard Things: Building a Business When There...
› Ben Horowitz
★★★★★ 113
Hardcover
£13.60 ✓prime



ReWork: Change the Way You Work Forever
› David Heinemeier...
★★★★★ 243
Paperback
£9.35 ✓prime



The Startup Way: How Entrepreneurial Management...
› Eric Ries
★★★★★ 8
Hardcover
£15.59 ✓prime



The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail...
› Clayton M. Christensen
★★★★★ 82
Paperback
£17.28 ✓prime



Sprint: How To Solve Big Problems and Test New Ideas in Just Five Days
› Jake Knapp
★★★★★ 54
Paperback
£10.19 ✓prime



The Four Steps to the Epiphany: Successful Strategies for Products That Sell...
› Steve Blank
★★★★★ 18
Hardcover
£27.29 ✓prime

Netflix



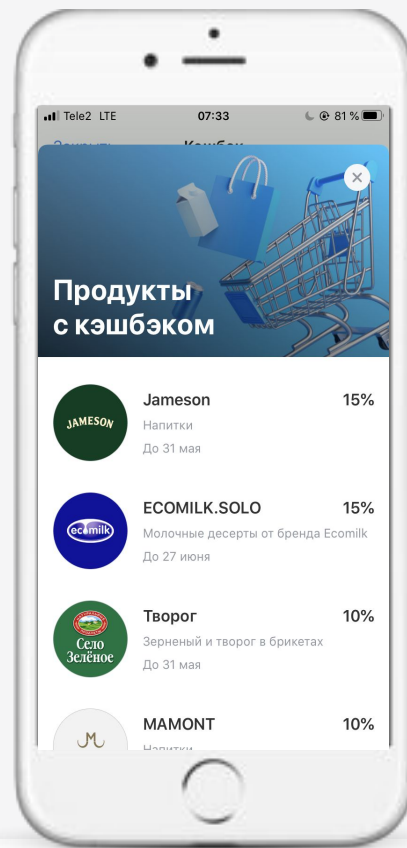
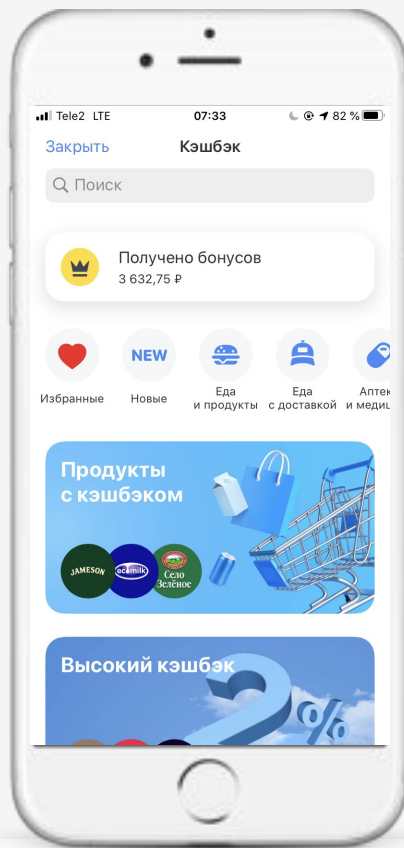
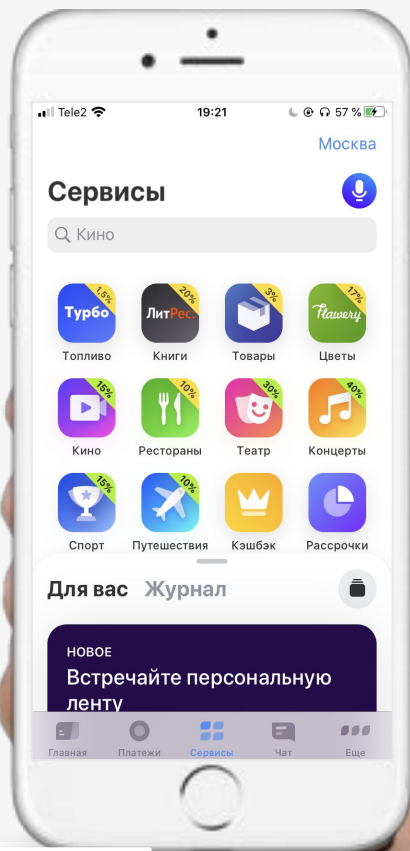
The BigChaos Solution to the Netflix Grand Prize

https://www.netflixprize.com/assets/GrandPrize2009_BPC_BigChaos.pdf

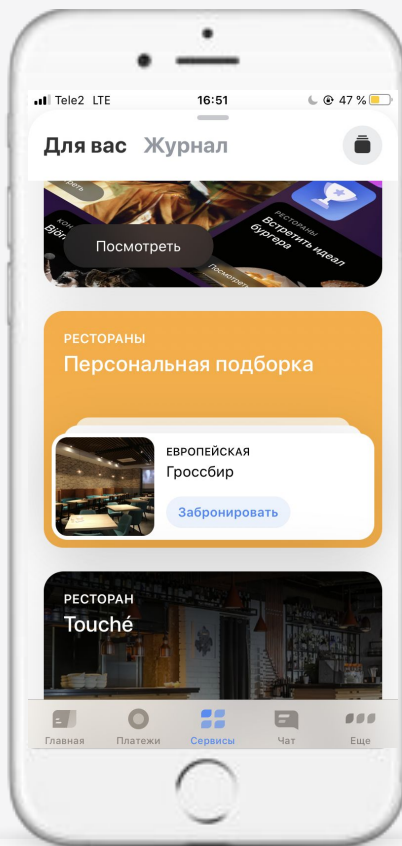
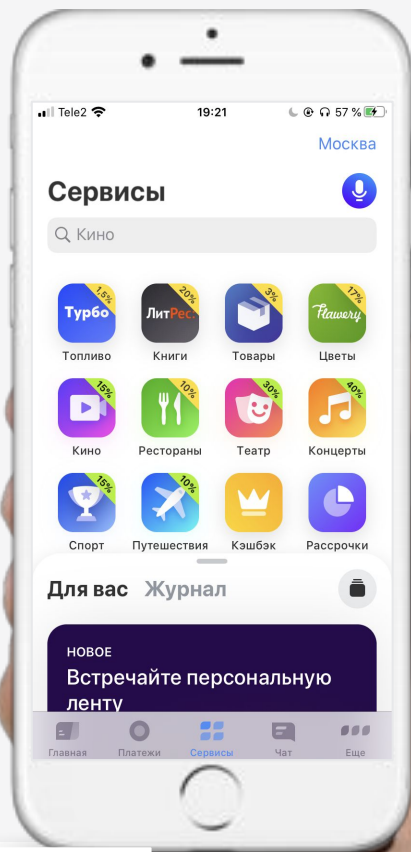
Netflix. Beyond what you thought



Apps-in-App



Apps-in-App



Формализация задачи

Постановка задачи ранжирования

X — множество объектов.

$X^l = \{x_1, \dots, x_l\}$ — обучающая выборка

$i < j$ на парах $(i, j) \in \{1, \dots, l\}^2$

Задача:

Построить ранжирующую функцию $a : X \rightarrow \mathbb{R}$ такую, что: $i < j \Rightarrow a(x_i) < a(x_j)$.

Линейная модель ранжирования:

$a(x; w) = \langle x, w \rangle$, где $x \mapsto (f_1(x), \dots, f_n(x)) \in \mathbb{R}^n$ — вектор признаков объекта x .

Пример 1. Ранжирование поисковой выдачи

D — коллекция текстовых документов.

Q — множество запросов.

$D_q \subseteq D$ — множество документов, найденных по запросу q .

$X = Q \times D$ — объектами являются пары (запрос, документ): $x \equiv (q, d), q \in Q, d \in D_q$.

Y — упорядоченное множество рейтингов.

$y : X \rightarrow Y$ — оценки релевантности, поставленные ассессорами (экспертами): чем выше оценка $y(q, d)$, тем релевантнее документ d по запросу q .

Правильный порядок определен только между документами, найденными по одному и тому же запросу q : $(q, d) < (q, d') \Leftrightarrow y(q, d) < y(q, d')$.

Релевантные ответы запросу q — это список документов d , упорядоченных с помощью функции ранжирования $a(q, d)$.

Пример 2. Рекомендации пользователям

U — пользователи.

I — предметы (фильмы, книги и т.д.).

$X = U \times I$ — объектами являются пары (пользователь, предмет).

$$(u, i) < (u, i') \Leftrightarrow y(u, i) < y(u, i')$$

Рекомендация пользователю u — это список предметов i , упорядоченный с помощью функции ранжирования $a(u, i)$.

Данные по типу feedback

user_id \ item_id	item #1	item #n
user #1	0	1	...
....	1	0	...
user #m	1

implicit

user_id \ item_id	item #1	item #n
user #1	0	5	...
....	2	0	...
user #m	3

explicit

Функции ранжирования

Функции ранжирования

1. **Pointwise** (точечные)

2. **Pairwise** (попарные)

3. **Listwise** (списочные)

Функции ранжирования

- **BPR** (Bayesian Personalised Ranking, pairwise)
- **WARP** (Weighted Approximate-Rank Pairwise)
- **RankNET** (pairwise)
- **LambdaRANK** (pairwise/listwise)
- **LambdaMART** (pairwise/listwise)

RankNET

Обучим функцию, которая по $\mathbf{x} \in \mathbb{R}^n$ выдает $f(\mathbf{x})$ и ранжирует документы по значению $f(\mathbf{x})$.

Для тестовых примеров \mathbf{x}_i и \mathbf{x}_j модель считает $s_i = f(\mathbf{x}_i)$ и $s_j = f(\mathbf{x}_j)$, а затем оценивает:

$$p_{ij} = p(\mathbf{x}_i \succ \mathbf{x}_j) = \frac{1}{1 + e^{-\alpha(s_i - s_j)}}$$

$q(\mathbf{x}_i \succ \mathbf{x}_j)$ - данные. Поэтому разумная использовать функцию ошибки:

$$C = -q_{ij} \log p_{ij} - (1 - q_{ij}) \log(1 - p_{ij}).$$

LambdaRank

Действительно, можно домножить градиент исходного функционала на то, насколько изменится nDCG, если поменять местами x_i и x_j :

$$\omega := \omega + \eta \frac{1}{1 + \exp(\langle \omega, x_j - x_i \rangle)} (x_j - x_i) |\Delta nDCG_{ij}| (x_j - x_i)$$



23



BPR: Bayesian Personalized Ranking

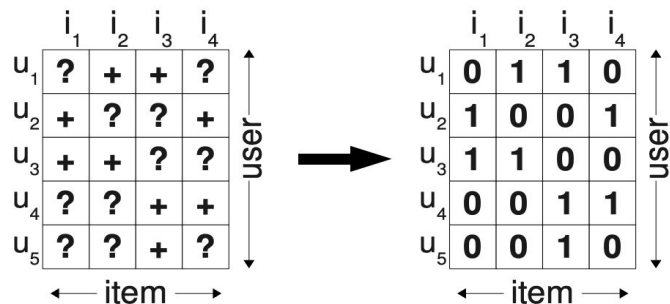


Рис. 1: Бинаризация implicit данных, где отсутствие интеракции кодируется нулем.

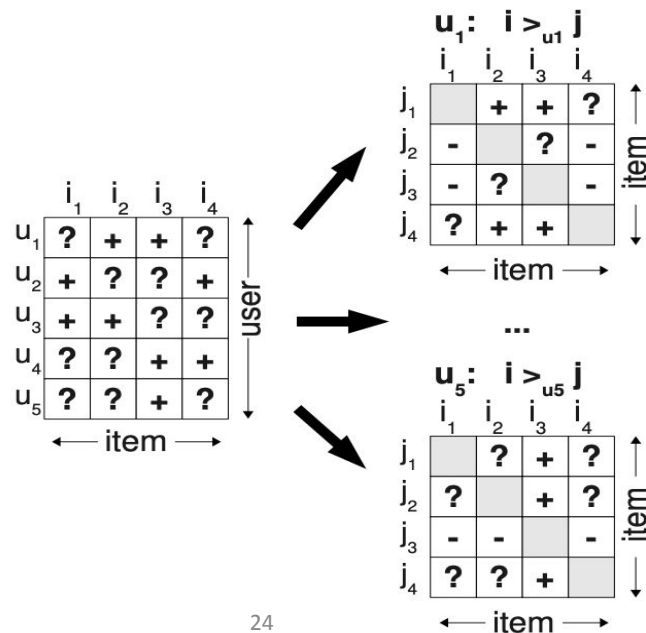


Рис. 2: Парные предпочтения пользователю по объектам для BPR. + означает, что пользователь предпочитает i объект j -му, знак минуса - наоборот.

BPR: Bayesian Personalized Ranking

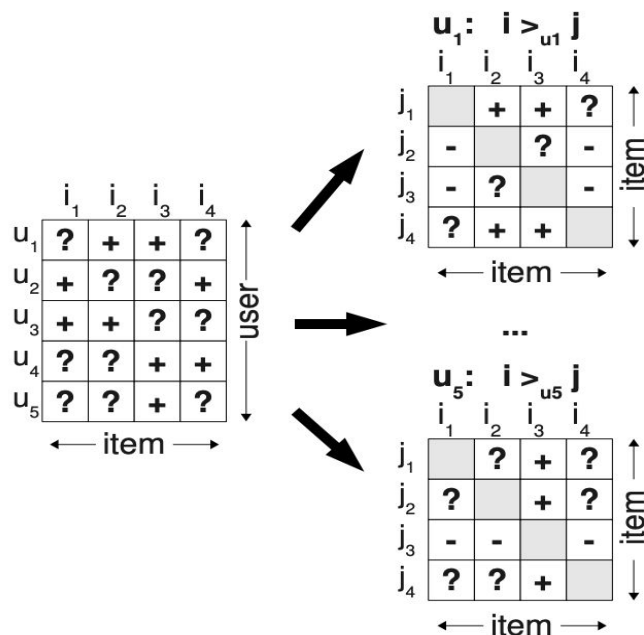
Пользователь предпочитает объект i по отношению к j :

$$(u, i, j) \in D_S$$

$$p(i <_u j | \Theta) = \sigma(\hat{r}_{uij}(\Theta)),$$

$$\min_{\Theta} \sum_{(u,i,j) \in \mathcal{D}_S} \ln \sigma(\hat{r}_{uij}) - \lambda \|\Theta\|^2$$

$$\Theta \leftarrow \Theta + \alpha \left(\frac{e^{-\hat{r}_{uij}}}{1 + e^{-\hat{r}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{r}_{uij} + \lambda \Theta \right)$$



Weighted Approximate-Rank Pairwise

- Смотрим на тройки: (**user**, **positive_item**, **negative_item**)
- Выбираем негативные примеры не случайно, а так, чтобы эти примеры нарушали текущее ранжирование модели, то есть, были выше, чем положительный пример:

Algorithm 1 Online WARP Loss Optimization

Input: labeled data (x_i, y_i) , $y_i \in \{1, \dots, Y\}$.

repeat

 Pick a random labeled example (x_i, y_i)

 Let $f_{y_i}(x_i) = \Phi_W(y_i)^\top \Phi_I(x_i)$

 Set $N = 0$.

repeat

 Pick a random annotation $\bar{y} \in \{1, \dots, Y\} \setminus y_i$.

 Let $f_{\bar{y}}(x_i) = \Phi_W(\bar{y})^\top \Phi_I(x_i)$

$N = N + 1$.

until $f_{\bar{y}}(x_i) > f_{y_i}(x_i) - 1$ or $N \geq Y - 1$

if $f_{\bar{y}}(x_i) > f_{y_i}(x_i) - 1$ **then**

 Make a gradient step to minimize:

$$L(\lfloor \frac{Y-1}{N} \rfloor) |1 - f_{y_i}(x_i) + f_{\bar{y}}(x_i)|_+$$

 Project weights to enforce constraints (2)-(3).

end if

until validation error does not improve.

Метрики оценки качества ранжирования

Зоопарк метрик качества ранжирования

- **Hitrate (Hit ratio)**
 - **Precision@k**
 - **Recall@k**
 - **AP@k, MAP@k, MNAP@k**
 - **DCG@k, NDCG@k**
 - **MRR**
- и другие :)

Измерение качества рекомендаций

Diversity (разнообразие) - число рекомендаций из разных категорий, степень различия рекомендаций между сессиями пользователя, различие рекомендаций между пользователями.

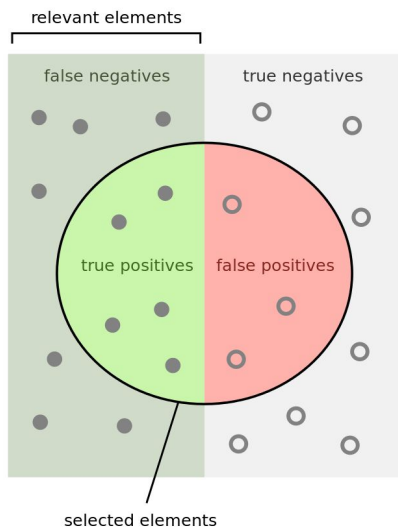
Novelty (новизна) - сколько среди рекомендаций новых для пользователей объектов.

Coverage (покрытие) - доля объектов, которые хотя бы раз побывали среди топа рекомендаций

Serendipity (догадливость) - способность угадывать неожиданные нетривиальные предпочтения пользователя.

- Можно выбрать один из них!
- Зачем выбирать, если можно оптимизировать линейную комбинацию из них!

Hitrate, Precision@k, Recall@k



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$\text{Hit Rate} = \frac{\# \text{ hits}}{\# \text{ hits} + \# \text{ misses}}$$

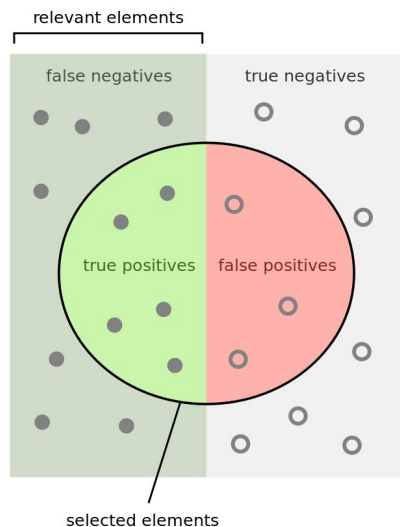
Precision – сколько из рекомендованных объектов оказались релевантными?

Recall – сколько из релевантных для пользователя объектов мы порекомендовали?

Эти метрики зависят от количества рекомендаций для каждого пользователя?

Precision@k, Recall@k – фиксируем для каждого пользователя топ k рекомендованных объектов, считаем метрику по бинарному таргету (0 – нерелевантная рекомендация, 1 – релевантная).

Precision@k и Recall@k в рекомендациях



Precision at K ($p@K$) — точность на топе K элементов — базовая метрика качества ранжирования для одного пользователя. **Recall@k** - по аналогии, в знаменателе - общее число релевантных объектов.

$$p@K = \frac{\sum_{k=1}^K r^{true}(\pi^{-1}(k))}{K} = \frac{k \text{ релевантных рекомендованных объектов}}{K \text{ рекомендованных объектов}}$$

$\pi^{-1}(k)$ - рекомендованный объект на k-й позиции в результате перестановки
 r^{true} - настоящий рейтинг (релевантности) объекта

How many selected items are relevant?

Precision = $\frac{\text{green}}{\text{green} + \text{red}}$

How many relevant items are selected?

Recall = $\frac{\text{green}}{\text{green}}$

Average precision @ k (AP@k)

- Учитывает порядок рекомендованных объектов в топе k.

$$ap@K = \frac{1}{K} \sum_{k=1}^K r^{true}(\pi^{-1}(k)) \cdot p@k$$

 = relevant documents for query 1

Ranking #1 

Mean average precision (MAP@k)











Считаем AP@K для каждого объекта и усредняем:

$$map@K = \frac{1}{N} \sum_{j=1}^N ap@K_j$$

Как вы думаете, что такое MNAP@K?











 = relevant documents for query 1

Ranking #1

										
Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

 = relevant documents for query 2

Ranking #2

										
Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

average precision query 1 = $(1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$

average precision query 2 = $(0.5 + 0.4 + 0.43)/3 = 0.44$

mean average precision = $(0.62 + 0.44)/2 = 0.53$

Discounted Cumulative Gain (DCG@k)

Чем выше релевантные элементы в списке рекомендаций, тем лучше.

$$CG@K = \sum_{k=1}^K r^{true}(\pi^{-1}(k))$$

$$DCG@K = \sum_{k=1}^K \frac{r^{true}(\pi^{-1}(k))}{\log_2(k+1)}.$$

Normalized Discounted Cumulative Gain (DCG@k)

$$nDCG@K = \frac{DCG@K}{IDCG@K}$$

где $IDCG@K$ - это идеальное (максимальное) значение $DCG@K$:

$$IDCG@K = \sum_{k=1}^K \frac{1}{\log_2(k+1)}$$

NDCG@k

Пример вычисления DCG и nDCG:

Дано множество документов, где каждый документ оценивается от 3 до 0, где 3 — очень релевантен, а 0 — не релевантен. Пусть таким множеством будет $S = \{D_1, D_2, D_3, D_4, D_5, D_6\}$, где оценка релевантности по опросу пользователей задается (в том же порядке) множеством $R = \{3, 2, 3, 0, 1, 2\}$.

Тогда $DCG@6 = \sum_{i=1}^6 \frac{rel_i}{\log(i+1)} = 3 + 1.262 + 1.5 + 0 + 0.387 + 0.712 = 6.861$.

i	rel_i	$\log(i+1)$	$\frac{rel_i}{\log(i+1)}$
1	3	1	3
2	2	1.585	1.262
3	3	2	1.5
4	0	2.322	0
5	1	2.585	0.387
6	2	2.807	0.712

NDCG@k

Идеальный порядок оценок релевантности $Ideal = \{3, 3, 2, 2, 1, 0\}$. DCG для данного множества будет следующим:

$$maxDCG@6 = \sum_{i=1}^6 \frac{rel_i}{\log(i+1)} = 3 + 1.893 + 1 + 0.861 + 0.387 + 0 = 7.141.$$

i	rel_i	$\log(i+1)$	$\frac{rel_i}{\log(i+1)}$
1	3	1	3
2	3	1.585	1.893
3	2	2	1
4	2	2.322	0.861
5	1	2.585	0.387
6	0	2.807	0

38

$$\text{Итого } nDCG@6 = \frac{DCG@6}{maxDCG@6} = \frac{6.861}{7.141} = 0.961.$$

Overview текущих подходов

Одна из таксономий

- **General** подходы (Matrix Factorization, Factorization Machines, Collaborative Filtering, user/item similarity based methods ...)
- **Content-based** models
- **Context-based** models
- **Sequential** based models
- **RL** for RecSys
- **Hybrid** models

Одна из таксономий

- **General** подходы (Matrix Factorization, Factorization Machines, Collaborative Filtering, user/item similarity based methods ...)
- **Content-based** models
- **Context-based** models
- **Sequential** based models
- **RL** for RecSys
- **Hybrid** models

Текущие проблемы RecSys моделей

- Большой профиль интересов у каждого пользователя
- Большие размерности данных
- Люди сами часто не знают чего хотят
- Недостаточно интеракций в разрезе на пользователя
- Проблема холодного старта
- Нестационарные интересы
- Зависимость от контекста и внешних факторов (настроение, погода, время суток, сезон и тд)
- Проблема подсчета метрик на исторических данных
- Разнообразие метрик без стандартизации и много еще чего...

Основные конференции с RecSys

- RecSys
 - KDD
 - SIGIR
 - NeurIPS
 - WWW
- et al.

Недостатки статей по RecSys

- Отсутствие сильных бейзлайнов
- Невоспроизводимость результатов
- Различия в метриках (k параметр, набор метрик)
- Разные датасеты
- Отсутствие выводов по неудачным экспериментам
- Отсутствие open-source кода

Тренды

- DL
- Causality
- Bandits & RL
- Interpretability & Explainability

Рекомендованные источники

- [ACM RecSys](#) 2020 tutorials and workshops
- Лекция Евгения Соколова по [задаче ранжирования](#). Да и в целом [курс](#) очень-очень хорош
- Лекция Евгения Соколова по [memory-based подходам, коллаборативной фильтрации, MF](#).
- Воронцов [про коллаборативную фильтрацию](#)
- Лекция Сергея Николенко про [подходы для рекомендательных систем](#)
- [ITMO Wiki. Задача ранжирования](#)
- Библиотека [RecBole](#) с 50+ алгоритмами RecSys и [комментарии про нее на митапе Tinkoff.RecSys](#)
- Библиотека [Microsoft recommenders](#) (много хороших реализаций моделей и ноутбуков к ним)
- Библиотека [implicit](#) в частности с оптимальной реализацией ALS. Оригинальная статья по [ALS](#)
- Библиотека [LightFM](#) с реализацией этого алгоритма по [статье](#). Это контентная модель - то есть, можно обучать как ALS, а можно еще добавлять признаки по пользователям и объектам.
- Обучение градиентного бустинга для задачи рекомендаций. [Catboost ranking task and metrics](#)
- [Рекомендации в Okko: как заработать сотни миллионов, перемножив пару матриц](#)
- Яндекс.Дзен [митапы по RecSys](#)