

Лекция 10. Ранжирование и рекомендательные системы

Александр Юрьевич Авдюшенко

МКН СПбГУ

21 апреля 2022



Факультет
математики
и компьютерных
наук
СПбГУ

- ▶ Какие типы зависимостей выделяют во временных рядах?
- ▶ Опишите парой предложений на идейном уровне модель прогнозирования ARIMA
- ▶ Что такое «гетероскедастичность»?

Ранжирование (learning to rank)

X — множество объектов

$X^\ell = \{x_1, \dots, x_\ell\}$ — обучающая выборка

$i \prec j$ — правильный порядок на парах $(i, j) \in \{1, \dots, \ell\}^2$

Задача:

построить ранжирующую функцию $a : X \rightarrow \mathbb{R}$ такую, что

$i \prec j \Rightarrow a(x_i) < a(x_j)$

Источники правильных ответов



ассесоры за работой

СПбГУ, Факультет математики и компьютерных наук

Обзор Фото 5 Отзывы 4 Особенности

4 отзыва

[По умолчанию ▾](#)

лучшее место - 2

лучший факультет - 1



Иван П.

Знаток города 2 уровня

★★★★★ 13 октября

Лучший факультет

👍 🗳️ 1

— ответить



Андрей К.

Знаток города 2 уровня

★★★★★ 5 марта

Лучшее место на свете

👍 🗳️ 1

Примеры задач ранжирования

Поисковая система

x_i — (запрос, i -ый документ)

Примеры задач ранжирования

Поисковая система

x_i — (запрос, i -ый документ)

Социальная сеть

x_i — (пользователь, i -ая единица контента)

Факторы ранжирования в социальной сети

Пользователь

- ▶ возраст, пол и т.д.
- ▶ content-based описание (TF-IDF, тематическое моделирование, embeddings)
- ▶ проявленные интересы (группы, подписки)

Факторы ранжирования в социальной сети

Пользователь

- ▶ возраст, пол и т.д.
- ▶ content-based описание (TF-IDF, тематическое моделирование, embeddings)
- ▶ проявленные интересы (группы, подписки)

Контент

- ▶ content-based описание (TF-IDF, тематическое моделирование, embeddings)
- ▶ проявленная заинтересованность к данному контенту
- ▶ рейтинг/популярность автора контента или самого контента (ссылки, лайки)

Факторы ранжирования в социальной сети

Пользователь

- ▶ возраст, пол и т.д.
- ▶ content-based описание (TF-IDF, тематическое моделирование, embeddings)
- ▶ проявленные интересы (группы, подписки)

Контент

- ▶ content-based описание (TF-IDF, тематическое моделирование, embeddings)
- ▶ проявленная заинтересованность к данному контенту
- ▶ рейтинг/популярность автора контента или самого контента (ссылки, лайки)

Общие

- ▶ повестка дня
- ▶ время суток

Content-based факторы. TF-IDF

TF-IDF (Term Frequency - Inverse Document Frequency)

Например, переводим текст в $[\text{tfidf}(w_1), \dots, \text{tfidf}(w_k)]$

$$\text{tfidf}(w_i) = \text{tf}_i \cdot \text{idf}_i = \text{tf}_i \cdot \log \frac{N}{N_i}$$

$\text{tf}_i = \frac{n_i}{\sum_j n_j}$ — частота i -ого слова в тексте,

N — число текстов, N_i — число текстов с данным словом

Content-based факторы

напоминание

Тематическое моделирование

Тема — условное дискретное вероятностное распределение на множестве терминов

$p(w|t)$ — вероятность термина w в теме t

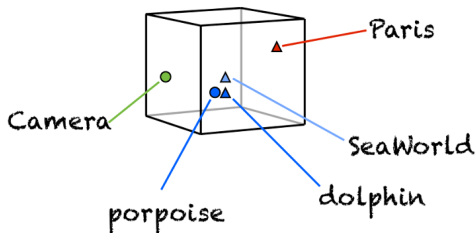
Тематический профиль документа — условное распределение

$p(t|d)$ — вероятность темы t в документе d

В базовой модели признак может быть бинарным — наличие или отсутствие темы.

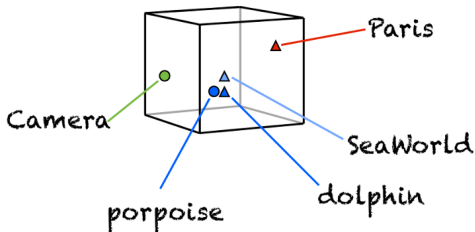
Эмбединги (embeddings)

Работает всё пройденное: при помощи нейронных сетей получаем вложения текстов и \ или изображений в векторное пространство.



Эмбединги (embeddings)

Работает всё пройденное: при помощи нейронных сетей получаем вложения текстов и \ или изображений в векторное пространство.



Вопрос

Как учесть взаимные ссылки между страницами в интернете?

Легендарный PageRank

интуиция

Документ **d** тем важнее, чем

- ▶ больше других документов **c** ссылаются на **d**
- ▶ важнее эти документы **c**
- ▶ меньше других ссылок имеют эти документы **c**



S. Brin, L. Page. The anatomy of a large-scale hypertextual Web search engine. 1998, Computer Networks and ISDN Systems, 30 (1-7): 107-117.

Вероятность попасть на страницу d , если кликать случайно:

$$PR(d) = \frac{1 - \delta}{N} + \delta \sum_{c \in D_d^{in}} \frac{PR(c)}{D_c^{out}}$$

- ▶ $D_d^{in} \subset D$ — множество документов, ссылающихся на d
- ▶ $D_c^{out} \subset D$ — документы, на которые ссылается c
- ▶ $\delta = 0.85$ — вероятность продолжать клики (damping factor)
- ▶ N — число документов в коллекции D

PageRank. Теорема Фробениуса-Перрона

Матрицу ссылок между сущностями можно представить в виде

$$P = \begin{pmatrix} p_{11} & \dots & p_{n1} \\ \dots & \dots & \dots \\ p_{1n} & \dots & p_{nn} \end{pmatrix}$$

где $p_{cd} = \frac{1}{|D_c^{out}|}$ — вероятность перейти с c -ой сущности на d -ую

$$\text{и } \sum_{i=1}^n p_{ci} = 1$$

Предположим, что все $p_{ij} > 0$, тогда по следствию из теоремы Фробениуса-Перрона существует собственный вектор

$$x = \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix}$$

такой что $P \cdot x = x$, тогда PageRank = x

Вопрос

Как измерить качество ранжирования?

Метрики качества ранжирования 1

Пусть $Y = \{0, 1\}$, $y(q, d)$ — релевантность

$a(q, d)$ — искомая функция ранжирования,

$d_q^{(i)}$ — i -й документ по убыванию $a(q, d)$

Precision (точность) — доля релевантных среди первых n :

$$P_n(q) = \frac{1}{n} \sum_{i=1}^n y(q, d_q^{(i)})$$

Average Precision — средняя P_n по позициям релевантных документов:

$$AP(q) = \sum_n y(q, d_q^{(n)}) P_n(q) / \sum_n y(q, d_q^{(n)})$$

Mean Average Precision — средняя AP по всем запросам:

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} AP(q)$$

Метрики качества ранжирования 2

Пусть $Y \subset \mathbb{R}$, $y(q, d)$ — релевантность,
 $a(q, d)$ — искомая функция ранжирования,
 $d_q^{(i)}$ — i -й документ по убыванию $a(q, d)$

Доля инверсий порядка среди первых n документов:

$$DP_n(q) = \frac{2}{n(n-1)} \sum_{i < j}^n \left[y(q, d_q^{(i)}) < y(q, d_q^{(j)}) \right]$$

Метрики качества ранжирования 3

Пусть $Y \subset \mathbb{R}$, $y(q, d)$ — релевантность,

$a(q, d)$ — искомая функция ранжирования,

$d_q^{(i)}$ — i -й документ по убыванию $a(q, d)$

Дисконтированная (взвешенная) сумма выигрышей:

$$DCG_n(q) = \sum_{i=1}^n \underbrace{G_q(d_q^{(i)})}_{gain} \cdot \underbrace{D(i)}_{discount}$$

$G_q(d) = (2^{y(q,d)} - 1)$ — большой вес релевантным документам

$D(i) = 1 / \log_2(i + 1)$ — большой вес в начале выдачи

Нормированная дисконтированная сумма выигрышей:

$$NDCG_n(q) = \frac{DCG_n(q)}{\max DCG_n(q)}$$

$\max DCG_n(q)$ — это $DCG_n(q)$ при идеальном ранжировании

Подходы к построению моделей ранжирования

- ▶ point-wise — поточечный
- ▶ pair-wise — попарный
- ▶ list-wise — списочный

RankNet: гладкий функционал качества ранжирования

$$Q(a) = \sum_{i \prec j} \mathcal{L}(a(x_j) - a(x_i)) \rightarrow \min$$

при $\mathcal{L}(M) = \log(1 + e^{-\sigma M})$ и линейной модели $a(x) = \langle w, x \rangle$

Метод стохастического градиента:

выбираем на каждой итерации случайную пару $i \prec j$:

$$w := w + \eta \cdot \frac{\sigma}{1 + \exp(\sigma \langle x_j - x_i, w \rangle)} \cdot (x_j - x_i)$$

Christopher J.C. Burges. From RankNet to LambdaRank to LambdaMART: An Overview // Microsoft Research Technical Report MSR-TR-2010-82. 2010.

List-wise. От RankNet к LambdaRank

Метод стохастического градиента:

$$w := w + \eta \cdot \underbrace{\frac{\sigma}{1 + \exp(\sigma \langle x_j - x_i, w \rangle)}}_{\lambda_{ij}} \cdot (x_j - x_i)$$

Оказывается для оптимизации негладких функционалов *MAP*, *NDCG*, *pFound* достаточно домножить λ_{ij} на изменение данного функционала при перестановке местами $x_i \rightleftharpoons x_j$

LambdaRank: домножение на изменение *NDCG* при $x_i \rightleftharpoons x_j$ приводит к её оптимизации

$$w := w + \eta \cdot \frac{\sigma}{1 + \exp(\sigma \langle x_j - x_i, w \rangle)} \cdot |\Delta NDCG_{ij}| \cdot (x_j - x_i)$$

Christopher J.C. Burges. From RankNet to LambdaRank to LambdaMART: An Overview // Microsoft Research Technical Report MSR-TR-2010-82. 2010.

List-wise. LambdaMART

MART (Multiple Additive Regression Trees) = Gradient Boosting

$$f_{T,i} := f_{T-1,i} + \alpha b(x_i), i = 1, \dots, \ell$$

Идея: будем искать такой базовый алгоритм b_T , чтобы вектор $(b_T(x_i))_{i=1}^{\ell}$ приближал вектор антиградиента $(-g_i)_{i=1}^{\ell}$

$$b_T := \arg \min_b \sum_{i=1}^{\ell} (b(x_i) + g_i)^2$$

Пусть $\eta_{ij} = \lambda_{ij} \cdot |\Delta NDCG_{ij}|$, пусть $\eta_i = \sum_{(i,j) \in I} \eta_{ij} - \sum_{(j,i) \in I} \eta_{ji}$

Положим $g_i = \eta_i$

won Track 1 of the 2010 Yahoo! Learning To Rank Challenge

Christopher J.C. Burges. From RankNet to LambdaRank to LambdaMART: An Overview // Microsoft Research Technical Report MSR-TR-2010-82. 2010.



Рекомендательные системы. Постановка задачи

U — множество субъектов (users)

I — множество объектов (items)

Y — оценки (ratings)

Даны

$$(u_t, i_t, y_t) \in D \subset U \times I \times Y$$

нужно предсказать для u для $(u, i) \notin D$

$R = (r_{ui})_{U \times I}$ — матрица отношений

Рекомендательные системы. Пример

Посты:



Музыка:



О себе: моим
предкам издревле
приписывали
магические качества.
Купил: шляпу фокусника
Лайкнул:



Товары:

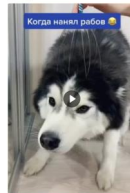


Фокусы XXL Tactic Games ...
1 450 Р 1 770 Р
Почемучкин - магазин игр...



Степ. Школа волшебства 1...
401 P
miki.shop

Клипы:



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

Исторически она использовалась в легендарном конкурсе
Netflix: 2007 год, \$1 млн

Подход 1 [оригинально Amazon.com]:

«клиенты, купившие i_0 , также покупали $I(i_0)$ »

Недостатки:

- ▶ рекомендации тривиальны (просто наиболее популярное предлагается)
- ▶ не учитываются интересы конкретного пользователя u_0
- ▶ проблема «холодного старта» — новый товар никому не рекомендуется
- ▶ надо хранить всю матрицу R

Подход 2:

«клиенты, похожие на u_0 , также покупали $I(u_0)$ »

- ▶ рекомендации тривиальны
- ▶ не учитываются интересы конкретного пользователя u_0
- ▶ проблема «холодного старта» — новый товар никому не рекомендуется
- ▶ надо хранить всю матрицу R
- ▶ нечего рекомендовать нетипичным/новым пользователям

Подход 3:

«вместе с объектами, которые покупал u_0 , часто покупают $I(u_0)$ »

- ▶ рекомендации часто тривиальны (нет коллаборативности)
- ▶ ~~не учитываются интересы конкретного пользователя u_0~~
- ▶ проблема «холодного старта» — новый товар никому не рекомендуется
- ▶ надо хранить всю матрицу R
- ▶ ~~нечего рекомендовать нетипичным/новым пользователям~~

Латентные модели рекомендаций

$$u \rightarrow [p_{1u}, \dots, p_{tu}, \dots, p_{Tu}] = p_u$$

$$i \rightarrow [q_{1i}, \dots, q_{ti}, \dots, q_{Ti}] = q_i$$

$$T \ll |U|, T \ll |I|$$

$$\langle p_u, q_i \rangle = \sum_{t=1}^T p_{tu} q_{ti} \text{ — рейтинг } i\text{-го объекта для } u\text{-ого субъекта}$$

Singular Value Decomposition (SVD-разложение)

$R = P^T \Delta Q$ — матрица рейтингов

$$\|R - P^T \Delta Q\|^2 \rightarrow \min_{P, \Delta, Q}$$

Матрицы P и Q должны быть ортогональны, но на практике это может нам мешать.

Модель латентных факторов (LFM)

$$J(\Theta) = \sum_{(u,i) \in D} \left(r_{ui} - \bar{r}_u - \bar{r}_i - \sum_{t \in T} p_{tu} q_{ti} \right)^2 \rightarrow \min_{P, Q}$$

$$p_{tu} = p_{tu} + \eta \varepsilon_{ui} q_{ti}, t \in T$$

$$q_{ti} = q_{ti} + \eta \varepsilon_{ui} p_{tu}, t \in T$$

\bar{r}_u — усреднение по оценкам пользователя (есть восторженные, у которых оценки рейтинга это 4 и 5, есть наоборот ставящие только 1 и 3 из 5)

\bar{r}_i — аналогично по объектам (сложно ненавидеть котика)

Модель латентных факторов (LFM). Достоинства

легко вводится регуляризация:

$$\varepsilon_{ui}^2 + \lambda \|p_u\|^2 + \mu \|q_i\|^2 \rightarrow \min_{p_u, q_i}$$

легко вводятся ограничения неотрицательности:

$p_{tu} \geq 0, q_{ti} \geq 0$ (метод проекции градиента)

легко вводится обобщение для ранговых данных:

$$\sum_{(u,i) \in D} \left(r_{ui} - \bar{r}_u - \bar{r}_i - \beta \left(\sum_{t \in T} p_{tu} q_{ti} \right) \right)^2 \rightarrow \min_{P, Q, \beta}$$

легко реализуются добавления ещё одного:

- ▶ клиента u
- ▶ объекта i
- ▶ значения r_{ui}

высокая численная эффективность на больших данных

Alternating Least Squares (ALS)

Попеременно точно (аналитически) находим минимумы то по одним координатам, то по другим:

$$p_u^*(\Theta) = \arg \min_{p_u} J(\Theta) = \left(Q_u^T Q_u + \lambda I \right)^{-1} Q_u^T r_u$$

$$q_i^*(\Theta) = \arg \min_{q_i} J(\Theta) = \left(P_i^T P_i + \lambda I \right)^{-1} P_i^T r_i$$

Работает достаточно быстро, так как каждый шаг можно распараллелить. Используется в известном фреймворке Spark обработки больших данных от Apache.

Учет неявной информации

Пусть имеются

r_{ui} — явные оценки (explicit)

s_{ui} — неявные оценки (implicit)

Пусть $c_{ui} = 1 + \alpha r_{ui}$ — взвешиваем имеющимися явными оценками

$$\sum_{(u,i) \in D} c_{ui} \left(s_{ui} - \bar{s}_u - \bar{s}_i - \sum_{t \in T} p_{tu} q_{ti} \right)^2 + \lambda \sum_{u \in U} \|p_u\|^2 + \mu \sum_{i \in I} \|q_i\|^2 \rightarrow \min_{P, Q}$$

Bell R.M., Koren Y., Volinsky C. The BellKor 2008 solution to the Netflix Prize.

Метрики в рекомендательных системах

Недостатки RMSE:

- ▶ у каждого пользователя свое представление о шкале оценок. Пользователи, у которых разброс оценок более широкий, будут больше влиять на значение метрики, чем другие
- ▶ Ошибка в предсказании высокой оценки имеет такой же вес, что и ошибка в предсказании низкой оценки. При этом в реальности предсказать оценку 9 вместо настоящей оценки 7 страшнее, чем предсказать 4 вместо 2 (по десятибалльной шкале)
- ▶ Можно иметь почти идеальную метрику RMSE, но иметь очень плохое качество ранжирования, и наоборот

Часто лучше использовать метрики ранжирования:

- ▶ MAP
- ▶ NDCG

Нетривиальные метрики качества

Другие свойства рекомендаций, влияющие на качество.

Для пользователя:

- ▶ разнообразие (diversity)
- ▶ неожиданность (surprise)
- ▶ новизна (novelty)
- ▶ догадливость (serendipity)

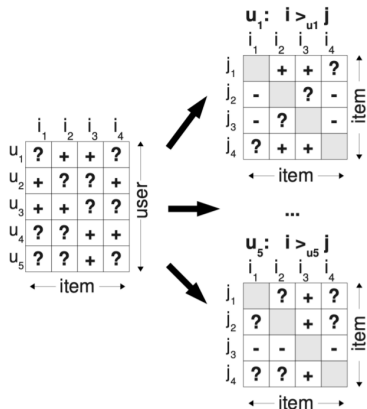
Для бизнеса:

- ▶ покрытие (coverage)
- ▶ заинтересованность в платформе

Bayesian Personalized Ranking (BPR)

На входе

- ▶ по сути только факт взаимодействия пользователь-объект
- ▶ Нет дизлайков :)



$$D_S = \{(u, i, j) | i \in I_u^+ \wedge j \in I \setminus I_u^+\}$$

$$p(\Theta | >_u) \propto p(>_u | \Theta) p(\Theta)$$

$$\begin{aligned} \prod_{u \in U} p(>_u | \Theta) &= \\ &= \prod_{(u,i,j) \in U \times I \times I} p(i >_u j | \Theta)^{\delta((u,i,j) \in D_S)} \cdot (1 - p(i >_u j | \Theta))^{\delta((u,j,i) \notin D_S)} = \\ &= \{\text{так как по всем парам сумма}\} \\ &= \prod_{(u,i,j) \in D_S} p(i >_u j | \Theta) \end{aligned}$$

$$\begin{aligned} p(i >_u j | \Theta) &= \sigma(\hat{x}_{uij}(\Theta)) \\ \hat{x}_{uij} &= \hat{x}_{ui} - \hat{x}_{uj} = \hat{r}_{ui} - \hat{r}_{uj} \\ \sigma(x) &= \frac{1}{1 + e^{-x}} \end{aligned}$$

Оптимизация привычным градиентным спуском.

BPR (Bayesian Personalized Ranking). Достоинства

- ▶ разумный учет неявного фидбека
- ▶ ориентированность на ранжирование

BPR (Bayesian Personalized Ranking). Достоинства

- ▶ разумный учет неявного фидбека
- ▶ ориентированность на ранжирование

Вопрос

Как быть с новыми пользователями и объектами?

BPR (Bayesian Personalized Ranking). Достоинства

- ▶ разумный учет неявного фидбека
- ▶ ориентированность на ранжирование

Вопрос

Как быть с новыми пользователями и объектами?

Вариант 1. Обучаем модель предсказывать скрытое состояние по пользователю.

Вариант 2. (для объектов) Многорукие бандиты — с некоторой вероятностью показываем каждый новый объект пользователю и пересчитываем вероятности, накапливая статистику взаимодействия.

Как обычно происходит на практике?

1. Сначала происходит быстрая генерация кандидатов (например, <https://github.com/facebookresearch/faiss>)
2. После этого применяется более тяжелая и точная ранжирующая модель

- ▶ постановка задачи ранжирования, возможные метрики
- ▶ подходы к построению моделей ранжирования:
поточечный, попарный, списочный
- ▶ SVD, LFM, ALS
- ▶ BPR

- ▶ постановка задачи ранжирования, возможные метрики
- ▶ подходы к построению моделей ранжирования:
поточечный, попарный, списочный
- ▶ SVD, LFM, ALS
- ▶ BPR

Что ещё можно посмотреть?

- ▶ Доклад про рекомендации ленты ВКонтакте на Open DataFest 2020
- ▶ Рекомендательная система на базе DataSphere