

Методы обучения ранжированию (Learning to Rank)

2 ноября 2017

Содержание

- 1 **Признаки и функционалы качества**
 - Построение признаков
 - Функционалы качества ранжирования
- 2 **Основные подходы к ранжированию**
 - Поточечный подход
 - Попарный подход
 - Списочный подход
- 3 **Обучение ранжированию по кликам**
 - Balanced interleaving
 - Teamdraftinterleaving(TDI)
 - Проблемы подхода

Определения и обозначения

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

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

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. Задача ранжирования поисковой выдачи

D — коллекция текстовых документов (documents)

Q — множество запросов (queries)

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

$X = Q \times D$ — объектами являются пары «запрос, документ»:

$$x \equiv (q, d), \quad q \in Q, \quad d \in D_q$$

Y — упорядоченное множество рейтингов

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

Правильный порядок определён только между документами, найденными по одному и тому же запросу q :

$$(q, d) \prec (q, d') \Leftrightarrow y(q, d) < y(q, d')$$

Именно об этой задаче сегодня пойдет речь.

Пример 2. Коллаборативная фильтрация

U — пользователи, users

I — предметы, items (фильмы, книги, и т.п.)

$X = U \times I$ — объектами являются пары «user, item»

Правильный порядок определён между предметами, которые выбирал или рейтинговал один и тот же пользователь:

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

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

В роли признаков объекта $x = (u, i)$ могут выступить $y(u', i)$ — рейтинги, поставленные другими пользователями u'

То есть, поиск коллаборации \Leftrightarrow отбор признаков

Задача ранжирования поисковой выдачи: типы признаков

- функции только документа d
- функции только запроса q
- функции запроса и документа (q, d)
- текстовые
 - слова запроса q встречаются в d чаще обычного
 - слова запроса q есть в заголовках или выделены в d
- ссылочные
 - на документ d много ссылаются
 - документ d содержит много полезных ссылок
- кликовые
 - на документ d часто кликают
 - на документ d часто кликают по запросу q

Могут быть и другие варианты: например, свежесть документа.

TF-IDF(q, d) — классический текстовый признак

TF-IDF(q, d) — мера релевантности документа d запросу q
 n_{dw} (term frequency) — число вхождений слова w в текст d ;
 N_w (document frequency) — число документов, содержащих w ;
 N — число документов в коллекции D ;

N_w/N — оценка вероятности встретить слово w в документе;

$(N_w/N)^{n_{dw}}$ — оценка вероятности встретить его n_{dw} раз;

$P(q, d) = \prod_{w \in q} (N_w/N)^{n_{dw}}$ — оценка вероятности встретить

в документе d слова запроса $q = \{w_1, \dots, w_k\}$ *чисто случайно*;

Оценка релевантности запроса q документу d :

$$-\log P(q, d) = \sum_{w \in q} \underbrace{n_{dw}}_{\text{TF}(w, d)} \underbrace{\log(N/N_w)}_{\text{IDF}(w)} \rightarrow \max.$$

$\text{TF}(w, d) = n_{dw}$ — term frequency;

$\text{IDF}(w) = \log(N/N_w)$ — inverted document frequency.

PageRank — классический ссылочный признак

- Документ d тем важнее,
- чем больше других документов c ссылаются на d ,
 - чем важнее документы c , ссылающиеся на d ,
 - чем меньше других ссылок имеют эти документы c .

Вероятность попасть на страницу d , если кликать случайно:

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

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

Sergey Brin, Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. 1998.

Кликовые признаки

Можно придумать множество различных «кликовых функций»

- Кликнутые против некликнутых.
- Длинные клики против коротких кликов или некликнутых.

Решение подзадач для формирования признаков:

- Классификация: есть ли длинный клик на документе?
- Регрессия: насколько длинный клик ожидаем?
- Попарная классификация: будет ли клик на один документ более длинным, чем на другой?

Онлайн- и офлайн-признаки

- **Офлайн-признаки:** можем вычислить в любой момент для любой поисковой системы без использования пользовательской информации
 - Собираем много запросов, по ним много документов, оцениваем на «релевантность» силами ассессоров
- **Онлайн-признаки:** меняем ранжирование для некоторой группы пользователей и следим за их поведением
 - АВ-тестирование: разбиваем пользователей на две группы («контроль» и «эксперимент»), вносим изменение для одной из этих групп и затем сравниваем группы по каким-нибудь показателям
 - Высокоуровневые признаки: количество запросов/кликов (конвертируются в долю). Низкоуровневые признаки: доля запросов без кликов, количество инверсий, позиция первого клика, средняя длина клика и т.п.

Точность и средняя точность

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:

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

Mean Average Precision, средняя AP по всем запросам:

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

Доля «дефектных пар»

Пусть $Y \subseteq \mathbb{R}$, $y(q, d)$ — 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:

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

Связь с коэффициентом ранговой корреляции (τ Кенделла):

$$\tau(a, y) = 1 - 2 \cdot DP_n(q).$$

Связь с AUC (area under ROC-curve) в задачах классификации
с двумя классами $Y = \{-1, +1\}$, $a: X \rightarrow Y$

$$AUC_n(q) = \frac{1}{\ell_- \ell_+} \sum_{i,j=1}^n [y_i < y_j] [a(x_i) < a(x_j)] = 1 - \frac{n(n-1)}{2\ell_- \ell_+} \cdot DP_n(q).$$

ℓ_- и ℓ_+ — количество объектов в классах -1 и $+1$ соответственно.

DCG — Discounted Cumulative Gain

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

Дисконтированная (взвешенная) сумма выигрышей:

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

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

$D(i) = 1/\log_2(i + 1)$ — бóльший 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

pFound — модель поведения пользователя

Пусть $Y \subseteq [0, 1]$,

$y(q, d)$ — релевантность, **оценка вероятности найти ответ в d** ,

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

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

Вероятность найти ответ в первых n документах:

$$pFound_n(q) = \sum_{i=1}^n P_i \cdot y(q, d_q^{(i)}),$$

где P_i — вероятность дойти до i -го документа:

$$P_1 = 1;$$

$$P_{i+1} = P_i \cdot (1 - y(q, d_q^{(i)})) \cdot (1 - P_{out}),$$

где P_{out} — вероятность прекратить поиск без ответа

pFound — модель поведения пользователя

Параметры критерия pFound:

- $P_{out} = 0.15$ — вероятность прекратить поиск без ответа;
- $y(q, d)$ — оценка вероятности найти ответ в документе:

оценка асессора	$y(q, d)$
Vital	0.61
Useful	0.41
Relevant+	0.14
Relevant—	0.07
Not Relevant	0.00

Гулин А., Карпович П., Расковалов Д., Сегалович И.

Оптимизация алгоритмов ранжирования методами машинного обучения // РОМИП-2009.

Основные подходы к ранжированию

- Point-wise — поточечный
- Pair-wise — попарный
- List-wise — списочный

Переход к гладкому функционалу качества ранжирования:

$$Q(a) = \sum_{i \prec j} \underbrace{[a(x_j) - a(x_i) < 0]}_{\text{Margin}(i,j)} \leq \sum_{i \prec j} \mathcal{L}(a(x_j) - a(x_i)) \rightarrow \min$$

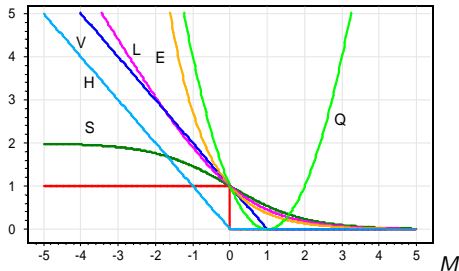
где $a(x)$ — алгоритм ранжирования;

$\mathcal{L}(M)$ — убывающая непрерывная функция отступа $\text{Margin}(i, j)$:

- $\mathcal{L}(M) = (1 - M)_+$ — RankSVM
- $\mathcal{L}(M) = \exp(-M)$ — RankBoost
- $\mathcal{L}(M) = \log(1 + e^{-M})$ — RankNet

Непрерывные аппроксимации пороговой функции потерь

Часто используемые непрерывные функции потерь $\mathcal{L}(M)$:



$$V(M) = (1 - M)_+$$

— кусочно-линейная (SVM);

$$H(M) = (-M)_+$$

— кусочно-линейная (Hebb's rule);

$$L(M) = \log_2(1 + e^{-M})$$

— логарифмическая (LR);

$$Q(M) = (1 - M)^2$$

— квадратичная (FLD);

$$S(M) = 2(1 + e^M)^{-1}$$

— сигмоидная (ANN);

$$E(M) = e^{-M}$$

— экспоненциальная (AdaBoost);

$[M < 0]$

— пороговая функция потерь.

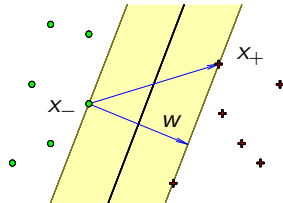
Напоминание: SVM — метод опорных векторов

Линейный классификатор:

$$a(x) = \text{sign}(\langle w, x \rangle - w_0), \quad w, x \in \mathbb{R}^n, \quad w_0 \in \mathbb{R}.$$

Задача обучения SVM:

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\ell} \xi_i \rightarrow \min_{w, w_0, \xi}; \\ M_i(w, w_0) \geq 1 - \xi_i, \quad i = 1, \dots, \ell; \\ \xi_i \geq 0, \quad i = 1, \dots, \ell. \end{cases}$$



где $M_i(w, w_0) = y_i(\langle w, x_i \rangle - w_0)$ — отступ объекта x_i .

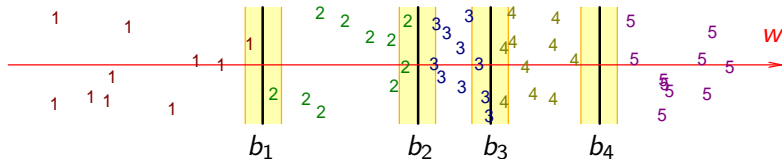
Эквивалентная задача безусловной минимизации:

$$Q(w, w_0) = \sum_{i=1}^{\ell} (1 - M_i(w, w_0))_+ + \frac{1}{2C} \|w\|^2 \rightarrow \min_{w, w_0}.$$

Ранговая классификация OC-SVM (Ordinal Classification SVM)

Пусть $Y = \{1, \dots, K\}$, функция ранжирования *линейная* с порогоми $b_0 = -\infty$, $b_1, \dots, b_{K-1} \in \mathbb{R}$, $b_K = +\infty$:

$$a(x) = y, \text{ если } b_{y-1} < \langle w, x \rangle \leq b_y$$



Постановка задачи SVM для ранговой классификации:

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\ell} [y_i \neq K] (\xi_i + \xi_i^*) \rightarrow \min_{w, b, \xi}; \\ b_{y_i-1} + 1 - \xi_i^* \leq \langle w, x_i \rangle \leq b_{y_i} - 1 + \xi_i; \\ \xi_i^* \geq 0, \quad \xi_i \geq 0. \end{cases}$$

Ranking SVM

Постановка задачи SVM для попарного подхода:

$$Q(a) = \frac{1}{2}\|w\|^2 + C \sum_{i \prec j} \underbrace{\mathcal{L}(a(x_j) - a(x_i))}_{\text{Margin}(i,j)} \rightarrow \min_a,$$

где $a(x) = \langle w, x \rangle$ — функция ранжирования,

$\mathcal{L}(M) = (1 - M)_+$ — функция потерь,

$M = \text{Margin}(i, j) = \langle w, x_j - x_i \rangle$ — отступ,

Постановка задачи квадратичного программирования:

$$\begin{cases} \frac{1}{2}\|w\|^2 + C \sum_{i \prec j} \xi_{ij} \rightarrow \min_{w, \xi}; \\ \langle w, x_j - x_i \rangle \geq 1 - \xi_{ij}, & i \prec j; \\ \xi_{ij} \geq 0, & i \prec j. \end{cases}$$

RankNet

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 q , $i \prec j$ случайно,

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

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

От RankNet до 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, rFound достаточно домножить λ_{ij} на изменение данного функционала при перестановке местами $x_i \leftrightarrow x_j$.

LambdaRank: домножение на изменение NDCG при $x_i \leftrightarrow x_j$ приводит к оптимизации NDCG:

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

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

LambdaMART

- **MART** (Multiple Additive Regression Trees) — градиентный бустинг над решающими деревьями.
- **LambdaMART** — совмещает MART и LambdaRank — к градиентам добавляется целевая метрика. Например, *NDCG*.
- Вариант: **LambdaSMART** (submodel): мы инициализируем первое дерево какой-нибудь обученной хорошей базовой моделью, а всё дальнейшее — это её уточнение.

Обучение ранжированию по кликам

- Качаем по запросу пользователя выдачу нескольких поисковых систем (Google, MSN, Excite, Altavista, Hotbot).
- Если пользователь кликал по выдаче, то считаем, что кликнутые документы – лучше, чем некликнутые, расположенные выше в ранжировании.
- На полученных парах обучаем SVM.
- Сравниваем различные варианты ранжирования по количеству генерируемых кликов – balanced interleaving.

Joachims T. Optimizing Search Engines using Clickthrough Data (2002)
http://www.cs.cornell.edu/people/tj/publications/joachims_02c.pdf

Balanced interleaving

- Пусть k – некоторый размер топа для выдачи I , k_a и k_b – количество документов из A и B в этом топе соответственно. Тогда для любого k должно быть выполнено $|k_a - k_b| \leq 1$.
- Пусть по данному запросу были кликнуты документы $h_a \in A \cap C$ и $h_b \in B \cap C$. Тогда в зависимости от знака величины $|h_a| - |h_b|$ засчитываем победу A , победу B , либо ничью.
- Сравнение на наборе запросов:

$$\Delta_{AB} = \frac{wins(A) + 0,5ties(A, B)}{wins(A) + wins(B) + ties(A, B)} - 0,5$$

- Если $\Delta_{AB} > 0$, то система A предпочтительнее. Если $\Delta_{AB} < 0$, то система B предпочтительнее.

Thorsten Joachims. Evaluating Retrieval Performance using Clickthrough Data (2002) https://www.cs.cornell.edu/people/tj/publications/joachims_02b.pdf

Teamdraftinterleaving(TDI)

- Делаем n шагов смешивания. На каждом шаге выбираем одну из двух систем случайным образом. Из выбранной системы добавляем в I наиболее высоко ранжированный документ, которого в I еще не было. Затем делаем то же самое для второй из систем.
- За каждой позицией в I закреплена только одна из систем.
- Системы сравниваются по общему количеству кликов аналогично предыдущему подходу.

Radlinski F., Kurup M., Joachims T. How does clickthrough data reflect retrieval quality (2008)

http://www.cs.cornell.edu/People/tj/publications/radlinski_etal_08b.pdf

Смещение

- Может возникать систематическое смещение результатов из-за учета дублей. На примере TDI:

$$A = \langle a, b, c \rangle \quad B = \langle e, a, f \rangle$$

$$I_{AA} = \langle a, e, b, f, \dots \rangle \quad I_{AB} = \langle a, e, f, b, \dots \rangle$$

$$I_{BA} = \langle e, a, b, f, \dots \rangle \quad I_{BB} = \langle e, a, f, b, \dots \rangle$$

- Для первых позиций смещений нет, но затем систематически сравниваем второй документ системы A с третьим документом системы B!

Подмешивания

- Документы бывают свежие (f) и обычные (w). Пусть, например:

$$A = \langle w, w, f \rangle \quad B = \langle w', f', w' \rangle$$

- Свежий документ из системы А будет вставать на 5-6 позиции, а свежий документ из системы В будет вставать на 3-4 позиции
- Если для свежего результата оптимальной с точки зрения кликов является, например, третья позиция, то выиграет система В, хотя в ней документ расположен субоптимально

Резюме в конце лекции

- Ранжирование — особый класс задач машинного обучения.
 - Критерий качества ранжирования зависит от приложения. Наилучшего универсального критерия не существует.
 - Три подхода: поточечный, попарный, списочный.
-
- *Tie-Yan Liu*. Learning to Rank for Information Retrieval. Springer-Verlag Berlin Heidelberg. 2011
 - *Hang Li*. A Short Introduction to Learning to Rank // IEICE Trans. Inf. & Syst., Vol.E94–D, No.10 October 2011.