«Машинное обучение»

Ранжирование

Александр Дьяконов

26 апреля 2022 года

План

Задача ранжирования (Learning to Rank)

Point-wise – поточечный подход
Pair-wise – попарный подход
List-wise – списочный подход

Ordinal Regression, RankSVM, RankNet, LambdaRank, RankBoost, LambdaMART

Оценка результатов поиска/рекомендаций

Задача ранжирования (Learning to Rank)

Есть обучающая выборка

$$\{x_i\}_{i=1}^m$$

На которой как-то задан порядок $\prec$

Построить алгоритм:

$$i \prec j \quad \Leftrightarrow \quad a(x_i) < a(x_j)$$

Например, $\{x_i = (q_i, d_i)\}_{i=1}^m$

формирование поисковой выдачи (поисковое ранжирование)
формирование ленты рекомендаций
упорядочивание вариантов ответа (ex: экспертной системы)

Подходы

Point-wise – поточечный

решаем как задачу регрессии

Pair-wise – попарный (считается лучше)

пытаемся правильно упорядочить пары объектов

List-wise – списком

пытаемся восстановить сразу весь порядок (список)

Поточечный подход: ранговая / порядковая регрессия (Ordinal Regression)

$$Y = \{1, 2, \ldots, l\}, \quad 1 \prec 2 \prec \ldots \prec l$$

введём пороги (отделяющие упорядоченные метки)

$$-\infty = \theta_0 < \theta_1 < \ldots < \theta_l = +\infty$$

классификация осуществляется так:

$$a_{w,\theta}(x) = y \quad \Leftrightarrow \quad \theta_{y-1} < b_{w,b}(x) \le \theta_y$$

J.D.M.Rennie, N.Srebro. «Loss functions for preference levels: regression with discrete ordered labels» // https://home.ttic.edu/~nati/Publications/RennieSrebroIJCAI05.pdf

Поточечный подход: ранговая / порядковая регрессия (Ordinal Regression)

Обучение по двум ближайшим порогам

$$\sum_{i=1}^{m} L(b_w(x)-\theta_{y_i-1}) + L(\theta_{y_i} - b_w(x)) \to \min_{w,\theta}$$

Обучение по всем порогам

$$\sum_{i=1}^{m}\sum_{j=1}^{l} L(\theta_j - b_w(x))\operatorname{sgn}(j-y_i) \to \min_{w,\theta}$$

Различные функции ошибки

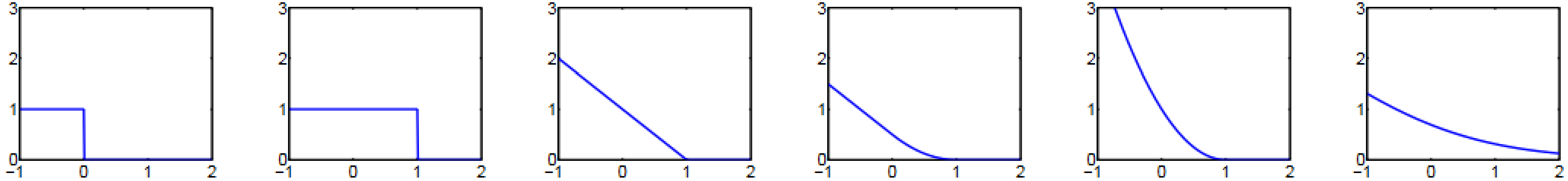


Figure 1: Different margin penalty functions $f(yz)$ (left to right): (1) sign agreement, (2) margin agreement, (3) hinge, (4) smooth hinge, (5) modified least squares, (6) logistic.

Попарный подход

$$i \prec j \quad \Leftrightarrow \quad a(x_i) < a(x_j)$$

Используем суррогатные функции

$$\sum_{i \prec j} I[a_w(x_i) \not< a_w(x_j)] = \sum_{i \prec j} I[a_w(x_i) \geq a_w(x_j)] \leq \sum_{i \prec j} L(a_w(x_j) - a_w(x_i)) \to \min$$

RankSVM $L(z) = \max[0, 1 - z]$

RankBoost $L(z) = \exp(-z)$

RankNet $L(z) = \log(1 + \exp(-z))$

RankSVM

$$\sum_{i \prec j} \max(0, 1 - (a_w(x_j) - a_w(x_i))) + \| w \|_2^2 \to \min_w$$

какой задаче квадратичного программирования соответствует:

$$\begin{cases} \frac{1}{2} \| w \|_2^2 + C \sum_{i \prec j} \xi_{ij} \to \min_{w,\xi} \\ w^{\mathrm{T}} (x_j - x_i) \geq 1 - \xi_{ij}, \; i \prec j \\ \xi_{ij} \geq 0, \; i \prec j \end{cases}$$

Ching-Pei Lee and Chih-Jen Lin «Large-scale Linear RankSVM»

https://www.csie.ntu.edu.tw/~cjlin/papers/ranksvm/ranksvml2.pdf

RankNet

$$\sum_{i\prec j}\log(1+\exp(-\sigma(w^{\mathrm{T}}x_j - w^{\mathrm{T}}x_i))) \to \min_w$$

по случайной упорядоченной паре делаем поправку:

$$w \leftarrow w - \eta\frac{\exp(-\sigma(w^{\mathrm{T}}x_j - w^{\mathrm{T}}x_i))}{1+\exp(-\sigma(w^{\mathrm{T}}x_j - w^{\mathrm{T}}x_i))}(-\sigma)(x_j - x_i)$$

$$w \leftarrow w + \eta\frac{\sigma}{1+\exp(+\sigma(w^{\mathrm{T}}x_j - w^{\mathrm{T}}x_i))}(x_j - x_i)$$

Christopher J.C. Burges From RankNet to LambdaRank to LambdaMART:
an overview // https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/MSR-TR-2010-82.pdf

Списочный подход: LambdaRank

Пусть есть какой-то функционал, который оценивает список Q

(в оригинале – NDCG)

При изменении $x_i \leftrightarrow x_j$ **он меняется на** ΔQ_{ij}

$$w \leftarrow w + \eta \frac{\sigma}{1 + \exp(+\sigma(w^{\mathrm{T}} x_j - w^{\mathrm{T}} x_i))} |\Delta Q_{ij}| (x_j - x_i)$$

– приближённая оптимизация Q

Списочный подход: LambdaRank

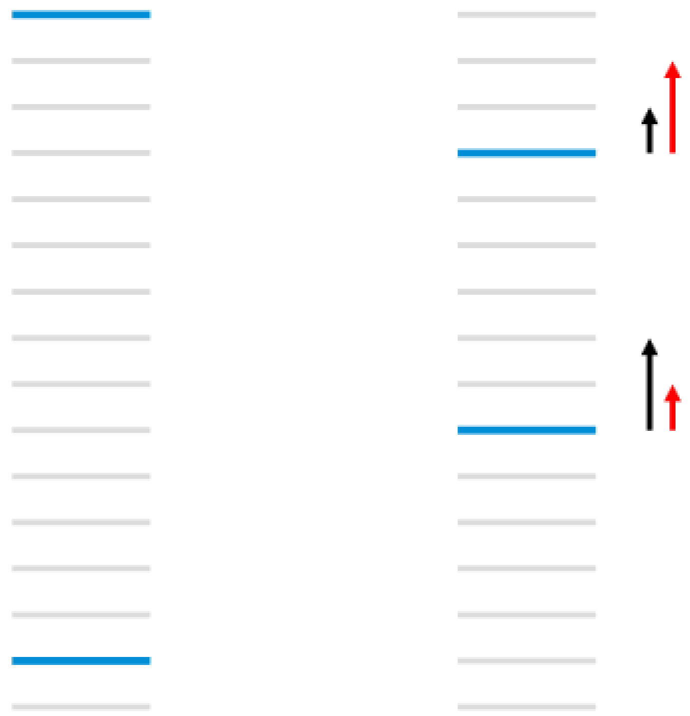


Fig. 1 A set of urls ordered for a given query using a binary relevance measure. The light gray bars represent urls that are not relevant to the query, while the dark blue bars represent urls that are relevant to the query. Left: the total number of pairwise errors is thirteen. Right: by moving the top url down three rank levels, and the bottom relevant url up five, the total number of pairwise errors has been reduced to eleven. However for IR measures like NDCG and ERR that emphasize the top few results, this is not what we want. The (black) arrows on the left denote the RankNet gradients (which increase with the number of pairwise errors), whereas what we'd really like are the (red) arrows on the right.

RankBoost

практически это AdaBoost, но на парах

RankBoost Algorithm

Initialize: initial distribution D over $X \times X$

Do for $t = 1, \dots, T$:

(1) Train **WeakLearn** using distribution D_t.

(2) **WeakLearn** returns a weak hypothesis h_t.

(3) Choose $\alpha_t \in R$

(4) Update weights: for $\forall (x_0, x_1)$:

$$D_{t+1}(x_0, x_1) = \frac{D_t(x_0, x_1) \exp(-\alpha_t (h_t(x_0) - h_t(x_1)))}{Z_t}$$

where Z_t is the normalization factor:

$$Z_t = \sum_{x_0, x_1} D_t(d_0, d_1) \exp(-\alpha_t (h_t(x_0) - h_t(x_1))).$$

Output: the final hypothesis: $H(x) = \sum_{t=1}^{T} \alpha_t h_t$

LambdaMART = LambdaRank + MART (Multiple Additive Regression Trees)

Algorithm: LambdaMART

set number of trees N, number of training samples m, number of leaves per tree L, learning rate η

for $i = 0$ to m **do**

  $F_0(x_i) = \text{BaseModel}(x_i)$ //If BaseModel is empty, set $F_0(x_i) = 0$

end for

for $k = 1$ to N **do**

  for $i = 0$ to m **do**

    $y_i = \lambda_i$

    $w_i = \frac{\partial y_i}{\partial F_{k-1}(x_i)}$

  end for

  $\{R_{lk}\}_{l=1}^{L}$ // Create L leaf tree on $\{x_i, y_i\}_{i=1}^{m}$

  $\gamma_{lk} = \frac{\sum_{x_i \in R_{lk}} y_i}{\sum_{x_i \in R_{lk}} w_i}$ // Assign leaf values based on Newton step.

  $F_k(x_i) = F_{k-1}(x_i) + \eta \sum_l \gamma_{lk} I(x_i \in R_{lk})$ // Take step with learning rate η.

end for

Пример: оценка релевантности запроса

Обучение это тройки «запрос, документ, метка»

$$(q, d, y)$$

$$x_i = (q_i, d_i)$$

Можно сделать генерацию признаков:

$$x_i = (q_i, d_i) \to (f_1(q_i), f_2(d_i), f_3(q_i, d_i))$$

- число слов в запросе
- число слов в документе
- число общих слов в запросе и документе
- представление (эмбединг) запроса
- категория документа
- популярность запроса

Пример: оценка релевантности запроса

Может быть целая нейросеть

$$D:\text{text}\to\mathbb{R}^k$$

предварительно можно перейти к триграммам

$a(q_i, d_i) = \cos(D(q_i), D(d_i))$ **– схожесть запросов**

Можно учить:
похожие (если был клик по документу при запросе) близко,
непохожие далеко.

примерно так работает DSSM

Ранжирование на практике

0) запрос

1) отбор кандидатов (~ 10^3)

2) ранжирование (GBM)

3) переранжирование и коррекция
ex: не показываем товары из одного каталога

4) выдача

Оценка результатов поиска/рекомендаций (другая тема)

Задача с бинарной релевантностью

$$x_1 \prec x_2 \prec \ldots \prec x_m$$

$y_i = 1$ **– релевантный объект**

$y_i = 0$ **– нерелевантный объект**

Задача ранжирования

Целевой признак может быть бинарным, но это не задача классификации

Precision at n

Точность на первых n элементах

$$p@n = \frac{y_1 + \ldots + y_n}{n}$$

Average Precision at n

Средняя точность на первых n элементах

$$ap@n = \sum_{k=1}^{n} \frac{P(k)}{\min(n,r)}$$

r **– мощность множества релевантных объектов (товаров, документов)**

n **– сколько рекомендаций будет учитываться**

$$P(k) = \begin{cases} p@k, & y_k = 1, \\ 0, & y_k = 0, \end{cases}$$

y_i **– бинарное значение релевантности**

Mean Average Precision

– усреднение ap@n по всем пользователям

Average Precision at n

Примеры (три релевантных объекта, $r = 3$):

$0 \succ 0 \succ 0$

$$ap@3 = \frac{1}{3}[0+0+0]$$

$0 \succ 0 \succ 1$

$$ap@3 = \frac{1}{3}\left[0+0+\frac{1}{3}\right]$$

$0 \succ 1 \succ 1$

$$ap@3 = \frac{1}{3}\left[0+\frac{1}{2}+\frac{2}{3}\right]$$

$1 \succ 0 \succ 0$

$$ap@3 = \frac{1}{3}\left[\frac{1}{1}+0+0\right]$$

$0 \succ 0 \succ 1 \succ 1 \succ 1$

$$ap@5 = \frac{1}{3}\left[0+0+\frac{1}{3}+\frac{2}{4}+\frac{3}{5}\right]$$

$1 \succ 1 \succ 1 \succ 0 \succ 0$

$$ap@5 = \frac{1}{3}\left[\frac{1}{1}+\frac{2}{2}+\frac{3}{3}+0+0\right]$$

Concordant – Discordant ratio

$$\frac{|\{(i,j)\,|\,y_i > y_j, 1 \le i < j \le m\}|}{|\{i\,|\,y_i = 1\}| \cdot |\{j\,|\,y_j = 0\}|}$$

Упорядочили: E, D, C, B, A (по убыванию релевантности)

На самом деле: B, E – релевантные

Пары «релевантный» – «нерелевантный»:

BA	EA
BC	EC
BD	ED

Качество упорядочивания: 4 / (2 + 4)

~ AUC ROC

Что ещё может встретиться… в задачах рекомендации

$$\frac{1}{|Z|}\sum_{z\in Z}\frac{|\{x_1,\ldots,x_z\}\cap\{x'_1,\ldots,x'_z\}|}{z}$$

$x_1,\ldots,x_n$ – упорядоченный список ответов

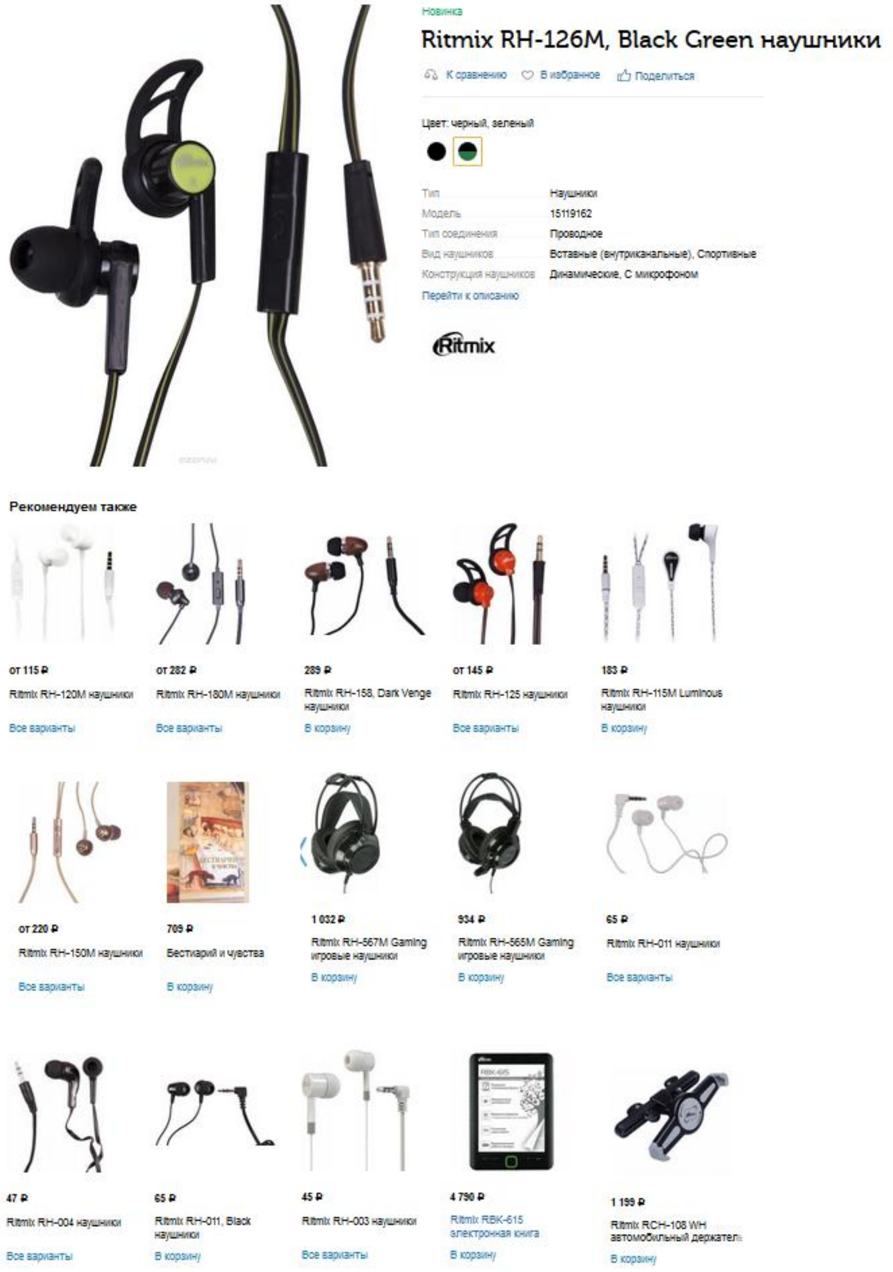
$x'_1,\ldots,x'_m$ – все релевантные

$$Z\subseteq\{1,2,\ldots,n\}$$

$$Z=\{5,10,15,20,25,30\}$$

когда логично применить?

Рекомендации



Mean Reciprocal Rank (MRR)

– это усреднение Reciprocal rank (RR) по всем ранжированиям, который сделал алгоритм.

$$RR = \frac{1}{\min\{i : y_i = 1\}}$$

Часто оптимизируют именно его!

Классические функционалы в поиске

Случай небинарной релевантности

Выдали id документов/товаров/…, а их ценность (релевантность):

$$y_1, \ldots, y_m$$

Cumulative Gain at n

$$\mathrm{CG@}n = y_1 + \ldots + y_n$$

Discounted Cumulative Gain at n

$$\mathrm{DCG@}n = \sum_{i=1}^{n} \frac{2^{y_i} - 1}{\log_2(i+1)}$$

Ещё вариант:

$$DCG@n = y_1 + \sum_{i=2}^{n} \frac{y_i}{\log_2(i)} = y_1 + y_2 + \frac{y_3}{\log_2 3} + \ldots + \frac{y_n}{\log_2 n}$$

Цена ошибок за неправильное ранжирование

$$\frac{1}{\log_2(1+1)} - \frac{1}{\log_2(1+2)} \approx 0.37$$

$$\frac{1}{\log_2(1+10)} - \frac{1}{\log_2(1+11)} \approx 0.01$$

$$\frac{1}{\log_2(1+10)} - \frac{1}{\log_2(1+20)} \approx 0.06$$

Normalized DCG

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

IDCG = ideal DCG

для того, чтобы не было зависимости от длины выдачи

Ещё подход к сравнению порядков:

Пусть алгоритм выдал

$$x_1 \prec x_2 \prec \ldots \prec x_m$$

Правильный порядок

$$x_{i_1} \prec x_{i_2} \prec \ldots \prec x_{i_m}$$

Надо сравнить:

$$(1, 2, \ldots, m)$$

$$(i_1, i_2, \ldots, i_m)$$

Ранговые корреляции…

Ещё подход к оценке ранжирования

Известны вероятности того, что объект является релевантным

$$p_i = p(x_i)$$

например, $p_i = \dfrac{2^{y_i} - 1}{\max(2^{y_\bullet})}$

~ пользователь выберет ссылку

Expected reciprocal rank (ERR)

$$ERR@n = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{k}p_k\prod_{i<k}(1-p_i)$$

Как интерпретировать?

Итоги

A Short Introduction to Learning to Rank

http://times.cs.uiuc.edu/course/598f14/l2r.pdf

Christopher J.C. Burges From RankNet to LambdaRank to LambdaMART: an overview // https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/MSR-TR-2010-82.pdf

https://paperswithcode.com/task/learning-to-rank