Ранжирование
(learning to rank)

$\mathbb{Y} = \mathbb{R}$

$\mathbb{Y} = \{1, \dots K\}$

$X = \{x_1, \dots x_\ell\} \subset \mathbb{X}$

$(i,j) \in R \subset \{1, \dots \ell\}^2 \Rightarrow a(x_i) > a(x_j)$

$\boxed{(i,j),\ (j,k),\ (k,i)}$ – могут быть

R – вместо целевых переменных
поисковое ранжирование:

$x = (q, d)$

в R – пары $x_i = (q_i, d_i)$ и $x_j = (q_j, d_j)$, где $q_i = q_j$

## Метрики качества ранжирования

Наивный подход:

1) назначить для каждого $x \in X$ $y \in \mathbb{R}$:

$(i,j) \in R \Rightarrow y_i > y_j$

2) обучаем $a(x)$ на $(x_i, y_i)$

метрика – MSE

контрпример: $\boxed{x_1 > x_2 > x_3}$

|  |  |  |  |
|---|---|---|---|
| y: | 3 | 2 | 1 |
| a: | 0 | -0.05 | -10 |

MSE >> 0, хотя ранжирование идеальное

Более правильные метрики:

AUC-ROC, DCG, MAP

mean
average
precicion

например:

$g(y) = 2^y - 1$

$$DCG@k(q) = \sum_{i=1} g(\underbrace{y_{(i)}}_{\text{правильный ответ документа с } i\text{-й позиции}}) \cdot d(i) \qquad d(i) = \frac{1}{\log(i+1)}$$

$$DCG@k = \frac{1}{|Q|} \sum_{q \in Q} DCG@k(q)$$

nDCG@k - нормируем на DCG для идеального ранжирования

## Каскадные метрики

ERR

$\boxed{\text{pFound}}$

$p_i$ - вероятность дойти до i-й позиции в выдаче

$p_1 = 1$

$$p_{i+1} = p_i(1 - \underbrace{y_{(i)}}_{\text{вероятность того, что польз. найдет ответ в i-м документе}})(1 - \underbrace{p_{out}}_{\text{вер-ть того, что пользователь задолбается}})$$

$$pFound@k(q) = \sum_{i=1}^{k} p_i y_{(i)}$$

$$pFound@k = \frac{1}{|Q|} \sum_{q \in Q} pFound@k(q)$$

---

Признаки в ранжировании

- запросные
- статические (по документам)
- динамические (по запросу и документу)

(1) BM25

$$BM25(q,d) = \sum_{i=1}^{n} IDF(q_i) \frac{tf(q_i,d)(k_1+1)}{tf(q_i,d) + k_1 \cdot (1-b+b

$i=1$

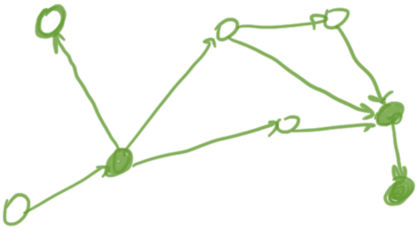
сумма по словам запроса

i-е слово запроса

средн. размер докум.

$k_1, b$ - гиперпараметры

② Page Rank



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

документы, ссылающиеся на d

документы, на которые ссылается c

СЛАУ

$$R = \frac{1-\delta}{|D|} + \delta \cdot A \cdot R$$

„матрица смежности"

$$R = (I - \delta R)^{-1} \frac{1-\delta}{|D|} \cdot \vec{1}$$

инициализ. R, пересчитываем до сходимости

---

Методы ранжирования

- pointwise
- pairwise
- listwise

① Pointwise

Тот самый наивный подход

$y_i$ - показатели релевантности

$$\frac{1}{l} \sum_{i=1}^{l} L(y_i, a(x_i)) \to \min_a$$

## ② Pairwise

$$\underbrace{\sum_{(i,j)\in R} [a(x_i) - a(x_j) < 0]}_{\text{число дефектных пар}} \to \min_a \quad \boxed{\le}$$

$(i,j) \in R \Rightarrow a(x_i) > a(x_j)$

$$[z<0] \le \tilde{L}(z)$$

$$\boxed{\le} \sum_{(i,j)\in R} \tilde{L}(a(x_i) - a(x_j)) \to \min_a$$

если $a$ - дифференц. модель, то можно обучать SGD, семплируя пары из $R$

1) $\begin{pmatrix} y_i = 100 \\ y_j = 0 \end{pmatrix} = \begin{pmatrix} y_i = 100 \\ y_j = 99 \end{pmatrix}$

чтобы все пары не были равнозначны, можно семплировать их из $R$ с вероятностями, пропорциональными $|y_i - y_j|$

2) можно сделать модель попарной $a(x_i, x_j)$

---

## RankNet

$a(x) = \langle w, x \rangle$

$\tilde{L}(z) = \log(1 + \exp(-\underset{\text{константа}}{\sigma} z))$

SGD:

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

Эмпирическое наблюдение:

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

$|\Delta F_{ij}|$ - изменение метрики ранжирования (например, nDCG) при обмене местами $x_i$ и $x_j$

в итоге будет примерно оптимизироваться метрика ранжирования

Lambda Rank