

# Машинное обучение

## Ранжирование



# Содержание лекции

- Постановка задачи
- Примеры применения
- Оценки качества
- Подходы к решению задачи
  - поточечный
  - попарный
  - списочный

# Постановка задачи

$X$  — 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  $x$

Часто на практике объекты разделяются на группы (list, списки), и их нужно ранжировать в пределах одной группы. При этом число групп велико.

# Пример 1. Ранжирование результатов 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 асессорами:

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  $q$ :

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

## Пример 2. Рекомендательные системы

$U$  — 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 выступать  $y(u', i)$  — рейтинги, поставленные другими пользователями  $u'$

# Оценки качества

- AUC
- Точность (precision)

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

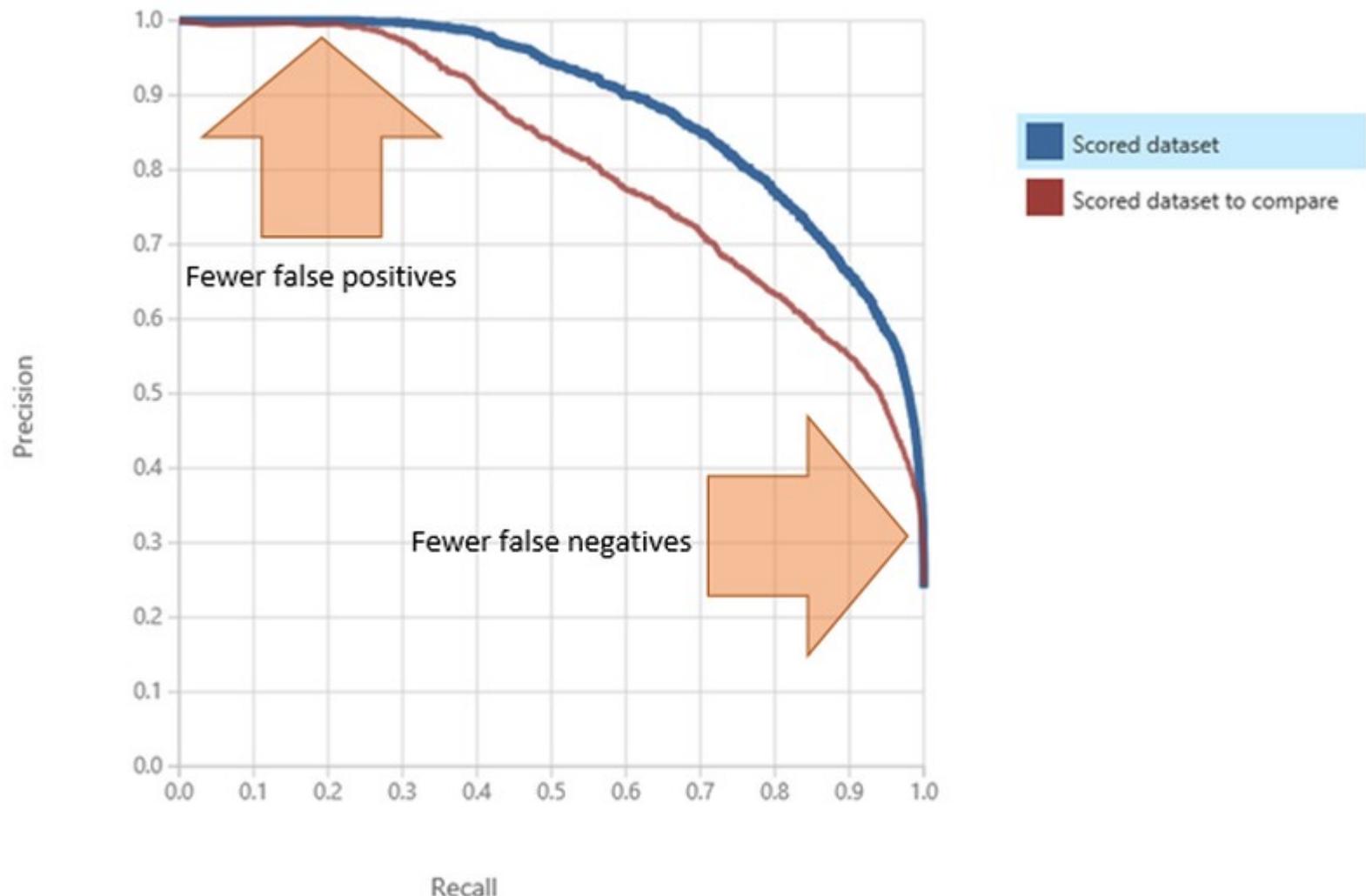
- Полнота (recall, TPR)

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

- Выпадение (fall-out, FPR) - вероятность нахождения нерелевантного ресурса

$$\text{fall-out} = \frac{|\{\text{non-relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{non-relevant documents}\}|}$$

# PR-кривая



Средняя точность (AveP) – площадь под PR-кривой.

Как выглядит график для случайного порядка?

# Оценки качества

- F-мера (F-measure, мера Ван Ризбергена)

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})}$$

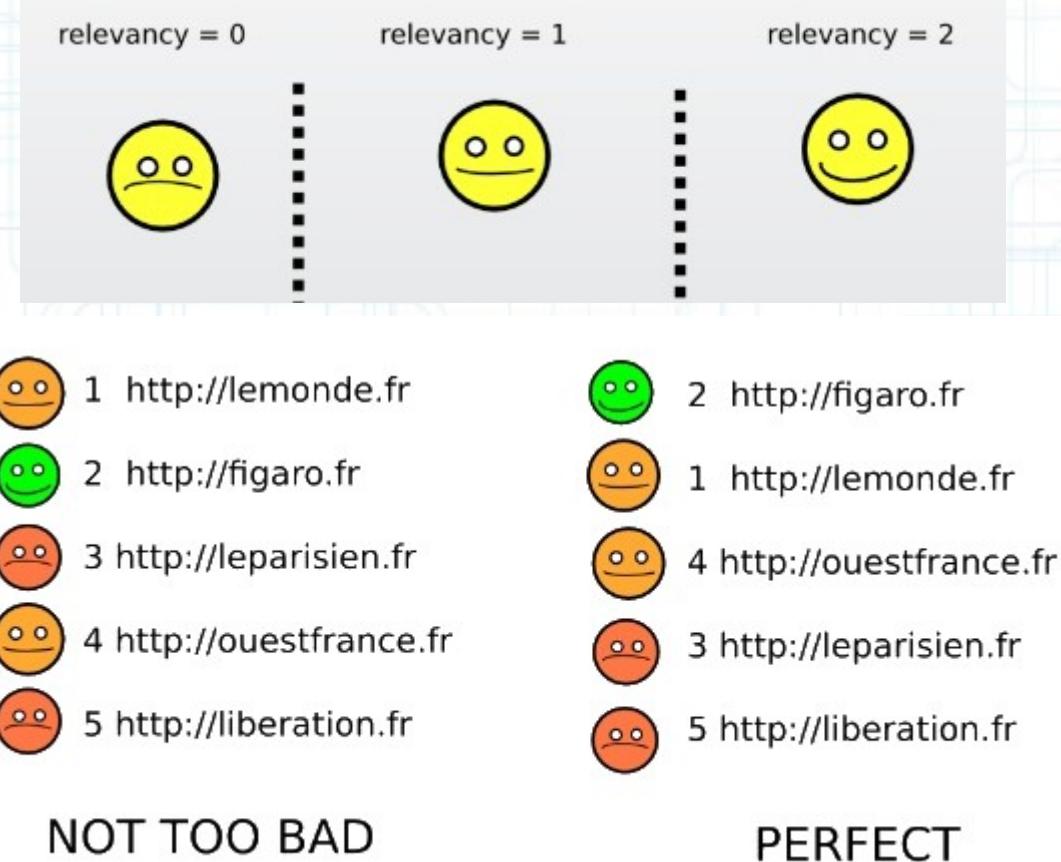
- CG (Cumulative gain)  $\text{CG}_p = \sum_{i=1}^p \text{rel}_i$
- DCG (Discounted cumulative gain)

$$\text{DCG}_p = \sum_{i=1}^p \frac{2^{\text{rel}_i} - 1}{\log_2(i + 1)}$$

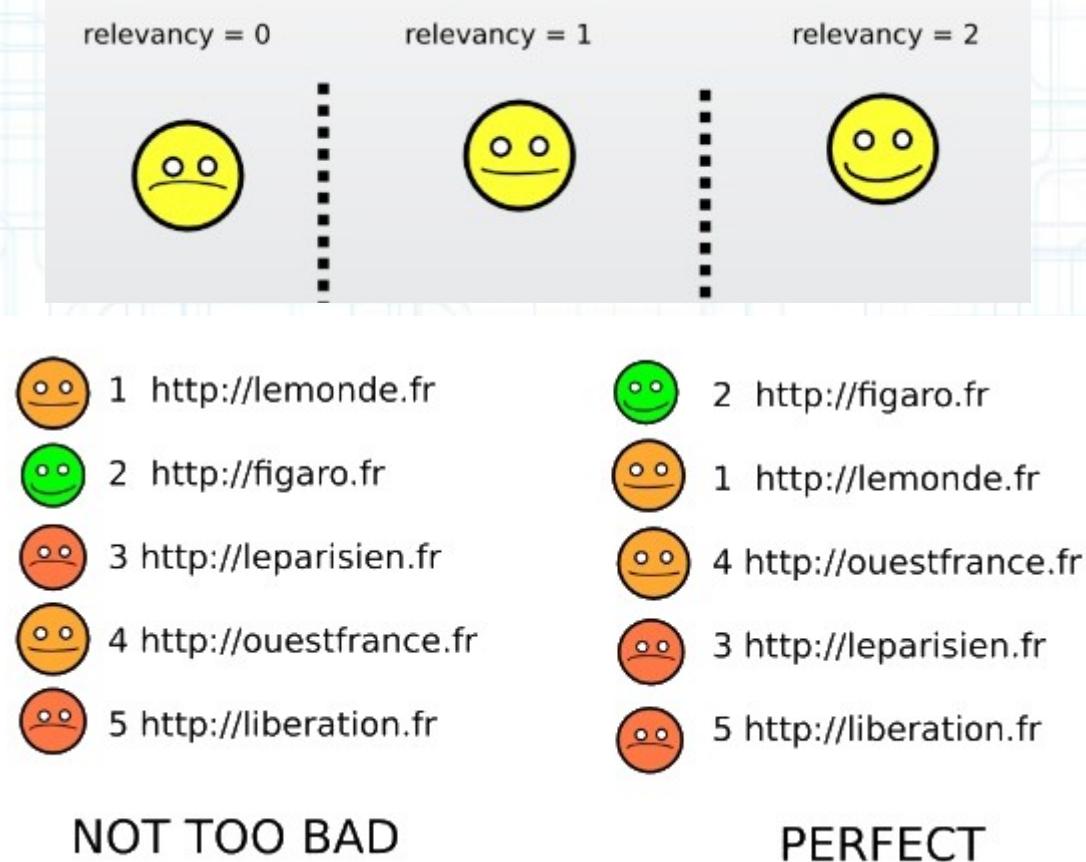
- Нормированный DCG (NDCG)

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

# Пример вычисления nDCG



# Пример вычисления nDCG



$$\text{DCG} = 1/\log(2) + 3/\log(3) + 0 + 1/\log(5) + 0$$

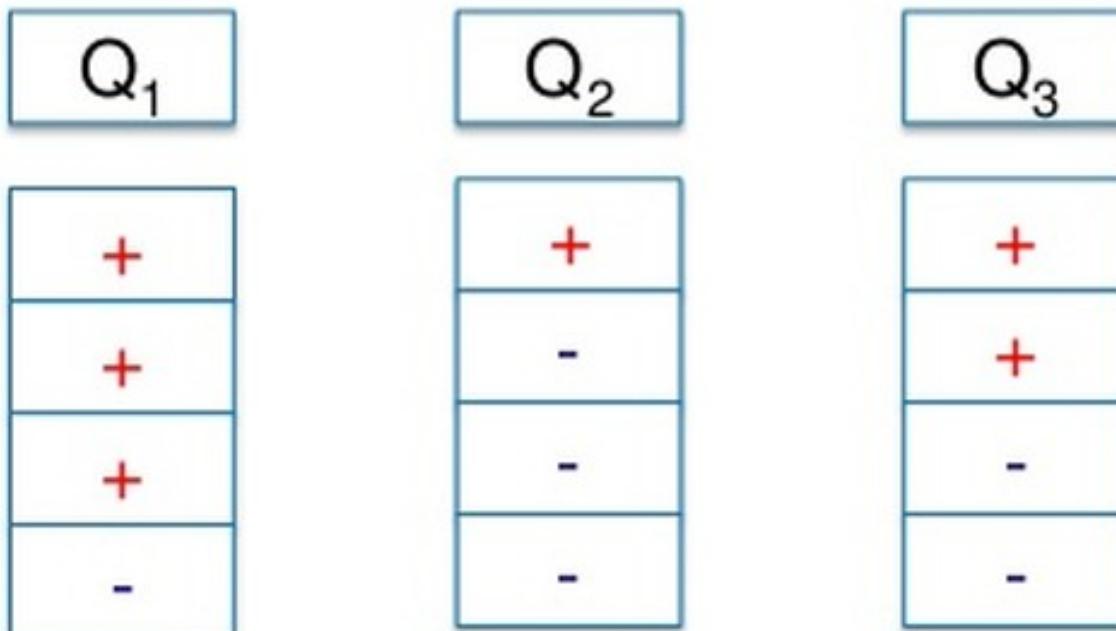
$$\text{IDCG} = 3/\log(2) + 1/\log(3) + 1/\log(4) + 0 + 0$$

# Подходы к решению задачи ранжирования

- Point-wise — поточечный: предсказывается ранг объекта
- Pair-wise — попарный: моделируется функция, ранжирующая пары объектов
- List-wise — списочный: объект — упорядоченный набор; оптимизируются параметры ранжирующей списки функции

# Point-wise

- Предположение: для обучающей выборки известны абсолютные значения ранга
- Сведем задачу ранжирования к задаче предсказания ранга (классификации или регрессии)
- Пример: Объект –  $(q, d)$ ; два класса: документ  $d$  релевантен запросу  $q$  или нет. Обучающую выборку должны готовить специально обученные ассесоры.



# Недостатки поточечного подхода

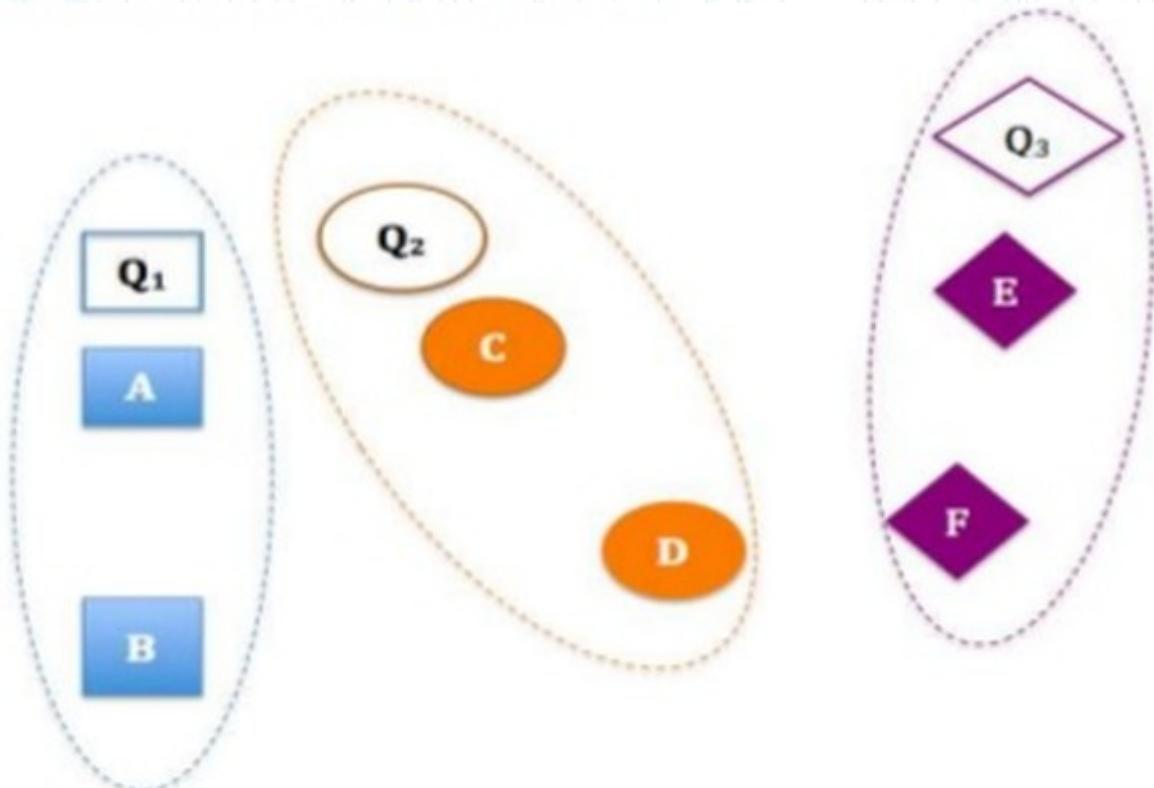
- Проблема: алгоритм рассматривает документы из разных запросов вместе, сравнивая между собой

Если один человек предпочитает классику, а другой – рок. Зачем определять силу их предпочтения?

- Любой (правильный и неправильный) порядок объектов с приближенно равными рангами штрафуется функционалом качества одинаково. Так как штраф зависит от величины ранга, а не от порядка

# Pair-wise

- Объект – пара ранжируемых элементов ( $x_1, x_2$ ). Требуется предсказать порядок:  $x_1 > x_2$  или  $x_1 < x_2$ .
- Обучающая выборка: множество известных отранжированных пар



# Pair-wise

- Метод обучения учится на парах.  
Например, SVM:

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

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,

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

# Pair-wise

- Пример 2: метод стохастического градиента для логистической регрессии (RankNet)

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

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

На каждой итерации берем случайно группу и пару  $i < j$ :

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

# Недостатки попарного подхода

- Оптимизируемый функционал качества оценивает глобальный порядок, а не порядок для одной группы (list)
- Не учитываются зависимости между сравниваемыми парами в общей группе

# Пример проявления недостатка

- В магазине 3 товара: a,b,c. Посетители сайта магазина ранжируют товары по убыванию предпочтений (обучающая выборка).
- Pair-wise подход посчитал вероятности:  $P(a>b) = 0.6$ ;  $P(a>c) = 0.3$ ;  $P(b>c) = 0.7$
- Вычислим вероятности всех возможных порядков

$P(>)$	a	b	c
a		0.6	0.3
b	0.4		0.7
c	0.7	0.3	

# Пример проявления недостатка

- $P(a>b>c) = P(a>b)*P(a>c)*P(b>c) = 0.126$
- $P(a>c>b) = 0.3*0.6*0.3 = 0.054$
- $P(b>a>c) = 0.4*0.7*0.3 = 0.084$
- $P(b>c>a) = 0.7*0.4*0.7 = 0.196$
- $P(c>a>b) = 0.7*0.3*0.6 = 0.126$
- $P(c>b>a) = 0.3*0.7*0.4 = 0.084$
- Все правильно?

$P(>)$	a	b	c
a		0.6	0.3
b	0.4		0.7
c	0.7	0.3	

# Пример проявления недостатка

- $P(a>b>c) = P(a>b)*P(a>c)*P(b>c) = 0.126$
- $P(a>c>b) = 0.3*0.6*0.3 = 0.054$
- $P(b>a>c) = 0.4*0.7*0.3 = 0.084$
- $P(b>c>a) = 0.7*0.4*0.7 = 0.196$
- $P(c>a>b) = 0.7*0.3*0.6 = 0.126$
- $P(c>b>a) = 0.3*0.7*0.4 = 0.084$
- Сумма всех вероятностей = 0.67
- Парадокс Кондорсе'. Метод Шульце?
- $P(a>b, b>c, c>a) = 0.294$

$P(>)$	a	b	c
a		0.6	0.3
b	0.4		0.7
c	0.7	0.3	

# Пример проявления недостатка

- $P(a>b>c) = P(a>b)*P(a>c)*P(b>c) = 0.126$
- $P(a>c>b) = 0.3*0.6*0.3 = 0.054$
- $P(b>a>c) = 0.4*0.7*0.3 = 0.084$
- $P(b>c>a) = 0.7*0.4*0.7 = 0.196$
- $P(c>a>b) = 0.7*0.3*0.6 = 0.126$
- $P(c>b>a) = 0.3*0.7*0.4 = 0.084$
- А на самом деле в обучающей выборке все пользователи сайта делились на три группы: 30% голосовало за порядок  $a>b>c$ , 30% - за  $c>a>b$ , 40% - за  $b>c>a$   
Других вариантов пользователи не предлагали!

$P(>)$	a	b	c
a		0.6	0.3
b	0.4		0.7
c	0.7	0.3	

# List-wise

- Объект – группа (list), внутри которой нужно произвести ранжирование
- Оптимизируемый функционал оценивает качество каждой группы
- Было в методе стохастического градиента для Pair-wise:

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

- Модифицируем:

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