

Ранжирование. Многозначная классификация. Рекомендательные системы



Содержание лекции

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

Постановка задачи ранжирования

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 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 :

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

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

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

Оценки качества

- Precision@k – точность среди первых k рекомендаций (не учитывает позиции правильных в топе, <1 если кол-во релевантных во всей выборке <k)
- Average Precision@k, AP@k – средняя P@i для всех позиций i **релевантных** рекомендаций
- MAP@k – Mean Average Precision@k – добавляется усреднение по всем поисковым выдачам/пользователям

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)

$$CG_p = \sum_{i=1}^p rel_i$$

- DCG (Discounted cumulative gain)

$$DCG_p = \sum_{i=1}^p \frac{2^{rel_i} - 1}{\log_2(i + 1)}$$

- Нормированный DCG (NDCG)

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

Пример вычисления nDCG



-  1 <http://lemonde.fr>
-  2 <http://figaro.fr>
-  3 <http://leparisien.fr>
-  4 <http://ouestfrance.fr>
-  5 <http://liberation.fr>

NOT TOO BAD

-  2 <http://figaro.fr>
-  1 <http://lemonde.fr>
-  4 <http://ouestfrance.fr>
-  3 <http://leparisien.fr>
-  5 <http://liberation.fr>

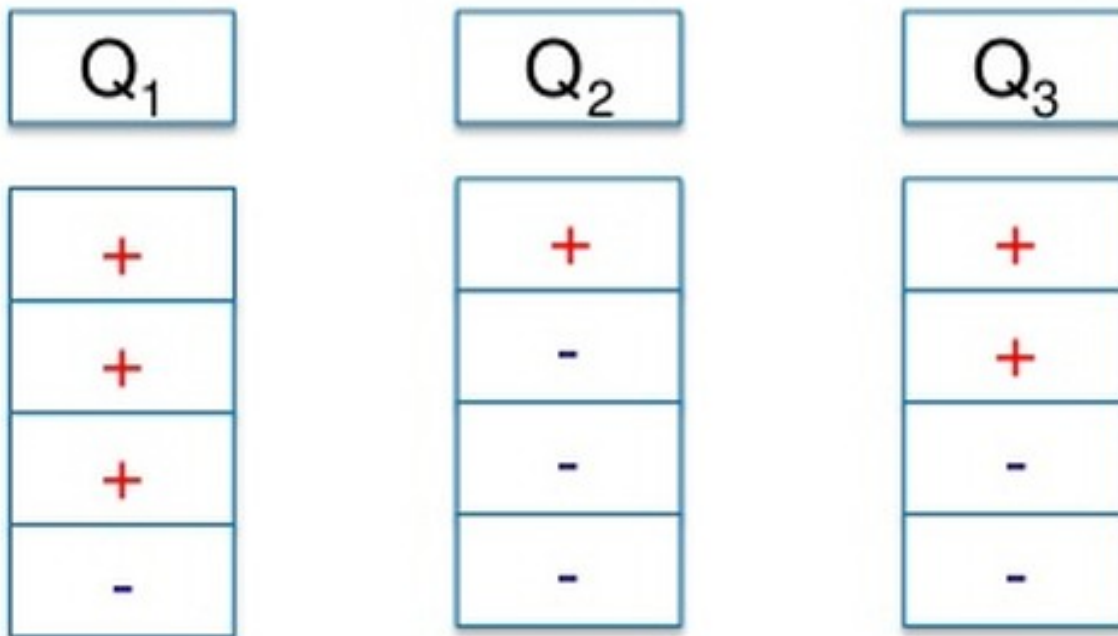
PERFECT

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 функции для максимизации некоторой меры порядка

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.

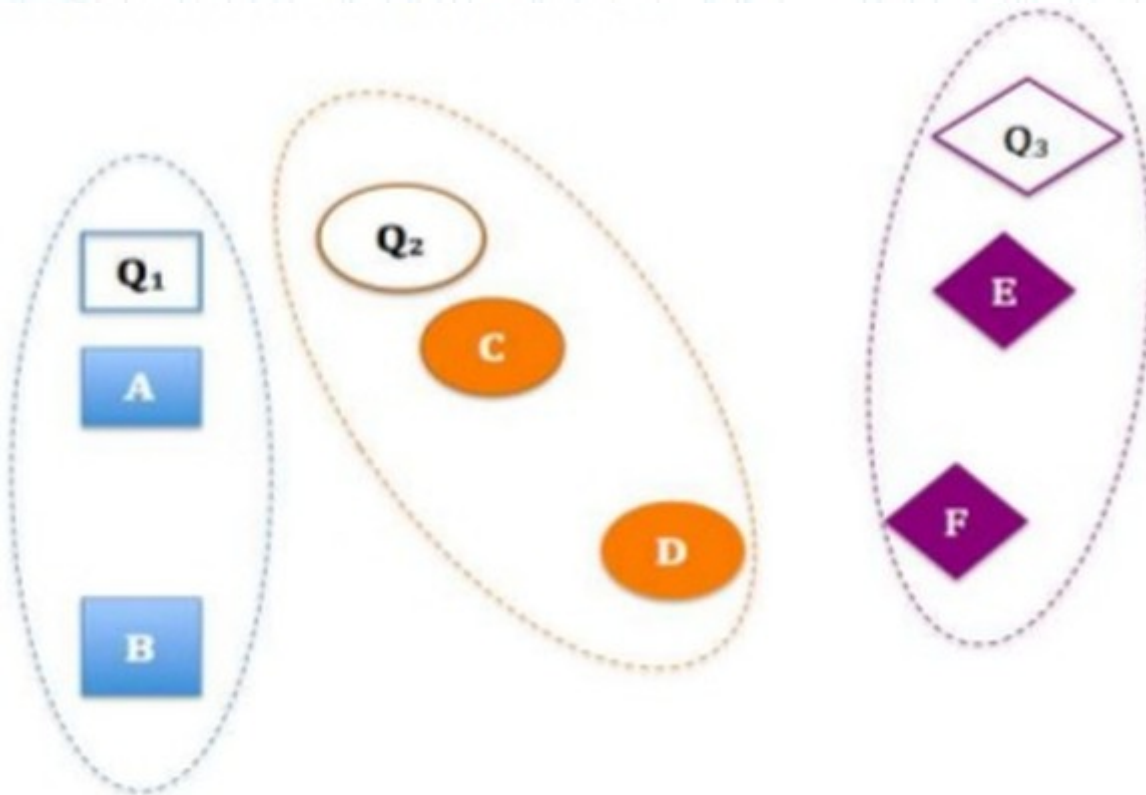


Недостатки поточечного подхода

- Проблема: алгоритм рассматривает документы из разных запросов вместе, сравнивая между собой. Если один человек предпочитает классику, а другой – рок. Зачем определять силу их предпочтения? Как можно уравнивать факт предпочтения песни меломаном и случайным слушателем?
- Любой (правильный и неправильный) порядок объектов внутри одного списка с приблизительно равными рангами штрафуются функционалом качества одинаково:
.....; (q1,d1); (q1,d2); (q1,d3);.....
Если пар очень много, то от порядка указанных мера почти не зависит т.к. они идут рядом в гигантском списке

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:

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

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,

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

Pair-wise

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

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

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

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:

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

Extreme Multilabel Classification

- Трактует группу (list) в качестве класса
- Один объект может входить в несколько групп => многозначная классификация
- Иерархия классов:

