Learning to rank

Машинное Обучение, 20!7

По материалам К.В. Воронцова

Малютин Евгений Алексеевич

Определения и обозначения

- X – множество объектов
- $X^l = (x_1 ... x_l)$, – выборка
- $i \prec j$ – правильный порядок

Задача:
Построить ранжирующую функцию a :

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

Линейная модель ранжирования:

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

Пример 1. Задача ранжирования поисковой выдачи

- D – коллекция текстовых документов (documents)
- Q – множество запросов (queries)
- $D_q \in D$ – множество документов, найденных по запросу q
- $X = Q \times D$ – объектами являются пары «запрос, документ»:

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

- Y – упорядоченное множество рейтингов
- $y : X \to Y$ – 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 1. 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

TF-IDF – term frequency - inverse document frequency

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;
$P(q,d) = \Pi_{w \in q}(N_w/N), ndw$ – оценка вероятности встретить в документе d слова запроса $q = (w_1, ..., w_k)$ чисто случайно;
Оценка релевантности запроса q документу d:

$$-logP(q,d) = \sum_{w \in q} n_{dw} log(N/N_w) \rightarrow max$$

$TF(w,d) = n_{dw} \rightarrow$ term frequency;
$IDF(w) = log(N/N_w)$ – inverted document frequency

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:

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

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'
То есть, поиск коллаборации $\Leftrightarrow$ отбор признаков

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:

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

Mean Average Precision:

$$MAP = \frac{1}{|Q|} \sum_q AP(x)$$

Доля дефектных пар

Пусть $Y \subseteq R$, y(d,q) – релевантность
$a(d, q)$ – искомая функция ранжирования.
$d_q^{(}i)$ – i-й документ по убыванию a(d,q)
Доля инверсий порядка среди первых n документов:

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

Связь с коэффициентом ранговой корреляции (τ Кенделла):

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

Связь с AUC:

$$AUC_n(q) = \frac{n(n-1)}{2l_- l_+} DP_n(q)$$

DCG – distributed cummulative gain

Пусть $Y \subseteq R$, y(d,q) – релевантность
$a(d, q)$ – искомая функция ранжирования.
$d_q^(i)$ – i-й документ по убыванию $a(d, q)$
Дисконтированная (взвешенная) сумма выигрышей:

$$DCG_n(q) = \sum_i^n G_q(d_q^{(i)}) D(i)$$

$G_q(d_q^{(i)}) = (2^{y(d,q)} - 1)$ – больший вес релевантным документам
$D(i) = 1/log_2(i + 1)$ – больший вес в начале выдачи
Нормированная дисконтированная сумма выигрышей:

$$NDCG = DCG_n(q)/maxDCG_n(q)$$

Методы ранжирований

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

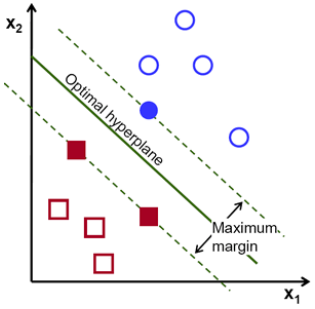
Переход к гладкому функционалу качества ранжирования:

$$Q(a) = \sum_{i \prec j} [a(x_j) - a(x_i) < 0] \leq \sum_{i \prec j} L(a(x_j) - a(x_i))) \to min$$

где a(x) — алгоритм ранжирования;
Тогда:

- $L(M) = (1 - M)+$ – RankSVM
- $L(M) = exp(-M)$ – RankBoost
- $L(M) = log(1 + e^{-M})$ – RankNet

SVM(что это?)



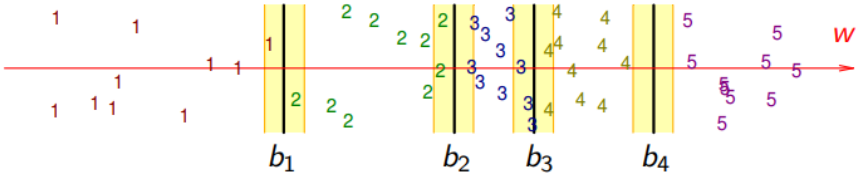
Возьмём две точки x_+, x_- на границе, тогда ширина разделяющей полосы:

$$\langle (x_+ - x_-), \frac{w}{||w||^2} \rangle = \frac{\langle w, x_+ \rangle - \langle w, x_- \rangle}{||w||} = \frac{(w_0 + 1) - (w_0 - 1)}{||w||} = \frac{2}{||w||}$$

Ранговая классификация – OC SVM

Пусть $Y = (1, ..., K)$, функция ранжирования линейная с порогами $b_0 = -\infty, b1, ..., b_{K-1} \in R, b_K = +\infty$
$a(x) = y$, если $b_{y-1} < \langle h_w, xi \rangle \leq b_y$

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



Постановка задачи SVM для ранговой классификации:

$$\begin{cases} \frac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{\ell}[y_i \neq K]\,(\xi_i + \xi_i^*) \to \min\limits_{w,b,\xi}; \\ b_{y_i-1} + 1 - \xi_i^* \leqslant \langle w, x_i \rangle \leqslant b_{y_i} - 1 + \xi_i; \\ \xi_i^* \geqslant 0, \quad \xi_i \geqslant 0. \end{cases}$$

Ranking SVM

Постановка задачи SVM для попарного подхода:

$$Q(a) = \frac{1}{2} C \sum_{i \prec j} L(x(x_j) - L(x_i)) \to \min_a$$

где $a(x) = \langle h_w, x_i \rangle$ – функция ранжирования,
$L(M) = (1 - M)_+$ – функция потерь
$M = Margin(i, j)\langle w, x_j - x_i \rangle$ – отступ
Постановка задачи квадратичного программирования:

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

RankNet

RankNet: гладкий функционал качества ранжирования:

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

при $L(M) = log(1 - e^{-\sigma M})$ и линейной модели $a(x) = \langle w, x \rangle$
SGD:
выбираем на каждой итерации $q, i \prec j$ случайно:

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

От RankNet до LambdaRank

SGD:

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

Оказывается, для оптимизации негладких функционалов $MAP, NDCG, pFound$ достаточно...

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

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

Ура

Резюме

- Ранжирование – особый класс задач машинного обучения.
- Критерий качества ранжирования зависит от приложения. Наилучшего универсального критерия не существует.
- Три подхода: поточечный, попарный, списочный. Теоретически списочный должен быть наилучшим. На практике всякое бывает

Что дальше?

- LambdaMART – бустинг над решающими деревьями от Microsoft.
- CatBoost – то же самое, но от Яндекса и на 2-м питоне