

Лекция 25

Обучение ранжированию

Е. А. Соколов
ФКН ВШЭ

17 июня 2017 г.

1 Методы ранжирования

На предыдущей лекции мы договорились, что ответы задаются не в виде пар, а в виде чисел y_1, \dots, y_ℓ . При этом считается, что если $y_i < y_j$, то для модели должно быть выполнено $a(x_i) < a(x_j)$.

Также мы обсудили ряд метрик качества ранжирования — среди них MAP, nDCG и pFound. Все они непосредственно учитывают порядок документов, и поэтому зависят от позиций, на которые их поместила модель. Поскольку зависимость позиций документов от параметров модели является кусочно-постоянной (позиции ведь принимают натуральные значения), непосредственно обучаться с данными метриками достаточно сложно. Поэтому многие методы ранжирования пытаются оптимизировать другие метрики, которые являются дифференцируемыми и при этом как-то оценивают качество ранжирования.

§1.1 Поточечные методы

В поточечном (pointwise) подходе предлагается забыть про то, что мы решаем задачу ранжирования, и независимо для каждого объекта x_i предсказывать ответ y_i . В зависимости от типа ответов получим задачу классификации или регрессии.

Если модель $a(x)$, которая получится в результате такого обучения, будет идеально восстанавливать целевую переменную, то и метрика качества ранжирования будет оптимизирована. Если же добиться идеального восстановления целевой переменной нельзя, то могут возникнуть проблемы — ведь поточечная метрика никак не учитывает порядок. С её точки зрения необходимо как можно точнее восстановить ответы, тогда как с точки зрения метрики ранжирования можно пожертвовать точностью предсказания ради корректности порядка (скажем, прогнозы 3 и 4 не будут отличаться от прогнозов 3.9 и 3.95 в плане качества ранжирования друг относительно друга).

§1.2 Попарные методы

Вспомним, что изначально мы определяли задачу ранжирования через пары объектов. Если записывать это формально, то получим функционал ошибки

$$\sum_{(i,j) \in R} [a(x_j) - a(x_i) < 0],$$

где R — множество пар, для которых известен порядок. Этот функционал не является дифференцируемым, но мы уже решали такую проблему в линейной классификации — надо заменить индикатор ошибки $[z < 0]$ на его гладкую верхнюю оценку $L(z)$:

$$\sum_{(i,j) \in R} [a(x_j) - a(x_i) < 0] \leq \sum_{(i,j) \in R} L(a(x_j) - a(x_i)).$$

В качестве оценки $L(z)$ можно брать, например, логистическую функцию $L(x) = \log(1 + e^{-\sigma x})$ с параметром $\sigma > 0$ — в этом случае получим метод RankNet. Оптимизировать данный функционал можно обычным стохастическим градиентным спуском. Если использовать линейную модель $a(x) = \langle w, x \rangle$, то один шаг будет иметь вид

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

Существует эмпирическое наблюдение, позволяющее перейти к оптимизации произвольной метрики ранжирования F . Оказывается, для этого надо домножить смещение на изменение метрики ΔF_{ij} , которое произойдёт при перестановке x_i и x_j местами в ранжировании:

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

Данный метод носит название LambdaRank.

Также по аналогии с методом опорных векторов можно вывести метод RankSVM:

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

Отметим, что нередко именно попарный подход оказывается лучшим при решении задачи ранжирования.

§1.3 Списочные методы

Мы уже отмечали выше, что непосредственно оптимизировать метрику качества ранжирования вряд ли получится из-за её дискретной структуры. Такая проблема возникала и раньше (например, при обучении метрик или при визуализации),

и решали мы её путём введения некоторого вероятностного распределения. Разберём метод ListNet, который позволяет непосредственно учитывать порядок объектов в процессе обучения.

Рассмотрим один запрос q , для которого надо отранжировать n_q документов (d_1, \dots, d_{n_q}) . Для этих документов известны истинные оценки релевантности (y_1, \dots, y_{n_q}) , которые определяют истинное ранжирование. Пусть также имеет некоторая модель $a(x)$, которая выдаёт оценки (z_1, \dots, z_{n_q}) . Метрика качества ранжирования (например, nDCG) измеряет, насколько ранжирование по оценкам модели близко к истинному ранжированию.

В постановке, которую мы только что описали, модель выдаёт одну конкретную перестановку документов для данного запроса, и мы измеряем качество этой перестановки. Сгладим этот процесс: предположим, что на самом деле модель выдаёт распределение на всех возможных перестановках документов, причём вероятность конкретной перестановки π определяется как

$$P_z(\pi) = \prod_{j=1}^{n_q} \frac{\varphi(z_{\pi(j)})}{\sum_{k=j}^{n_q} \varphi(z_{\pi(k)})},$$

где φ — произвольная неубывающая и строго положительная функция. Данные вероятности обладают рядом полезных свойств:

- Вероятности $P_z(\pi)$ задают распределение на множестве всех перестановок n_q элементов.
- Пусть перестановка π ставит объект x_i выше объекта x_j , и $z_i > z_j$. Если поменять эти объекты местами в перестановке (то есть поставить x_j выше, чем x_i), то новая перестановка будет иметь меньшую вероятность, чем старая. Иными словами, чем ближе перестановка к оптимальной, тем выше её вероятность.
- Максимальную вероятность имеет перестановка, которая сортирует объекты по убыванию z_i ; минимальную вероятность имеет обратная к ней перестановка.

Таким образом, данные вероятности отдают предпочтение тем перестановкам, которые ближе к сортировке объектов по предсказаниям алгоритма. Значит, мы действительно получим способ «сглаживания» ранжирования документов по прогнозам z_i .

Всего перестановок объектов $n_q!$, и посчитать по ним всем матожидание не представляется возможным. Чтобы упростить подсчёты, рассмотрим вероятность $P_z(j)$ попадания объекта x_j на первое место после перестановки. Можно показать, что они вычисляются по формуле

$$P_z(j) = \frac{\varphi(z_j)}{\sum_{k=1}^{n_q} \varphi(z_k)}$$

Данные вероятности образуют распределение на всех n_q документах. Также для объектов с $z_i > z_j$ выполнено $P_z(i) > P_z(j)$ — то есть введённые вероятности задают на объектах порядок, совпадающий с ранжированием по оценкам модели.

Теперь мы можем сравнить истинное ранжирование документов и ранжирование по оценкам модели, посчитав кросс-энтропию между соответствующими им распределениями:

$$Q(y, z) = - \sum_{j=1}^{n_q} P_y(j) \log P_z(j).$$

Если взять, например, $\varphi(x) = \exp(x)$, то кросс-энтропия будет дифференцируема по оценкам модели — а значит, можно найти производные и по параметрам модели. Благодаря сглаживанию мы смогли ввести функционал, который отражает требования к перестановке объектов, и при этом позволяет обучение градиентными методами.