

Машинное обучение: ранжирование (MLR) сегодня вся лекция отсебятина

Кураленок И.Е.

Яндекс

10 мая 2012 г.

План

- 1 Данные
 - Оценка ранжирования
 - Виды данных для обучения
- 2 Целевые функции
 - Pointwise
 - Pairwise
 - Listwise
- 3 Известные датасеты
- 4 Проблемы ранжирования
- 5 Развитие области

Где в поиске используется ранжирование?

- Фильтрация результатов
- Общее ранжирование на нижнем
- Переранжирование вертикальных поисков

Качество ранжирования

Качество ранжирования — важнейшая характеристика поисковой системы, которая определяет длинную долю (!).

На сегодняшний момент качество зависит от:

- 1 Оценки качества поиска.
- 2 Способа построения DS.
- 3 Факторов поиска.
- 4 Качества обучения.

Сигналы для ранжирования

- Классическая оценка ранжирования (асессоры)
- Оценка пар (mechanical turk, etc.)
- Пользовательское поведение
 - Клики
 - Парное сравнение
 - Поведение на смешивании
 - Динамическое ранжирование

Кренфилдская методология

- Переведем ранжирование документов в последовательность чисел
- Оценим последовательности чисел
 - Как множества срезов (p10, r1000, 11-point graph, etc.)
 - Целиком (dcg, ndcg, pF, etc.)
- Усредним по запросам (веса, вид усреднения, etc. на выбор)

Релевантность

Релевантность — это способ сказать насколько документ подходит запросу

К определению релевантности было много подходов, один из наиболее интересных предложил Stefano Mizzaro в (MFRS) Этот подход сейчас де-факто стандартный, но к нему есть масса вопросов:

- Что делать с неоцененными (b-pref, inducted DCG, etc.)
- Как написать правильную инструкцию?
- Как правильно взвесить разные уровни релевантности?
- Слишком много аспектов релевантности...
что круче авторитетность или свежесть?
- Разнообразие

ЗЫ: у меня есть про это полугодовой курс :).

Данные

В IR данные состоят из:

- Запросов (50 (штук)–100к+, нормально 10к)
выдуманных из головы (TREC), взятых из логов (IMAT2009, Yahoo LTR, LETOR, etc.), специально подобранных
- Документов (10к–10млн+, нормально 300к)
pooling (много где), topN + дооценки, topМного, специально найденные
- Оценок (10к-10млн+, нормально от 100к)
от экспертов задающих запросы (TREC), описания с переводом на "описательный" уровень (IMAT2009), неизвестно откуда

А еще экспертов может быть много и их оценки могут быть неконсистентны (!)

Области значений факторов

Для обучения нам приходится переводить факторы в $[0, 1]^n$, что не всегда естественно:

- например raw факторы нам приходится приводить в BM25 etc;
- категориальные факторы приводятся в бинарные/n-арные, в итоге получают лишнее отношение порядка;
- неявные факторы (например collaborative filtering) не могут быть использованы;
- оптимальное распределений значений фактора зависима от метода обучения (!);
- невозможно использовать контекстно-зависимые факторы.

Виды целевых функций

Целевая функция зависит от способа измерения качества.
Если дан стандартный DS, то идеального ранжирования можно достичь многими способами:

- научиться ставить правильные метки документам (pointwise);
- построить отношения частичного порядка (pairwise);
- предсказать наиболее верное ранжирование (listwise).

Угадывание оценок

Наиболее естественный с точки зрения ранжирования на базовом способ.

Можно пойти несколькими способами:

- взвесить оценки ($g : R \rightarrow \mathbb{R}$) и найти функцию наиболее близкую к ним;
- построить систему взвешенных бинарных классификаторов и устроить структуру ранжирования;
- сделать мультиклассификатор и повесить классы.

Угадывание оценок: buzzwords

OPRF (1989), SLR (1992), IR-SVM (2006), McRank (2007)

Построение порядка

$$r : (q, d_1, d_2) \rightarrow \{-1, 1\}$$

Однако, для поиска на базовом мы не знаем всего множества \mathbb{D} , поэтому делаем функцию ранжирования в 2 этапа:

$$f : (q, d) \rightarrow \mathbb{R}$$
$$r(q, d_1, d_2) = \begin{cases} 1, & f(q, d_1) \geq f(q, d_2) \\ -1, & f(q, d_1) < f(q, d_2) \end{cases}$$

Именно такая кривь не позволяет нам работать с контекстом.

Построение порядка: buzzwords

RankSVM (2000), RankBoost (2003), LambdaRank(2006),
GBRank (2007), RankRLS (2007), SortNet (2008), MatrixNet
(2009), YetiRank (2010), CRR (2010)

Построение ранжирования

$$r : (q, \mathbb{D}^k) \rightarrow \mathbb{R}$$

Очень перспективно, однако снова упираемся в невозможность работать со всем \mathbb{D} . В итоге:

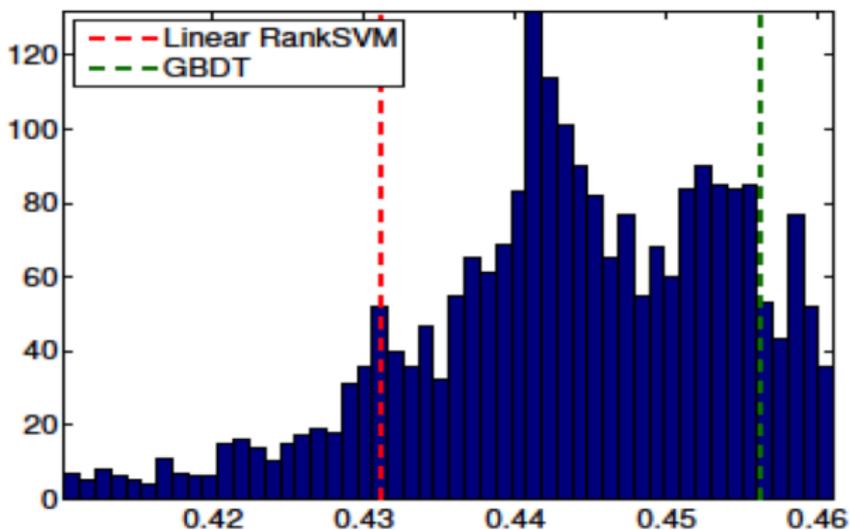
$$f : (q, d) \rightarrow \mathbb{R}$$
$$r(q, D) = \mu(q, D, \pi(q, D|f))$$

Построение ранжирования: buzzwords

классика AdaRank (2007), ListNet (2007), SoftRank (2008)
learning to blend BolzRank (2009), GBlend (2010)

Кто же круче?

*А вы, друзья, как ни садитесь,
Всё в музыканты не годитесь*



Где можно посоревноваться в ранжировании?

Много статей написано, однако большая их часть оперирует “игрушечными” данными.

Чтобы академия сделала что-то полезное для индустрии были открыты такие коллекции:

LETOR 2007г, данные от MS Research, есть несколько вариантов (4 штуки: от игрушечных, до нормальных)

Интернет-математика 2009 2009 (KO), данные Яндекс, выгружен полный пул (!), на котором мы сами на тот момент настраивались.

Yahoo LTRC 2010, Yahoo (KO) 2 дороги: обычное ранжирование и transfer learning.

Что сложного в web-ранжировании?

- 1 Формулы по оцененным: нужна специальная процедура
- 2 Существенная стоимость набора данных
- 3 Непонятное отношение целевой функции к жизни
- 4 Необходимость смешивания
- 5 Изменение окружения: интернет меняется
- 6 Положительная обратная связь

Чем сложны пользователи?

Можно было бы научиться предсказывать клики и максимизировать долю трафика напрямую.

Сложности работы с пользовательскими данными:

- 1 умеют менять свои интересы в момент поиска;
- 2 хотят “популярное” а не релевантное;
- 3 неоднородны;
- 4 надежные данные есть только по высокочастотникам (простым запросам);
- 5 оценки смещены позицией (positional bias).

Что можно поделаться с пользователями

- Искать не “клики”, а поведение (длинные клики, например).
- Попарные оценки документов выдачи.
- Обучение по смешанным выдачам.
- Персонализация ранжирования.

“Другие” факторы ранжирования

Есть много факторов, которые мы не можем использовать из-за постановки задачи в \mathbb{R}^n .

- Факторы на последовательностях.
- Ассоциации.
- Категориальные данные.

On-line ранжирование

Кроме web есть еще поиск по коллекциям, intranet, и т.п.
Многие (в т.ч. и Я) предоставляют поиск по этим данным,
однако формула в этом поиске фиксированна:

- не учитывает особенностей коллекции.
- не использует клики (так как свойств их не знает).

Если бы можно было формулу обучать on-line поиск учился бы сам в процессе использования (!)

Обучение смешиванию

Есть несколько параллельных источников данных: новости, блоги, web, etc. У всех у них разные факторы поиска, надо смешать результаты и получить хороший результат.

Динамическое ранжирование

Возможно делать активное обучение в процессе работы. Для этого можно позаимствовать логику у multi-armed bandits:

$$g : (q, \mathbb{D}^k) \rightarrow \mathbb{R}$$
$$\operatorname{argmin}_A Ng(q, D^*) - \sum_{i=0}^N g(q, D_i(A, X_i))$$

где X_i – опыт (поведение) полученное на всех предыдущих выдачах.

Перенос оценок

У Я есть много оценок по России, можно ли использовать эти оценки для Турции?

Пока ответ удручающий: НЕТ

Развитие оценки качества поиска

На сегодня в мире НЕТ ни одного хорошего способа оценки качества поиска.