Яндекс на РОМИП'2009. Оптимизация алгоритмов ранжирования методами машинного обучения

© Гулин Андрей, Карпович Павел, Расковалов Денис,

Сегалович Илья

Яндекс
gulin@yandex-team.ru, pavelkarpovich@yandex.ru,
denplusplus@yandex-team.ru, iseg@yandex-team.ru

Аннотация

Данная статья является отчетом об участии в дорожке поиска по веб-коллекции конференции РОМИП'2009. Описывается опыт использования методов машинного обучения при оптимизации качества ранжирования поисковой программы по наборам документов BY.WEB и KM.RU.

1. Модель ранжирования

Задача ранжирования состоит в упорядочении документов коллекции по убыванию степени их соответствия запросу, т.е. более релевантные документы должны иметь более высокий ранг. Для решения этой задачи большинство поисковых систем используют «функции релевантности» (способ получить числовую оценку соответствия документа запросу). Другими словами, поисковая программа вычисляет значение релевантности документа в контексте запроса и сортирует коллекцию согласно данному числу.

Как правило, релевантность является функцией от набора факторов. В качестве факторов выступают различные числовые характеристики, которые должны помогать различать релевантные документы и нерелевантные. Для многих поисковых систем

результирующая функция релевантности является простой комбинацией небольшого множества (5-15 штук) факторов (сумма [1], взвешенная сумма [2]). Некоторые сложные факторы могут быть сами использованы в качестве самостоятельных функций ранжирования.

Наш подход использует значительное количество факторов - ранжирование коллекции BY.WEB основано на 163 компонентах. Большинство из факторов представляют собой простые числовые характеристики документа или запроса. Ключевым моментом в построении ранжирования является способ комбинации факторов, т.е. вид функции релевантности. Для получения функции ранжирования используются методы машинного обучения [3]. Такой подход позволяет достаточно легко добавлять новые факторы, тем самым развивая и улучшая поисковую систему.

2. Факторы ранжирования

При работе с коллекциями BY.WEB и KM.RU были использованы различные наборы факторов. Как было обозначено выше, ранжирование корпуса BY.WEB использует 163 фактора. Поиск по коллекции KM.RU построен на 69 факторах.

Индексация документов для корпуса KM.RU была проведена при помощи общедоступной программы «Яндекс.Сервер» (http://company.yandex.ru/technology/server/). В данном случае факторы ранжирования являются прямыми функциями от текста документа и поискового запроса. Приведем несколько примеров используемых факторов:

- наличие точного вхождения запроса в тексте документа;
- наличие точного вхождения запроса в заголовке документа;
- группа факторов, состоящая из различных модификаций формулы Okapi_BM25 (http://en.wikipedia.org/wiki/Okapi_BM25);
- русскоязычность документа.

Коллекция BY.WEB представляет собой выборку документов белорусского интернета по состоянию на май 2007 года. В связи с этим при ранжировании документов корпуса использовалась информация о ссылках и расширенный набор из 163 факторов. Примеры из расширенного набора факторов:

- логарифм количества ссылок на документ;
- процент ссылок на документ, содержащих точное вхождение запроса.

3. Машинное обучение

Мы опишем процесс получения формулы релевантности для коллекции BY.WEB. Способ получения формулы для корпуса KM.RU полностью идентичен.

3.1 Обучающая выборка

Для применения методов машинного обучения по данным таблиц релевантности дорожки BY.WEB конференции РОМИП'2008 была сформирована обучающая выборка. В обучающей выборке были представлены документы для 499 поисковых запросов, коллекция содержала ~45000 пар <запрос, документ> с известными оценками релевантности. Для каждой из пар <запрос, документ> были вычислены факторы. Оценки релевантности брались из таблицы OR-оценки. В данных таблиц содержатся оценки трех типов: «vital», «notrelevant», «cantbejudged». При обучении оценки преобразовывались в числовые значения: тип «vital» переводился в оценку релевантности равную 0.4, остальным типам соответствовало значение 0 (значения являются эвристиками).

Естественной постановкой задачи машинного обучения было бы приближение оценки релевантности на множестве обучающей выборки и построение функции ранжирования, исходя из минимизации функционала среднеквадратичного отклонения от значения оценок. Хорошее решение данной задачи гарантирует построение правильного порядка документов обучающей выборки. Однако требование о построении приближения оценки релевантности является более сильным, чем требование о построении правильного порядка документов, поэтому мы использовали в качестве целевого функционала для задачи оптимизации другую функцию - pfound.

3.2 Метрика pfound

Предположим, что мы отсортировали n документов с известными оценками релевантности по запросу в каком-то порядке. Мы опишем модель, в которой пользователь просматривает список документов сверху вниз в поисках релевантного документа. Значение метрики pfound будет оценкой вероятности найти релевантный результат в нашей модели. Формула метрики pfound выглядит следующим образом:

$$pfound = \sum_{i=1}^{n} pLook[i] * p\,\mathrm{Re}l[i]$$

где ***pLook[i]*** – вероятность просмотреть i-й документ из списка, ***pRel[i]*** – вероятность того, что i-й документ окажется релевантным.

Значениями ***pRel[i]*** в нашей модели являются оценки релевантности по запросу. Как было отмечено выше, мы преобразовали тип оценки «vital» из таблиц релевантности в числовое значение 0.4, т.е. мы считаем, что документ, помеченный как «vital», с вероятностью 40% окажется релевантным для пользователя. Типы «notrelevant» и «cantbejudged» соответствуют вероятности 0%.

Для оценки вероятности просмотра i-го результата мы используем два предположения: пользователь просматривает результаты поиска последовательно сверху вниз; он прекращает процесс в случае нахождения релевантного результата либо может остановиться без каких-то определенных причин («надоело»). Формула ***pLook[i]*** взята следующая:

$$pLook[i] = pLook[i-1] * (1 - p\,\mathrm{Re}l[i-1]) * (1 - pBreak)$$

где pLook[i-1] – вероятность того, что пользователь просмотрит (i-1)-ю позицию; 1-pRel[i-1] – вероятность того, что пользователь не удовлетворится (i-1)-й позицией; 1-pBreak – вероятность того, что он не остановится по независящим от нас причинам (в нашей модели pBreak выбрана равной 0.15).

Мы описали расчет значения pfound в контексте одного запроса. Целевой задачей процесса машинного обучения является максимизация усредненного значения метрики по всем 499 запросам обучающей выборки.

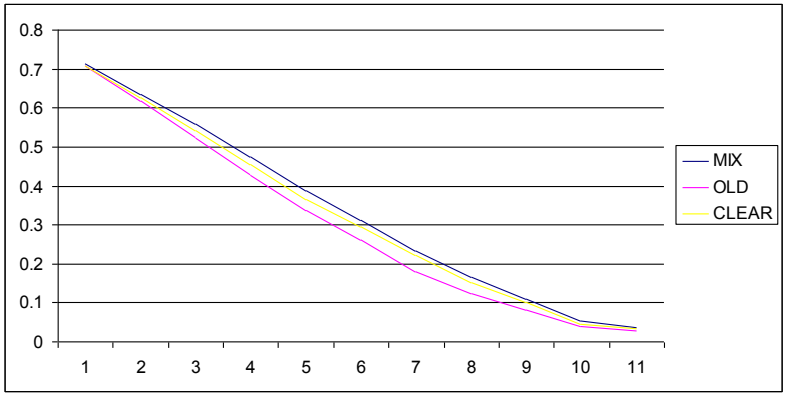
3.3 Способ оптимизации

В качестве функции релевантности ранжирования коллекций BY.WEB и KM.RU использовались полиномы. Вид полиномов подбирался генетическим алгоритмом. Для подбора коэффициентов при мономах использовалась модификация стохастического алгоритма Differential Evolution (http://en.wikipedia.org/wiki/Differential_evolution).

4. Результаты

Для каждой из коллекций на оценку было послано по три варианта ранжирования, которые отличаются используемыми формулами релевантности. Первый тип (CLEAR) - формулы, полученные в процессе машинного обучения из предыдущей секции на данных РОМИП'2008. Второй тип (OLD) – формулы, полученные с использованием внутренних данных Яндекса (обучающие выборки веб-документов большого размера, 10 000 - 20 000 запросов). Третий тип (MIX) – сумма формул CLEAR и OLD. В связи с тем, что объемы обучающих выборок при получении формул CLEAR были небольшими, варианты с формулами MIX и OLD были отправлены на оценку для исследования эффектов переобучения.

Приведем 11-точечные графики TREC OR-оценки коллекции BY.WEB для трех вариантов:



Наихудшим образом показывает себя формула OLD, которая получена по данным, отличным от коллекции BY.WEB. Однако при помощи OLD получается побороться с эффектом переобучения формулы CLEAR, и лучшим ранжированием является MIX.

По сравнению с другими поисковыми системами предложенные нами способы ранжирования показывают хорошие результаты. Топ систем по метрике Precision(10) для BY.WEB:

MIX	CLEAR	xxx-4	xxx-12	OLD	xxx-11	xxx-10
0.48849	0.4857	0.4816	0.4743	0.4681	0.4622	0.42363

Литература

[1] А.В. Сафронов "HeadHunter на РОМИП-2008". Труды РОМИП 2007-2008, Санкт-Петербург: НУ ЦСИ, 2008. (http://romip.ru/romip2008/2008_03_headhunter.pdf)

[2] С. Татевосян, Н. Брызгалова "KM.RU на РОМИП-2008. Оптимизация параметров поискового алгоритма". Труды РОМИП 2007-2008, Санкт-Петербург: НУ ЦСИ, 2008. (http://romip.ru/romip2008/2008_07_km.pdf)

[3] Tie-Yan Liu, "Learning to rank for information retrieval", WWW-2008 (http://research.microsoft.com/en-us/people/tyliu/learning_to_rank_tutorial_-_www_-_2008.pdf)