Отчет по конкурсу Relevance Prediction Challenge

Сергей Гуда
Ростов-на-Дону, Россия
gudasergey@gmail.com

Денис Рябов
Ростов-на-Дону, Россия
dryabov@yandex.ru

ABSTRACT

В статье представлен отчет команды S-n-D об идеях и методах, используемых при участии в конкурсе «Relevance Prediction Challenge», проводимого компанией Яндекс в 2011 году в рамках цикла конкурсов «Интернет-математика».

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]

General Terms

Algorithms, Experimentation, Performance

Keywords

Click Log Analysis, Random Forest

1. ВВЕДЕНИЕ

В 2011 году для хранения информации о количестве проведенных за год конкурсов «Интернет-математика» впервые понадобился второй бит информации — ведь ранее компания Яндекс проводила один конкурс в год (или же не проводила вовсе), а в этот раз таких конкурсов было два. И если первый был посвящен теме, которая на первый взгляд к поиску имеет отдаленное отношение — выделение изображений, относящихся к одной и той же панораме, — то второй конкурс к поисковым технологиям имеет самое непосредственное отношение.

Суть конкурса [1] состоит в том, чтобы на основе данных о работе пользователя с результатами поисковой выдачи (клики по ссылкам, переформулировка запроса), определить релевантность той или иной ссылки введенному запросу с учетом региона пользователя.

Это только на первый взгляд кажется, что если пользователь кликнул по ссылке — значит он нашел то, что искал. В действительности, пользователь может кликать все ссылки подряд в поиске нужной информации, а может ограничиться несколькими первыми и не дойти до действительно релевантного документа.

«В цифрах» задача конкурса следующая. Используя известные (асессорские) оценки для 8410 пар Запрос-Регион (71930 сочетание Запрос-Регион-Ссылка) необходимо предсказать релевантность ссылок для ещё 3219 пар на основе имеющейся истории пользовательских запросов и кликов по ссылкам. Качество предсказания определяется организатором конкурса по средней величине AUC [2] (чем ближе эта величина к единице — тем лучше результат, при результаты лучших команд в конкурсе находились в интервале $0.6 - 0.7$). Объем истории (практически 17 Гб «сырых» данных, 340 796 067 строк текста) исключает какую-либо возможность ее ручной обработки.

В своем подходе к решению данной задачи мы придерживались простой схемы: по информации о кликах пользователей составлялась матрица объектов-признаков (см. [3]), которая затем подавалась на вход стандартному классификатору. В качестве объектов естественно выбрать тройки (QueryID, RegionID, URLID), тогда целевым параметром будет релевантность документа запросу в заданном регионе: 0 или 1. Для каждой такой тройки можно вычислить массу параметров, которые составят ее признаковое описание. В частности, в попытках улучшить описанный далее метод, мы рассматривали реализацию, включающую более 2000 различных параметров.

Эксперименты с различными классификаторами в Rapid Miner'e [4] показали, что лучшим для данной задачи является случайный лес [5] из пакета Weka [6]. Для ранжирования документов одного запроса использовались возвращаемые классификатором вероятности классификации документа как относящегося к классу релевантных (значения релевантности 1). Все попытки авторов улучшить стандартный классификатор Random Forest для учета особенностей задачи, закончились неудачей. Финальный результат был получен на стандартном методе, почти за месяц до окончания конкурса.

Процесс подготовки матрицы объектов-признаков шел в несколько этапов. Вначале рассчитывались «сырые» параметры троек (QueryID, RegionID, URLID). Затем они подвергались преобразованию: в парах признаков с большой мощностью множества совпадающих элементов один из признаков заменялся их отношением. Такое преобразование переводит биссектрису $f_1 = f_2$ на плоскости признаков (f_1, f_2) в прямую $f_1 = 1$, что облегчает обуче-

ние случайного леса. После этого происходило удаление малоинформативных признаков.

Далее описываются подробности методики: подготовка описывающих тройку (QueryID, RegionID, URLID) признаков, параметры стандартного классификатора Random Forest, описание неудавшегося подхода, основанного на минимизации AUC; описание исходного кода программы.

2. ПОДГОТОВКА ДАННЫХ

Естественно, что обрабатывать исходные данные в «сыром» виде является малоперспективной задачей, тем более если предполагать, что разработанные алгоритмы могут быть использованы на «реальном производстве». Также не менее очевидно, что от исходных данных необходимо перейти к некоторым статистическим характеристикам — новым параметрам, которые можно использовать для дальнейшего анализа и которые легко обновлять по мере поступления новых данных.

Какие же параметры можно извлечь из предложенного материала? Мы выделили две группы параметров — характеризующих запрос в целом (это позволяет учесть, например, ситуацию, когда пользователь переформулирует запрос), и характеризующих конкретную ссылку в выдаче.

Для запроса:

- Вероятность быть последним в сессии
- Среднее количество ссылок
- Среднее количество кликов
- Средняя продолжительность работы с запросом как время, прошедшее от запроса до последнего клика
- Средняя продолжительность работы с запросом как время, прошедшее от запроса до следующего запроса в сессии
- Средняя позиция ссылки, по которой кликнут первой
- Среднее время оценивания первой страницы выдачи как время от запроса до первого клика

Для ссылки:

- Средняя позиция по данному запросу
- Вероятность клика по ссылке
- Какой (в среднем) по счету кликают на эту ссылку
- Вероятность того, что по ссылке кликнут последней
- Какой (в среднем) по счету кликают на эту ссылку в обратном порядке
- Вероятность того, что по ссылке кликнут последней
- Вероятность клика по ссылке, которая на одну позицию выше данной
- Вероятность того, что ниже в выдаче имеются ссылки, по которым кликали (данный параметр позволяет косвенно учесть повторный просмотр страницы выдачи)
- Среднее время просмотра ссылки как время до следующего клика (применимо к ссылкам, на которые кликали не последними)
- Среднее время просмотра ссылки как время до следующего запроса (применимо к ссылкам, на которые кликали последними)
- Вероятность того, что ссылка находится на последней просмотренной странице выдачи
- Средняя позиция ссылки на странице выдачи (1-10)
- Среднее количество позиций до ссылки, по которой кликнули перед данной
- Среднее количество позиций до ссылки, по которой кликнули после данной
- Среднее количество позиций до ближайшей предыдущей ссылке в выдаче, по которой кликнули
- Среднее количество позиций до ближайшей следующей ссылке в выдаче, по которой кликнули
- Вероятность того, что по ссылке кликнули два раза подряд
- Вероятность того, что к данной ссылке вернулись после клика по одной из нижерасположенных ссылок
- Вероятность того, что после клика по данной ссылке пользователь кликал по ссылкам выше в выдаче

Везде, где выше в списке упоминается «среднее время», имеется в виду среднее значение от величины $\log(1+t)$. Использование логарифмической шкалы в данном случае оправдано большим разбросом значений временных интервалов.

Указанные вероятности и средние значения считались независимо для четырех типов группировки данных:

- (QRU) Усреднение для каждой тройки Запрос-Регион-Ссылка
- (QU) Усреднение для каждой пары Запрос-Ссылка (усредненные по региону значения, для тех случаев когда статистики по нужному региону может не быть)
- (RU) Усреднение для каждой пары Регион-Ссылка (учет соответствия ссылки региону)
- (U) Усреднение для каждой Ссылки (интегральные характеристики ссылки в целом)

Также на вход классификатора подавались относительные характеристики, а именно отношения для каждого из параметров, подсчитанные на основе разных способов группировки: QRU/QU, QRU/RU, QRU/U, RU/U, и QU/U.

Итого $26 \times (4+5) = 234$ входных параметра для задачи классификации.

Полученные признаки содержат множество совпадающих элементов, например, параметры, относящиеся к группам QRU и QU, совпадают для всех троек (QueryID,

RegionID, URLID) с парой (QueryID, URLID), встречающейся только для одного региона RegionID. Случайный лес плохо классифицирует пары признаков, бóльшая часть значений которых сосредоточена на прямой, не параллельной осям координат. Поэтому в дальнейшем признаки подвергались преобразованию. В цикле на каждой итерации определялась пара признаков с наибольшим числом уникальных совпадающих пар значений, и признак пары с бóльшим числом нулевых значений заменялся отношением. Цикл продолжался, пока процент уникальных совпадающих значений признаков пары был больше 50%.

3. МЕТОД РАСЧЕТА

Случайный лес пакета Weka использовался со следующими параметрами:

- максимальная глубина деревьев $d = 14$;
- число признаков при выборе разреза $K = 10$;
- начальное значение генератора случайных чисел $seed = 1$ (от него ничего не зависит).

Код случайного леса был переделан так, чтобы построение деревьев шло до максимально возможной глубины, а тестирование классификатора можно было проводить с различными значениями глубины d.

4. ПОДХОД НА ОСНОВЕ МИНИМИЗАЦИИ AUC

За несколько дней до окончания конкурса в условиях жесткого цейтнота был запрограммирован метод ранжирования, основанный на минимизации AUC для каждого запроса. Для вычисления AUC есть еще одна простая формула, аналогичная представленной в [2]

$$AUC = 1 - \frac{\tau}{\tau_{max}}, \qquad (1)$$

где τ — количество инверсий в отсортированном алгоритмом ряду URLID, $\tau_{max} = n_0 n_1$ — максимально возможное число инверсий в этом ряду, n_0, n_1 — количество нерелевантных и релевантных URLID. Например, если релевантные URLID обозначать «+», а нерелевантные «−», то в ряду $+ - - + - - -$, отсортированном алгоритмом по возрастанию релевантности, $n_0 = 5$, $n_1 = 2$, $\tau = 2$, $\tau_{max} = 10$; инверсии образуют второй и четвертый URLID, третий и четвертый URLID.

В соответствии с формулой (1), максимизация AUC эквивалентна минимизации числа инверсий. Отсюда получается альтернативный метод: в качестве объектов рассматривать четверки (QueryID, RegionID, URLID1, URLID2); целевым параметром брать факт образования инверсии: образуют URL1 и URL2 инверсию или нет. По предсказанным вероятностям инверсий всех пар URLID одного запроса определяется такой порядок URL, при котором сумма вероятностей инверсий будет минимальной. Это и есть искомое ранжирование.

Сначала вероятность образования инверсии двумя URLID считалась, с использованием отдельных деревьев случайного леса, обученного предсказывать релевантность троек (QueryID, RegionID, URLID). Рассматривались способы ранжирования набора URL одного запроса каждым деревом леса, и в качестве вероятности бралась относительная частота инверсий. Когда данный подход не принес ожидаемого улучшения, на матрице объектов-признаков для четверок (QueryID, RegionID, URLID1, URLID2) был натренирован собственный классификатор. Признаковым описанием четверки (QueryID, RegionID, URLID1, URLID2) выступало отношение параметров, описывающих тройки (QueryID, RegionID, URLID1) и (QueryID, RegionID, URLID2), и параметры первой тройки (QueryID, RegionID, URLID1). При составлении обучающей выборки для каждого запроса выбиралось одинаковое количество пар (URLID1, URLID2) с различными значениями релевантности. Преимуществом пользовались пары с меньшей средней позицией в выдаче.

Авторы предполагают, что причина неудачи данного подхода по сравнению с описанным ранее может скрываться в программистской ошибке или в чересчур сильной редукции значений релевантности в обучающей выборке до двух: 0 и 1. Это привело к невозможности использовать при обучении пары URL с одинаковыми значениями релевантности.

5. ИСХОДНЫЙ КОД

Код написан в стиле экстремального программирования двумя программистами (поэтому не судите строго). Проект состоит из трех частей: части, написанной на языке Matlab, части на языке C++ и пакета Weka на Java с некоторыми изменениями.

Часть, написанная на Matlab'е, отвечает за представление логов в виде таблицы и сохранение их в двоичной форме (файл makeTableRows.m), построение небольшого числа признаков (функция evaluateFeatures), подготовке файла для отправки на сайт конкурса.

Часть 2, написанная на C++, готовит матрицу объектов-признаков.

Часть 3 при помощи Weka отвечает за тренировку классификатора RandomForest и локальное тестирование (класс mainTestingRandomForest.java).

6. REFERENCES

[1] http://imat-relpred.yandex.ru/.

[2] C. X. Ling, J. Huang, and H. Zhang. AUC: a statistically consistent and more discriminating measure than accuracy. In *Proceedings of the Eighteenth International Joint Conference of Artificial Intelligence*, pages 519–526, 2003.

[3] К. В. Воронцов. «Машинное обучение». Курс лекций на сайте http://www.machinelearning.ru/.

[4] http://rapid-i.com/.

[5] L. Breiman. Random forests. *Machine Learning*, 45:5, 2001.

[6] http://www.cs.waikato.ac.nz/ml/weka/.