электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 004.021

Алгоритм ранжирования MatrixNet

Ватулин А.М., *студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»

Котович М.А., *студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»

Научный руководитель: Гапанюк Ю.Е., к.т.н, доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»
gapyu@bmstu.ru

Качество поиска

Основная задача поисковой системы – дать ответ на вопрос пользователя. Когда пользователь задает запрос, поисковая система не обращается к каждому сайту в интернете, а ищет по базе известных ей страниц – поисковому индексу. Там она находит все страницы со словами из запроса. Ссылки на эти страницы пользователь и видит на страницах результатов поиска. Поисковая система должна расположить найденные страницы в нужном порядке – так, чтобы сверху оказались наиболее подходящие пользователю (наиболее **релевантные**). Упорядочивание результатов поиска в соответствии с запросом пользователя называется **ранжированием**. Именно ранжирование определяет качество поиска – то есть качество ответа на вопрос, заданный в поисковой строке.

Машинное обучение

Поисковая система должна научиться строить правило, которое определяет для каждого запроса, какая страница является хорошим ответом на него, а какая – нет. Для этого поисковая машина анализирует свойства веб-страниц и поисковых запросов. У всех страниц есть какие-то признаки. Некоторые из них – **статические** – связаны с самой страницей, например, количество ссылок на эту страницу в интернете. Некоторые признаки – **динамические** - связаны одновременно с запросом и страницей – например, присутствие в тексте страницы слов запроса, их количество и расположение. У

поискового запроса тоже есть свойства, например, геозависимость - это означает, что для хорошего ответа на этот запрос нужно учитывать регион, из которого он был задан. Свойства запроса и страницы, которые важны для ранжирования и которые можно измерить числами, называются **факторами ранжирования**. Для точного поиска важно учитывать очень много разных факторов.

Кроме факторов ранжирования поисковой системе необходимы образцы – запросы и страницы, которые люди считают подходящими ответами на эти запросы. Оценкой того, насколько та или иная страница подходит для ответа на тот или иной запрос, занимаются специалисты – асессоры. Они берут поисковые запросы и документы, которые поиск находит по этим запросам, и оценивают, насколько хорошо найденный документ отвечает на заданный запрос. Из запросов и хороших ответов составляется обучающая выборка. Она должна содержать самые разные запросы, причем в тех же пропорциях, в которых их задают пользователи. На обучающей выборке поисковая система устанавливает зависимость между страницами, которые асессоры посчитали релевантными запросам, и свойствами этих страниц. После этого она может подобрать оптимальную формулу ранжирования – которая показывает релевантные запросу сайты среди первых результатов поиска.

MatrixNet

В жадном алгоритме оптимизации, используемом Яндексом, функции релевантности документа d относительно запроса q выглядит следующим образом:

$$f_r(q,d) = a_1 h_1(q,d) + a_2 h_2(q,d) + \ldots + a_n h_n(q,d) \qquad (1)$$

MatrixNet - это метод получения функций семейства «слабых алгоритмов обучения» (или «weak leaners»), которые в формуле (1) обозначаются как $h_k(q,d)$. Алгоритм получения функций $h_k(q,d)$ очень сложен. Нужно только заметить, что количество функций $h_k(q,d)$ достаточно большое, десятки тысяч. Коэффициенты a_k – малые величины. Теоретически возможны варианты, когда младшие a_k достаточно большие, но практика показала, что это не так. И a_k могут быть меньше нуля, т.е. некоторые слагаемые дают отрицательный вклад в релевантность. Обучение проходит на базе оцененных пар (запрос, документ), число которых скорее всего уже больше 5 миллионов. Для понимания особенностей работы ранжирования Яндекса гораздо

большую ценность представляет то, что получается на выходе этого алгоритма: функции $h_k(q,d)$, которые заданы деревьями решений.

В качестве простого примера функции заданной деревом решений можно рассмотреть конструкцию, представленную на рисунке 1.

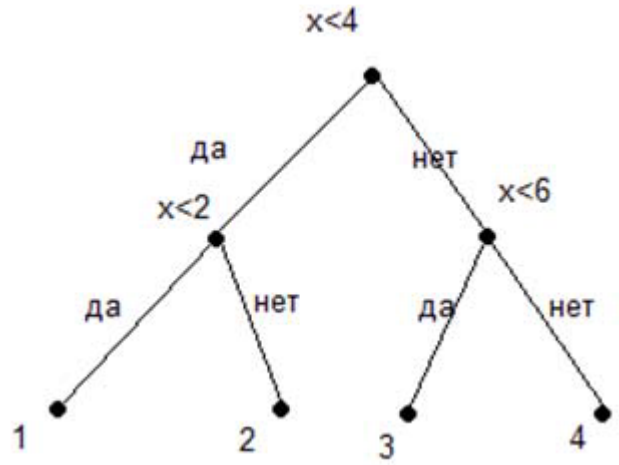

Рис. 1. Функция одного аргумента, заданная деревом решений.

У этой функции есть две особенности:

- она кусочно-постоянная. Т.е. на некоторых интервалах при изменении x, значение самой функции не меняется. Например, при $2 \leq x < 4$ (т.е. при x от 2 до 3,999999999....) $f(x) = 2$
- при переходе x через условие ветвления, функция меняется скачком. Например, $f(3{,}9999999) = 2, f(4) = 3$.

Для наглядности следует рассмотреть график этой функции на рисунке 2.

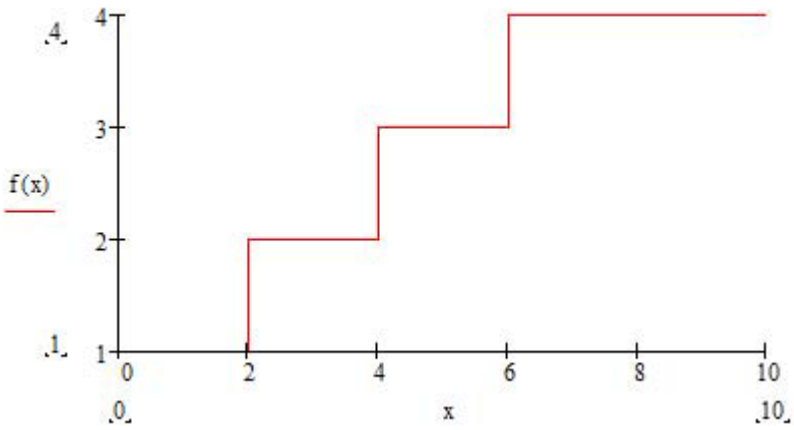

Рис. 2. График функции, заданной деревом решений

Теперь можно усложнить пример и рассмотреть функцию от двух переменных $f(x_1,x_2)$ на рисунке 3. Здесь $f(3,1) = 3, f(3,2) = 4, f(3,3) = 4$ и т.д.

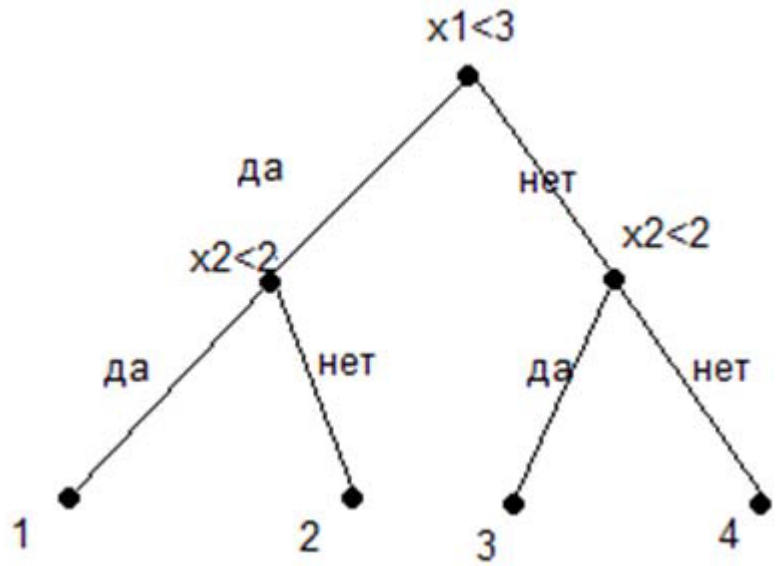

Рис. 3. Функция двух аргументов, заданная деревом решений.

Главное отличие дерева для функции $h_k(q,d)$, от рассмотренных выше в том, что условиями ветвления выступают не аргументы функции (q-запрос, d-документ), а некоторые функции от них $f_1(q,d)$, $f_2(q,d)$, именуемые признаками документа или свойствами (features). В качестве признаков могут выступать: количество точных вхождений запроса в документ, количество точных вхождений в заголовок документа, количество вхождений лемм слов запроса в документ, длина документа и т.д., в общем, все то, что Яндекс считает свойствами документа.

Теперь можно рассмотреть простое дерево для функции $h(q,d)$ на рисунке 4.

Для начала следует ввести следующие определения и обозначения:

- Лексическая единица (лексема) - слово как самостоятельная единица языка, рассматриваемая во всей совокупности своих форм и значений.
- Лемма L_k – каноническая форма лексемы k, k=1..M (M-размерность множества лемм коллекции документов).
- Встречаемость леммы в документе (внутренняя частота леммы $IF(L_k,d)$) – число вхождений леммы L_k в документ d, где =1..D (D-размерность множества документов коллекции).

$f_1(q,d)$ – количество вхождений лемм слов запроса в тег body, $f_2(q,d)$ – плотность лемм запроса в теге body, $f_3(q,d)$ - количество вхождений лемм запроса в заголовок

документа. Следует заметить, что в примерах значения функций возрастают слева на право. Это сделано для простоты восприятия, на самом деле они могут быть перемешаны как угодно. Так же следует уточнить, что все условия ветвления взяты случайно.

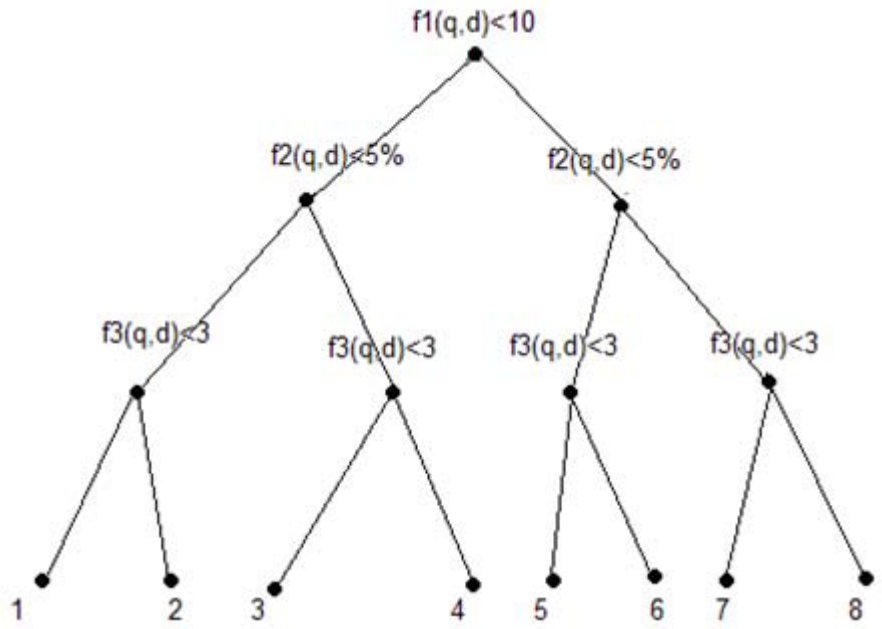

Рис. 4. Пример простейшего дерева для функции $h(q, d)$

Пусть некоторый документ d при запросе q имеет следующие свойства: $f_1(q, d) = 5, f_2(q, d) = 10\%$ $f_3(q, d) = 2$. Значение функции $h(q, d)$ для этого документа будет равно трём.

Реальные деревья решений, полученные при помощи алгоритма MatrixNet, обладают следующими свойствами:

- Глубина (т.е. число горизонтальных уровней или число ветвлений) дерева – число ограниченное. По некоторым данным это ограничение равно 10, т.е. на нижнем (10-м) уровне имеется 210 узлов, соответствующих значениям фунции $h(q, d)$.
- На каждом уровне ветвление происходит только по одному и тому же условию (собственно именно только такие примеры мы и рассматривали). Т.е. всего в каждой функции может использоваться не более 10 признаков документа.

Поскольку поисковая система работает с очень большими объемами информации, по каждому запросу ей нужно проверить признаки миллионов страниц, определить их релевантность и соответственно упорядочить. Так, чтобы сверху оказались более подходящие страницы. Чтобы проверить свойства всех страниц по очереди, нужно либо очень много серверов, которые могут быстро обработать информацию обо всех страницах,

либо очень много времени – а поиск должен работать быстро, иначе пользователи не дождутся результатов. Матрикснет позволяет проверить очень много факторов за короткое время и без существенного увеличения вычислительных мощностей.

Поиск ведется одновременно на тысячах серверов. Каждый сервер ищет по своей части индекса и формирует список самых лучших результатов. В него гарантированно попадают все самые релевантные запросу страницы.

Дальше из этих списков составляется один общий, и страницы, попавшие туда, упорядочиваются по формуле ранжирования – той самой длинной и сложной формуле, построенной с помощью Матрикснета, с учетом всех факторов и их комбинаций. Таким образом, наверху поисковой выдачи оказываются все самые релевантные сайты – и пользователь почти мгновенно получает ответ на свой вопрос.

Примерное устройство процесса ранжирования показано на рисунке 5.

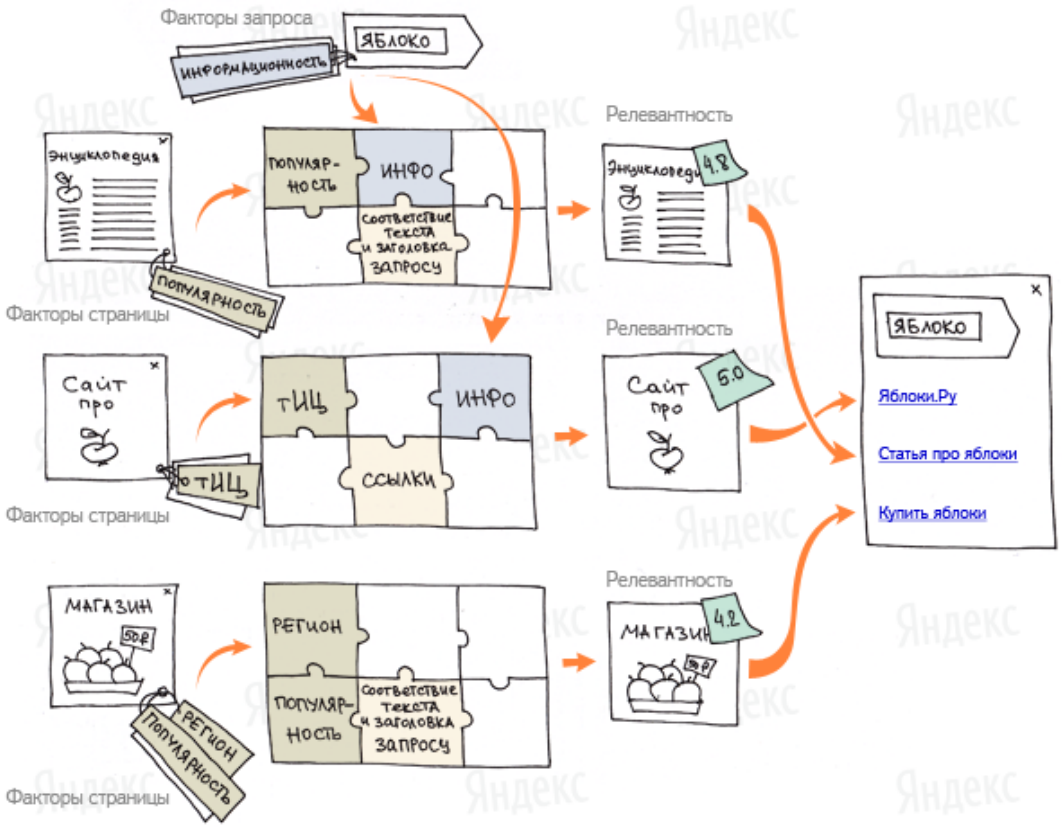

Рис. 5. Ранжирование

Заключение

Алгоритм ранжирования MatrixNet активно используется в поисковой системе Yandex. Матрикснет применялся в проекте по распознаванию личности человека (пола,

возраста, характера) из его поисковых запросов. Данный алгоритм может быть использован в различных видах деятельности, где требуется сортировка, выделение главного и второстепенного.

Список литературы

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ: пер. с англ. под ред. А. Шеня. М.: МЦНМО, 2002. 960 с.: 263 ил. [Cormen T., Leiserson C., Rivest R. Introduction to Algorithms. London: MIT Press, 1180 p.].
2. Зябрев И., Пожарков О., Пожаркова И. Моделирование алгоритма текстового ранжирования Яндекса при помощи MatrixNet. Режим доступа: http://www.altertrader.com/publications21.html (дата обращения 18.04.2015).
3. Зябрев И., Пожарков О. Статистические методы исследования алгоритмов текстового ранжирования поисковых систем. Режим доступа: http://www.altertrader.com/publications18.html (дата обращения 18.04.2015).
4. Зябрев И., Пожарков О. Спектральное оценивание лексических единиц в задачах лингвистического моделирования. Режим доступа: http://www.altertrader.com/publications16.html (дата обращения 18.04.2015).