Модели информационного поиска. PageRank Лекция № 3 курса «Алгоритмы для Интернета»

Юрий Лифшиц*

12 октября 2006 г.

Содержание

1. Модели информационного поиска

Модель информационного поиска имеет три ключевых аспекта.

1. Формат представления документа. Под документом мы будем понимать некий объект, содержащий информацию в зафиксированном виде. Документы могут содержать тексты на естественном или формализованном языке, изображения, звуковую информацию и т.д.

2. Формат представления запроса. Под запросом мы понимаем формализованный способ выражения информационных потребностей пользователя системы. Для этого используется язык поисковых запросов, синтаксис которых варьируется от системы к системе.

3. Функция соответствия документа запросу. Степень соответствия запроса и найденного документа (релевантность) — субъективное понятие, поскольку результаты поиска, уместные для одного пользователя, могут быть неуместными для другого.

*Законспектировал Максим Мозговой.

1.1. Булевская модель

Рассмотрим некоторый словарь $T = \{t_1, \ldots, t_n\}$, где t_i — термы. Термами могут быть слова, какие-то бессмысленные комбинации цифр, букв (почтовые индексы, телефонные номера и т.д.). Некоторые группы слов также считаются одним термом. Термы — не то же самое, что и слова. Например, Яндекс все падежи одного существительного может считать одним термом.

Документ — это некоторое подмножество словаря, набор термов: $D \subset T$, иначе говоря $D \in \{0,1\}^n$: на k-й позиции вектора стоит единица в том случае, когда k-е слово из словаря принадлежит документу, и ноль, если слово не принадлежит ему.

Запрос — булевская формула, например «t_5 OR t_7 AND NOT t_{12}», что означает, что необходимо найти документы, которые включают пятый или седьмой термы, но не включают двенадцатый.

Если формула выполнена на некотором документе, то будем считать, что этот документ соответствует запросу.

Такая модель иногда используется во внутренних корпоративных системах поиска, базах данных. Основным недостатком булевской модели является крайняя жесткость и непригодность для ранжирования. Если слово, указанное в запросе, присутствует в документе, то он считается найденным, в противном случае — не найденным. Не будет найден и документ, в котором встречаются только синонимы слова, указанного в запросе, в случае, когда само слово в документе не встречается.

1.2. Векторная модель

Имеется словарь из термов, как в булевской модели. Каждый документ представляется мультимножеством слов. Мультимножество — неупорядоченная коллекция, аналогичная множеству, но допускающая наличие в коллекции одновременно двух и более одинаковых значений. Каждый терм — это координата векторного пространства, говорящая о том, насколько «сильно» он входит в документ. Таким образом, каждый документ — это набор из n чисел. Определим матрицу M по формуле

$$M_{ij} = TF_{ij} \cdot IDF_i,$$

где TF_{ij} (Term Frequency, частота терма) — относительная доля слова i в документе j; IDF_i (Inversed Document Frequency) — величина, обратная количеству документов, содержащих слово i. Другими словами, это количество всех документов, поделенное на количество документов, которые содержат слово i.

Разберемся, в чем состоит «физический смысл» M_{ij}. Первый сомножитель показывает, насколько данное слово подходит данному документу. Для примера рассмотрим слово «Пьер» и произведение Л. Н. Толстого «Война и мир». Слово «Пьер» окажется достаточно часто встречающимся словом, и первый сомножитель (доля слова «Пьер» среди всех слов романа) будет велик. Теперь посмотрим на второй сомножитель. Его величина зависит от того, является ли слово общеупотребительным или редким: чем более редким окажется слово, тем больше будет сомножитель. За счет этого слово «Пьер» для романа будет более значимо, чем, например, слово «дворянин», даже если они встречались в тексте одинаковое число раз. Таким образом, в двух словах можно сказать так: M_{ij} — степень соответствия слова i документу j. Каждый документ представляется в этой матрице в виде столбца (j фиксировано, i меняется).

Для того чтобы подсчитать меру релевантности, представим сначала запрос в виде вектора с координатами 0 или 1: $Q =$ «t_3 AND t_5»$= \{0,0,1,0,1,0,\ldots,0\}$.

Каждый документ — набор таких координат: много нулевых координат (это те термы, которые не встречаются) и несколько ненулевых координат.

Мерой релевантности $R(Q, D_j)$ будем считать косинус угла между вектором запроса Q и документом D_j. Для того, чтобы подсчитать это число возьмем скалярное произведение векторов Q и D_j:

$$R(Q, D_j) = \cos\alpha = \frac{QD_j}{|Q||D_j|}.$$

Нормализация необходима для того, чтобы уравнять веса документов с разным количеством слов.

Пример выражения для меры релевантности документа D_3 и запроса $Q = \{0, 0, 1, 0, 1, 0, \ldots, 0\}$:

$$R(Q, D_3) = \frac{(TF_{33} \cdot IDF_3 + TF_{53} \cdot IDF_5)}{\sqrt{2}\,|D_3|}.$$

1.3. Вероятностная модель

В 1977 году Робертсон (Robertson) и Спарк-Джоунз (Sparck-Jones) обосновали и реализовали вероятностную модель. Релевантность в этой модели рассматривается как вероятность того, что данный документ может оказаться интересным пользователю. При этом подразумевается наличие уже существующего первоначального набора релевантных документов, выбранных пользователем или полученных автоматически при каком-нибудь упрощенном предположении. Вероятность оказаться релевантным для каждого следующего документа рассчитывается на основании соотношения встречаемости термов в релевантном наборе и в остальной части коллекции.

Документом будем считать множество слов без учета частоты встречаемости слова в документе. Можно также представить множество в виде обычного булевского вектора $D = \{d_1, \ldots, d_n\}$, где n — количество всех термов, а d_i может принимать значения из множества $\{0, 1\}$. Запросом будем считать множество слов.

Соответствие документа запросу будем строить следующим образом: представим себе, что для каждого фиксированного запроса Q_k у нас имеются распределения вероятностей на всех документах «быть релевантным» и «быть нерелевантным» запросу Q_k. Обозначается это соответственно как $P(R|Q_k, D)$ и $P(\overline{R}|Q_k, D)$. Тогда функцией соответствия будем считать отношение двух этих величин:

$$\frac{P(R|Q_k, D)}{P(\overline{R}|Q_k, D)}.$$

Теперь вспомним теорему Байеса:

$$P(a|b) = P(b|a)\frac{P(a)}{P(b)},$$

где

- $P(a)$ — априорная вероятность гипотезы a;
- $P(b)$ — вероятность наступления события b;
- $P(a|b)$ — вероятность гипотезы a при наступлении события b (апостериорная вероятность);
- $P(b|a)$ — вероятность наступления события b при истинности гипотезы a.

Применим ее для числителя и знаменателя дроби, стоящей в функции соответствия:

$$P(R|Q_k, D) = \frac{P(D|R, Q_k)P(R|Q_k)}{P(D|Q_k)};$$

$$P(\overline{R}|Q_k, D) = \frac{P(D|\overline{R}, Q_k)P(\overline{R}|Q_k)}{P(D|Q_k)};$$

$$\frac{P(R|Q_k, D)}{P(\overline{R}|Q_k, D)} = \frac{P(D|R, Q_k)P(R|Q_k)}{P(D|Q_k)} \frac{P(D|Q_k)}{P(D|\overline{R}, Q_k)P(\overline{R}|Q_k)} = \frac{P(R|Q_k)}{P(\overline{R}|Q_k)} \frac{P(D|R, Q_k)}{P(D|\overline{R}, Q_k)}.$$

Заметим, что первый сомножитель $\frac{P(R|Q_k)}{P(\overline{R}|Q_k)}$ одинаков для всех документов, так как в нем не фигурирует D, и мы его дальше можем не рассматривать. Предполагая независимость всех слов (это очень сильное, и на практике неверное предположение), второй сомножитель можно представить в виде произведения:

$$\frac{P(D|R, Q_k)}{P(D|\overline{R}, Q_k)} = \prod_{i=1}^{n} \frac{P(x_i = d_i|R, Q_k)}{P(x_i = d_i|\overline{R}, Q_k)},$$

где x_i — случайный документ, а d_i — число; $P(x_i = d_i|R, Q_k)$ — вероятность того, что i-й терм будет одновременно присутствовать или отсутствовать у случайного документа, релевантного нашему запросу, так же, как и в документе D. В произведении $\prod_{i=1}^{n} P(x_i = d_i|R, Q_k)$ будут именно те вероятности, которые описывают сам документ D, и оно будет равно $P(D|R, Q_k)$ в предположении независимости всех слов.

Пример. Чтобы понять, как работает полученная формула, запишем ее для следующих данных:

Запрос Q	кот	собака		
Документ D	кот	собака	слон	
Все известные термы	кот	собака	слон	бегемот

$$\frac{P(D|R,Q)}{P(D|\overline{R},Q)} = \frac{P(\text{содержится «кот»}|R,Q)}{P(\text{содержится «кот»}|\overline{R},Q)} \times \frac{P(\text{содержится «собака»}|R,Q)}{P(\text{содержится «собака»}|\overline{R},Q)} \times$$
$$\times \frac{P(\text{содержится «слон»}|R,Q)}{P(\text{содержится «слон»}|\overline{R},Q)} \times \frac{P(\text{не содержится «бегемот»}|R,Q)}{P(\text{не содержится «бегемот»}|\overline{R},Q)}.$$

Введем следующие обозначения:

$$p_{ik} = P(x_i = 1|R, Q_k);$$
$$q_{ik} = P(x_i = 1|\overline{R}, Q_k).$$

p_{ik} показывает, какова вероятность того, что в случайном релевантном k-му запросу документе i-е слово присутствует, а q_{ik} — соответственно, вероятность того, что в случайном нерелевантном k-му запросу документе i-е слово присутствует.

Перепишем произведение следующим образом:

$$\prod_{i=1}^{n} \frac{P(x_i = d_i|R, Q_k)}{P(x_i = d_i|\overline{R}, Q_k)} = \prod_{i \in Q_k} \frac{P(x_i = d_i|R, Q_k)}{P(x_i = d_i|\overline{R}, Q_k)} \prod_{i \notin Q_k} \frac{P(x_i = d_i|R, Q_k)}{P(x_i = d_i|\overline{R}, Q_k)}.$$

Второе произведение равно 1, так как мы считаем, что случайное слово, не входящее в запрос, равновероятно может встретиться нам как в документе, релевантном запросу, так и в документе, который запросу нерелевантен: $p_{ik} = q_{ik}$.

Работая далее с первым произведением, разбиваем его так:

$$\prod_{i \in Q_k} \frac{P(x_i = d_i|R, Q_k)}{P(x_i = d_i|\overline{R}, Q_k)} = \prod_{i \in Q_k \cap D} \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \prod_{i \in Q_k} \frac{1 - p_{ik}}{1 - q_{ik}}.$$

Второй сомножитель будет одинаков для всех документов. Забудем про него и возьмем логарифм от первого:

$$\log \left[\prod_{i \in Q_k \cap D} \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \right] = \sum_{i \in Q_k \cap D} \log \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})}.$$

Для применения полученной формулы нужно знать p_{ik} и q_{ik}. Пусть у нас уже есть некий набор текстов (учебная коллекция), про которые мы знаем, релевантны они запросу Q_k или нет. Тогда мы можем использовать формулы

$$p_{ik} = \frac{r_{ik}}{r_k};$$

$$q_{ik} = \frac{f_{ik} - r_{ik}}{f_k - r_k},$$

где f_k — общее число документов, r_k — число релевантных документов, r_{ik} — число релевантных документов, содержащих слово i, а f_{ik} — общее число документов со словом i.

2. PageRank

PageRank — это алгоритм, позволяющий оценить, насколько данная интернет-страница популярна, то есть это функция от интернет-страницы, которую можно сосчитать заранее.

Сергей Брин в 1998 году предложил следующую идею: определять рейтинг страницы через количество ведущих на нее ссылок и рейтинг ссылающихся страниц.

Google PageRank учитывает не все ссылки. Поисковая система отфильтровывает ссылки с искусственно созданных сайтов, специально предназначенных для скопления ссылок. Некоторые ссылки могут не только не учитываться, но и отрицательно сказаться на ранжировании ссылающегося сайта.

Также в Интернете используются и другие методы оценки популярности страницы:

- учет частоты обновляемости страницы (чем чаще страница обновляется, тем она «лучше»);
- учет посещаемости (чем больше пользователей посещают страницу, тем она «лучше»);
- учет регистрации в каталоге-спутнике поисковой системы.

2.1. Модель случайного блуждания

Рассмотрим сеть из вершин (страницы) и ориентированных ребер (ссылки). Будем моделировать передвижение пользователя по сети следующим образом: пользователь стартует в случайной вершине. С вероятностью ε пользователь переходит в случайную вершину, а с вероятностью $1-\varepsilon$ он переходит по одному из случайных исходящих ребер. На практике предполагают, что $\varepsilon \approx 0{,}15$.

Представим себе, что этот пользователь бродит так бесконечно долго. Для каждого k можно определить $PR_k(i)$ как вероятность оказаться в вершине i через k шагов. Пусть пользователь делает перемещение один раз в секунду. Тогда для каждой страницы существует какая-то вероятность, что пользователь окажется в ней через, например, миллиард секунд. Тогда предельная вероятность оказаться в i-й вершине и есть PageRank:

$$PR(i) = \lim_{k\to\infty} PR_k(i).$$

2.2. Основное уравнение PageRank

Пусть $T_1, \ldots, T_n$ — вершины, из которых идут ребра в i, $C(X)$ — обозначение для исходящей степени вершины X.

Поскольку мы стартуем в случайной вершине, то $PR_0(i) = 1/N$, где N — количество всех страниц.

По определению получаем следующее рекуррентное уравнение:

$$PR_k(i) = \varepsilon/N + (1-\varepsilon)\sum_{j=1}^{n} \frac{PR_{k-1}(T_j)}{C(T_j)}.$$

Перейдем к пределу и получим:

$$PR(i) = \varepsilon/N + (1-\varepsilon)\sum_{j=1}^{n} \frac{PR(T_j)}{C(T_j)}.$$

На практике вместо $PR(i)$ обычно используют $PR_{50}(i)$, вычисленное по итеративной формуле.

2.3. PageRank как собственный вектор матрицы всех ссылок

Имеет место достаточно красивый факт о том, что фактически PageRank — собственный вектор матрицы всех ссылок. Определим матрицу L следующим образом:

- если нет ребра из i в j, то $l_{ij} = \varepsilon/N$;

- если ребро есть, то $l_{ij} = \varepsilon/N + (1-\varepsilon) \cdot \frac{1}{C(i)}$.

Введем обозначения:

$$\overline{PR_k} = (PR_k(1), \ldots, PR_k(N))^T;$$
$$\overline{PR} = (PR(1), \ldots, PR(N))^T.$$

Тогда выполняются соотношения:

$$\overline{PR_k} = L^k \cdot \overline{PR_0};$$
$$\overline{PR} = L \cdot \overline{PR}.$$

Отсюда делаем вывод, что $\overline{PR}$ — собственный вектор матрицы всех ссылок L.

Задача

Докажите, что расстояние между векторами $\overline{PR_k(i)}$ и $\overline{PR(i)}$ экспоненциально быстро по k стремится к нулю.

Источники

[1] Sergey Brin and Larry Page. The Anatomy of a Search Engine
http://www-db.stanford.edu/pub/papers/google.pdf

[2] Илья Сегалович. Как работают поисковые системы
http://company.yandex.ru/articles/article10.html

[3] Amy Langville and Carl Meyer. Deeper Inside PageRank
http://meyer.math.ncsu.edu/Meyer/PS_Files/DeeperInsidePR.pdf

[4] Norbert Fuhr. Probabilistic Models in Information Retrieval
http://www.is.informatik.uni-duisburg.de/bib/fulltext/ir/Fuhr:92.pdf

[5] Страница курса
http://logic.pdmi.ras.ru/~yura/internet.html