Информационный поиск
Архитектура поисковых систем
PageRank

Лекция N 3 курса
"Алгоритмы для Интернета"

Юрий Лифшиц

ПОМИ РАН - СПбГУ ИТМО

Осень 2006

Авторы алгоритмов ссылочной популярности

В ноябре 1997 при запросе собственного названия только одна из четырех ведущих поисковых систем выдавала себя в первой десятке.

Брин и Пейдж, "Анатомия поисковых систем", 1998

Sergey Brin, Larry Page, Jon Kleinberg:

План лекции

1. Модели информационного поиска
   - Булевская модель
   - Векторная модель
   - Вероятностная модель
2. Архитектура поисковой системы
3. PageRank

Часть I

Формально, что такое документ?

Формально, что такое запрос?

При каком условии мы считаем, что документ соответствует запросу?

Булевская модель

Словарь: $T = \{t_1, \dots t_n\}$
Документ: $D \subset T$, иначе говоря $D \in \{0,1\}^n$
Запрос: $t_5 \quad \mathrm{OR} \quad t_7 \quad \mathrm{NOT} \quad t_{12}$

Соответствие:
Формула запроса должна быть выполнена на документе.

Недостатки модели?

Векторная модель

Снова коллекция документов, каждый из которых теперь является **мультимножеством** слов.

Определим матрицу M по формуле $M_{ij} = TF_{ij} \cdot IDF_i$, где:

- Частота терма TF_{ij} — относительная доля слова i в тексте j
- Обратная встречаемость в документах IDF_i — величина, обратная количеству документов, содержащих слово i

Физический смысл M_{ij} — степень соответствия слова i тексту j

Запрос: $t_3 \quad \mathrm{AND} \quad t_5$ (разрешаем только AND)

Релевантность в векторной модели

Запишем запрос в виде вектора:
$Q = "t_3 \quad \mathrm{AND} \quad t_5 - \{0,0,1,0,1,0,\dots,0\}$

Мерой релевантности будет **косинус** между запросом и документом:

$$R(Q,D) = \frac{Q \cdot D}{|D||Q|}$$

Вероятностная модель для чайников

Документ: множество слов (булевский вектор) $D = \{d_1, \dots, d_n\}$
Запрос: Q_k — тоже, но храним как множество

Соответствие:

- Зафиксируем запрос Q_k
- Пусть есть распределение вероятностей на всех текстах "быть релевантным запросу Q_k": обозначаем $P(R|Q_k, D)$
- Пусть есть распределение вероятностей на всех текстах "быть НЕрелевантным запросу Q_k": обозначаем $P(\bar{R}|Q_k, D)$
- Функцией соответствия будет их отношение (или логарифм этой дроби): $\frac{P(R|Q_k,D)}{P(\bar{R}|Q_k,D)}$

Вычисляем функцию соответствия

Воспользуемся теоремой Байеса ($P(a|b) = P(b|a)\frac{P(a)}{P(b)}$):

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

Первый сомножитель одинаков для всех документов. Предполагая независимость всех слов, второй сомножитель можно представить как произведение:

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

Вычисляем функцию соответствия II

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

Введем обозначения: $p_{ik} = P(x_i = 1|R, Q_k)$ и $q_{ik} = P(x_i = 1|\bar{R}, Q_k)$. Предположим, что для каждого слова i, не входящего в запрос,

$$p_{ik} = q_{ik}$$

Теперь мы можем переписать нашу дробь:

$$\prod_{i \in Q_k \cap D} \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \prod_{i \in Q_k} \frac{1 - p_{ik}}{1 - q_{ik}}$$

Вычисляем функцию соответствия III

$$\prod_{i \in Q_k \cap D} \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \prod_{i \in Q_k} \frac{1 - p_{ik}}{1 - q_{ik}}$$

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

$$\sum_{i \in Q_k \cap D} c_{ik}, \qquad \text{где } c_{ik} = \log \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})}$$

Подбор параметров

$$\sum_{i \in Q_k \cap D} c_{ik}, \qquad \text{где } c_{ik} = \log \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})}$$

Для использования полученной формулы нужно знать p_{ik} и q_{ik}.

Рецепт: пусть у нас уже есть некий набор текстов, про которые мы знаем, релевантны они запросу Q_k или нет. Тогда мы можем использовать формулы:

$$p_{ik} = \frac{r_i}{r} \quad \text{и} \quad q_{ik} = \frac{f_i - r_i}{f - r},$$

Угадываете смысл обозначений?

Подбор параметров II

$$p_{ik} = \frac{r_i}{r} \quad \text{и} \quad q_{ik} = \frac{f_i - r_i}{f - r},$$

Тут f — общее число документов, r — число релевантных документов, r_i число релевантных документов, содержащих слово i, а f_i — общее число документов со словом i.

Часть II

В каком формате запоминать интернет-страницы?

В какой структуре данных их хранить?

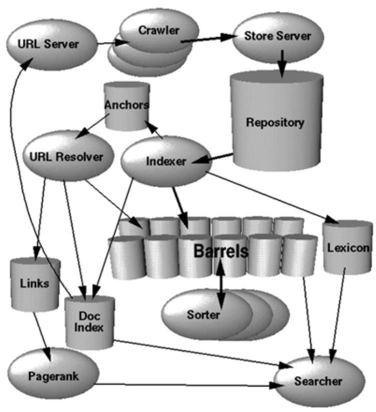
Как обрабатывать запрос пользователя?

Анатомия поисковой системы

Любая поисковая система содержит три базовые части:

- Робот (он же краулер, спайдер или индексатор)
- Базы данных
- Клиент (обработка запросов)

Схема из [Brin,Page, 1998]



Прямой и обратный индекс

Прямой индекс — записи отсортированы по документам

Номер документа
Отсортированный список слов
Для каждого слова: первые несколько вхождений, частота вхождений, формат вхождений

Обратный индекс — записи отсортированы по словам

Номер слова
Отсортированный список документов
Для каждого документа: информация о вхождении

Релевантность

Характеристики, влияющие на позицию в списке ответов?

- Наличие слов на сайте
- Частота слов
- Форматирование
- Близость слов друг к другу
- Количество ссылок с других страниц на данную
- Качество ссылок
- Соответствие тематик сайта и запроса
- Регистрация в каталоге, связанном с поисковой системой

Как работает клиент?

1. Разбирает запрос на слова
2. Переводит слова в их идентификаторы
3. Для каждого слова находит в обратном индексе список документов, его содержащих
4. Одновременно бежит по этим спискам, ища общий документ
5. Для каждого найденного документа вычисляет степень релевантности
6. Сортирует образовавшийся список по релевантности

Качество поиска

Как оценить качество поиска?

- **Полнота:** отношение количества найденных релевантных документов к общему количеству релевантных документов
- **Точность:** доля релевантных документов в общем количестве найденных документов
- **Benchmarks:** показатели системы на контрольных запросах и специальных коллекциях документов
- **Оценка экспертов**

Не пропустите, 23 ноября — приглашенная лекция Игоря Некрестьянова "Оценка качества интернет-поиска"

Часть III

Как определить ссылочную популярность страницы (PageRank)?

Как быстро вычислить приближение PageRank?

PageRank: постановка задачи

Хотим для каждой страницы сосчитать показатель ее "качества".

Идея [Брин, 1998]: Определить рейтинг страницы через количество ведущих на нее ссылок и рейтинг ссылающихся страниц

Другие методы:

Учет частоты обновляемости страницы
Учет посещаемости
Учет регистрации в каталоге-спутнике поисковой системы

Модель случайного блуждания

Сеть:

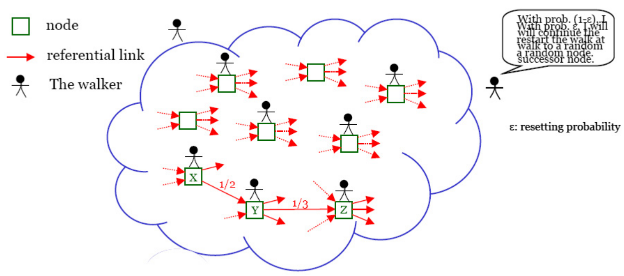
Вершины
Ориентированные ребра (ссылки)

Передвижение пользователей по сети

Стартуем в случайной вершине
С вероятностью ε переходим в *случайную* вершину
С вероятностью $1 - \varepsilon$ переходим по случайному исходящему ребру

Предельные вероятности

Для каждого k можно определить $PR_k(i)$ как вероятность оказаться в вершине i через k шагов
Факт: $\lim_{k \to \infty} PR_k(i) = PR(i)$, то есть для каждой вершины есть предельная вероятность находится именно в ней



Основное уравнение PageRank

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

Утверждение: $PR(i) = \varepsilon/N + (1-\varepsilon)\sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)}$

Кто может доказать?

По определению $PR_k(i)$ верно следующее:
$PR_0(i) = 1/N$
$PR_k(i) = \varepsilon/N + (1-\varepsilon)\sum_{i=1}^{n} \frac{PR_{k-1}(T_i)}{C(T_i)}$
Нужно просто перейти к пределу!

Практическое решение: вместо $PR(i)$ используют $PR_{50}(i)$, вычисленное по итеративной формуле.

PageRank как собственный вектор

Определим матрицу L:
Если нет ребра из i в j, то $l_{ij} := \varepsilon/N$
Если ребро есть, то $l_{ij} := \varepsilon/N + (1-\varepsilon)\cdot\frac{1}{C(j)}$

Введем обозначения:
$\overline{PR_k} = (PR_k(1), \dots, PR_k(N))$
$\overline{PR} = (PR(1), \dots, PR(N))$

Получаются соотношения:
$PR_k = L^k\, PR_0$
$PR = L\, PR$

Задачи

Докажите, что по расстояние между векторами $PR_k(i), PR(i)$ экспоненциально быстро (по k) стремится к нулю

Главные моменты

Сегодня мы узнали:

- Модели информационного поиска: булевская, векторная, вероятностная
- Поисковая система (1) скачивает и анализирует интернет-страницы, (2) записывает в базу и сортирует ее, (3) обрабатывает запросы, выводя лучшие страницы по функции релевантности
- PageRank — это предельная вероятность оказаться на web-сайте в результате случайного блуждания по ссылкам.

Вопросы?

Источники

Страница курса `http://logic.pdmi.ras.ru/~yura/internet.html`

Использованные материалы:

Sergey Brin and Larry page
The Anatomy of Search Engine
http://www-db.stanford.edu/pub/papers/google.pdf

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

Langville and Meyer
Deeper Inside PageRank
http://meyer.math.ncsu.edu/Meyer/PS_Files/DeeperInsidePR.pdf

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