Определения и формулировки по курсу «Алгоритмы для Интернет»

Составили: Надежда Поликарпова, Владимир Точилин и Борис Ярцев

1 Суффиксные деревья

1.1 Определения

- *Текст* — строка из n символов $t_1 \dots t_n$
- *Суффикс* — окончание $t_i \dots t_n$ текста T
- *Суффиксное дерево* — способ представления текста. Строится при помощи присоединения символа \$ к концу текста, взять все $n+1$ суффиксов, «подвесить» их за начала и «склеить» все ветки, идущие по одинаковым буквам. В листьях хранятся индексы начал суффиксов в тексте
- *Online-подход* — данные на вход подаются частями. Алгоритм, использующий online-подход, читает последоватьно данные и на каждом шаге получает готовое решение
- *Префикс* — любое начало текста $t_1 \dots t_i$
- *Неявное суффискное дерево* — это суффиксное дерево без символа \$ на конце. Некоторое суффиксы в нем заканчиваются не в листьях, а на ребрах и во внутренних вершинах
- *Фаза* — последовательность построения по суффиксному дереву для текста $t_1 \dots t_k$ суффискного дерева для текста $t_1 \dots t_k t_{k+1}$
- *Суффиксная стрелка* — ссылка, ведущая из внутренней вершины, соответствующей суффиксу префикса $t_i \dots t_k$, во внутреннюю вершину, соответствующую суффиксу префикса $t_{i+1} \dots t_k$
- *Хвост суффикса* — то место, в котором мы остановимся, если будем читать суффикс от корня дерева
- *Глубина вершины* — число ребер, по которым надо пройти из корня, чтобы попасть в вершину

1.2 Утверждения

- Суффиксные деревья могут применяться для поиска подстрок. Чтобы найти шаблон P в тексте T, нужно добавить символ \$ к тексту, построить для текста суффиксное дерево, а затем от корня суффиксного дерева последовательно спускаться по символам шаблона P вниз. Либо мы в какой-то момент не сможем спуститься, и тогда шаблона нет, либо индексы на листьях поддерева, в котором мы оказались дадут нам искомые индексы вхождения шаблона
- Суффиксные деревья могут применяться для поиска наибольшей общей подстроки. Для поиска наибольшей общей подстроки нужно «склеить» две подстроки символом %, добавить в конец \$, построить суффиксное дерево. После этого для каждой вершины отметить, есть ли среди ее листьев суффиксы, которые как содержат символ %, так и не содержат. Самая глубокая вершина с двумя типами суффиксов и даст нам наибольшую общую подстроку
- Суффиксная стрелка для суффикса префикса $t_i \dots t_k$ всегда будет заканчиваться в вершине

- При навигации в фазе i при помощи схемы «Вверх-Прыжок-Вниз» будет совершено не более $4i$ переходов
- После применения «пустого правила» — все дальнейшие продления пустые
- После того, как мы применили правило ответвления и создали новый лист, в следующих фазах к этому листу будет применяться только удлинение

2 Преобразование Берроуза-Вилера

TBD

3 Модель информационного поиска. PageRank

3.1 Определения

- *Модель информационного поиска* — формат представления документа + формат представления запроса + функция соответствия документа запросу (*релевантности*).
- В векторной модели *матрица соответветствия слов документам* $M_{ij} = TF_{ij} \cdot IDF_i$, 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.
- В векторной модели *мера релевантности* $R(Q, D_i) = \frac{QD_i}{|Q||D_i|}$.
- В вероятностной модели *мера релевантности*

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

  где $P(R \mid Q_k D)$ — вероятность того, что документ D релевантен запросу Q_k, $P(\overline{R} \mid Q_k D)$ — вероятность того, что неревантен.
- *PageRank* — это алгоритм, позволяющий оценить, насколько данная интернет-страница популярна
- В модели случайного блуждания $PR_k(i)$ — вероятность оказаться в вершине i через k шагов. *Page Rank* $PR(i) = \lim_{k \to \infty} PR_k(i)$.
- *Основное уравнение Page Rank.* Пусть $\{T_i\}$ — вершины, из которых идут ребра в i, $C(X)$ — исходящая степень вершины X.

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

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

- *Матрица всех ссылок* L: $l_{ij} = \varepsilon/N$, если нет ребра из i в j, иначе $l_{ij} = \varepsilon/N + (1 - \varepsilon) \cdot \frac{1}{C(i)}$.

3.2 Утверждения

- Меру релевантности в вероятностной модели можно посчитать как:

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

где p_{ik} — вероятность того, что в случайном релевантном k-му запросу документе i-е слово присутствует, q_{ik} — вероятность того, что в с лучайном нерелевантном k-му запросу документе i-е слово присутствует.

- Эти числа можно посчитать так:

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

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

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

- $\overline{PR} = (PR(1), \ldots, PR(N))$ есть собственный вектор матрицы всех ссылок.

4 Сложные сети

4.1 Определения

- *Сеть* — набор вершин и связей между ними — ребер.
- *(Не)ориентированная сеть* — сеть, ребра которой (не)ориентированные.
- *Двудольная сеть* содержит вершины двух различных типов, а ребра соединяют только вершины различающихся типов.
- *Социальные сети* — сети социальных взаимоотношений между людьми.
- *Информационные сети* — сети отношений между информационными объектами.
- *Технологические сети* показывают «физические» связи в нашем трехмерном мире.
- *Биологические сети* — сети внутри и между животными, разтениями, людьми.
- *Среднее кратчайшее расстояние*

$$l = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i \geq j} d_{ij},$$

где d_{ij} — кратчайшее расстояние от вершины i до вершины j, измеренное в числе ребер.

- *Среднее гармоническое кратчайшее расстояние*

$$l^{-1} = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i > j} d_{ij}^{-1}.$$

- *Транзитивность = кластеризация* — возможность разделить сеть на дизъюнктные группы таким образом, чтобы было много ребер внутри групп и мало ребер между группами.
- *Треугольник* — три вершины, связанные друг с другом.

- *Вилка* — вершина с двумя ребрами, идущими от этой вершины к двум другим
- *Коэффициент кластеризации*

$$C = \frac{3 \times \text{число треугольников в сети}}{\text{число «вилок»}} = \frac{6 \times \text{число треугольников в сети}}{\text{число путей длины 2}}.$$

- *Коэффициент кластеризации через усреднение по вершинам*

$$C' = \frac{1}{n} \sum_i \frac{\text{число треугольников с вершиной } i}{\text{число «вилок», центром которых является вершина } i}.$$

- *Степенное распределение* p_k — доля вершин в сети, имеющих степень k, вероятность того, что случайно выбранная вершина имеет степень k
- *Степенное распределение, усредненное по возрастающим интервалам* для каждого $2^t \geq k < 2^{t+1}$

$$p'_k = \frac{1}{2^t} \sum_{j=2^t}^{2^{t+1}-1} p_j.$$

- *Пуассоновское распределение* — $p_k = \frac{z^k e^{-z}}{k!}$, где z — константа.
- *Assortative mixing* — эффект более частого соединнения ребрами вершин одного типа, чем вершин разных типов
- *Коэффициент assortative mixing*

$$Q = \frac{[\sum_i P(i \mid i)] - 1}{N - 1},$$

  $1 \leq i \leq N$, N — число классов в сети, $P(j \mid i) = \frac{e_{ij}}{\sum_k e_{ij}}$ — условная вероятность того, что «мой друг представляет класс j при условии, что я сам представляю класс i»
- *Промежуточность вершины по кратчайшим путям (shortest-path betweenness)* — частота встречаемости данной вершины на кратчайших путях между другими вершинами.
- *Промежуточность случайного блуждания (random-walk betweenness)* — усредненное по всем парам вершин количество путей между ними, проходящих через заданную вершину.

5 Байесовские сети

TBD

6 Классификация текстов

TBD

7 Метод опорных векторов

TBD

8 Семантический Веб

8.1 Определения

- *Семантический Веб* — новая концепция развития Веба и сети Интернет, принятая и продвигаемая W3C (World Wide Web Consortium), снабжение Интернет страниц описаниями, которые понятны компьютерам.
- *Интернет* — всемирная система объединенных компьютерных сетей, построенная на использовании протоколов (TCP/IP) для связи и маршрутизации пакетов данных.
- *Веб* — глобальное информационное пространство, основанное на физической инфраструктуре Интернета и протоколе передачи данных HTTP.
- *Агент* — программа, работающая без непосредственного управления со стороны человека или другого постоянного контроля, созданная для достижения целей, поставленных перед ней пользователем.
- *Синтаксис* — набор правил построения фраз языка, позволяющий определить корректные предложения в этом языке. Основным инструментом синтаксиса является наличие правил проверки, позволяющих судить о том, удовлетворяет ли текст синтаксису или нет.
- *Семантика* — система правил истолкования отдельных языковых конструкций. Семантика определяет смысловое значение предложений языка.
- *RDF* (Resource Description Framework) — язык, отвечающий за синтаксис документов Семантического Веба. В нем широко используются ссылки на онтологии для определения смысла слов.
- *OWL* (Ontology Web Language) — язык описания онтологий.
- *Класс* — это концепция в онтологии. Классы являются основными блоками OWL и обычно образуют таксономическую иерархию (т.е. систему подкласс-надкласс). *Индивидуальные элементы* — это элементы классов. *Свойства* позволяют нам утверждать общие факты о членах классов и особые факты об индивидах. Характеристики свойств: симметричность, транзитивность, функциональность.

9 Оценка качества информационного поиска

TBD

10 Проектирование протоколов

10.1 Определения

- Задачи выбора функций оплаты изучаются в разделе *Mechanism Design*, а способы их вычисления — в разделе *Algorithmic Mechanism Design*.
- В задаче о кратчайшем пути надо платить $D_{AB}^{a_k=\infty}(\overline{a}) - D_{AB}^{a_k=0}(\overline{a})$, где $D_{AB}^{a_k=\infty}(\overline{a})$ — длина минимального пути в предположении $a_k = \infty$, а $D_{AB}^{a_k=0}(\overline{a})$ — в предположении $a_k = 0$.
- В *аукционе Викри* победитель — это тот, кто назвал максимальную цену, но платит он по второй объявленной цене.

10.2 Утверждения

- В задаче о кратчайшем пути при описанной функции оплаты: Зафиксируем ходы всех участников, кроме k-го. Тогда любая попытка соврать k-го агента не принесет ему большего выигрыша, чем честное признание.
- При k подрядчиках жадное расписание (задание отправляется к тому, кто объявил наименьшее время его исполнения) не более чем в k раз проигрывает любому другому.
- Следующая формула всегда удовлетворяет свойству «честность — лучшая политика», то есть для такой функции оплаты каждому агенту вранье не приносит никакой пользы:

$$p_i(\overline{a}, o) = \sum_{j \neq i} v_j(t_j, o) + h_i(t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n),$$

  где $v_j(t_j, o)$ — функция внутреннего выигрыша агентов.

11 Открытые проблемы

11.1 Определения

- *Критерии выбора задачи —*
  - непосредственная связь с разработкой новых технологий;
  - знакомство с областями приложений;
  - взаимосвязь нескольких научных направлений;
  - сводимость других задач к данной;
  - возможность проверить правильность разрабатываемого решения;
  - новизна.
- *Формат представления задачи*
  - область (технология) для использования;
  - формализация описания;
  - вовлеченные научные направления;
  - близкие классические результаты;
  - план исследований.
- В задаче Tag Propagation: формула *распространения метки*:

$$T_k(i) = T_{k-1}(i) + \alpha \sum_{j \in J(i)} \frac{T_{k-1}(j) - T_{k-2}(j)}{Out(j)},$$

  где $J(i)$ — множество вершин, ссылающихся на i, а $Out(j)$ — общее количество исходящих ссылок из j.
- *Алгоритм Фитча* — позволяет найти оптимальное дерево с минимальным количеством мутаций (изменений от предков к потомкам).