# Открытые проблемы по веб-алгоритмам

Лекция N 11 курса
"Алгоритмы для Интернета"

Юрий Лифшиц

ПОМИ РАН - СПбГУ ИТМО

Осень 2006



Не говорите мне, что эта проблема сложна. Будь она проста, не было бы проблемы.

*Фердинанд Фош*

Вы не можете решить проблему, пока не признаете, что она у вас есть.

*Харви Маккей*

Старайся создавать такие проблемы, решение которых известно только тебе.

*Принцип Берка*



## План лекции

1. Как поставить хорошую задачу?
2. Задача 1: Крупномасштабная фильтрация
3. Задача 2: Распространение меток
4. Задача 3: Выявление структур



## Введение

Какую задачу следует считать хорошей, интересной, важной?

В каком формате мы будем представлять задачи?



## Мои критерии

- Непосредственная связь с разработкой новых технологий
- Знакомство с областью приложений
- Взаимосвязь нескольких научных направлений
- Другие задачи сводятся к данной
- Легко проверить, работает ли разрабатываемое решение
- Новизна (часто сопровождается отсутствием хорошей формальной постановки)

**Я не использую:**

- Техническую сложность
- Известность и возраст задачи
- Известность автора задачи

Ваши критерии для выбора темы курсовой/дипломной работы?



## Анкета задачи

1. Область (технология) для использования?
2. Пример строгой постановки?
3. Вовлеченные научные направления?
4. Близкий классический результат?
5. План исследований?
6. Ваши конструктивные идеи?



## Отказ от ответственности

Эта лекция является **экспериментом**:

1. Задачи придуманы после поверхностного знакомства с веб исследованиями
2. В ближайшее время будет проведен поиск по литературе и сопоставление с известными результатами

Таким образом, представляемые задачи могут быть уже поставлены и даже решены!



ЗАДАЧА 1:

## Крупномасштабная фильтрация

Large-scale filtering

Как построить быстрый алгоритм для персонального сбора новостей?

## 1.1. Технологическая задача

Персональный сбор новостей:
У каждого пользователя есть набор предпочтений: конкретные авторы, ключевые слова, метки (темы), пороги популярности, ссылки на предпочтения друзей

Каждое новостное сообщение имеет описание: текст, голоса, рекомендации, метки, репутацию автора, комментарии

**Задача фильтрации:**
Для каждого пользователя определить десять наиболее интересных ему сообщений

**Примеры:**Google News, Google Reader, Yandex Lenta, Livejournal Friends, ...



## 1.2. Формализация

- Опишем систему предпочтений каждого участника **красным** нормализованным вектором (точкой на сфере) в $n$-мерном пространстве признаков
- каждое описание новости будет нормализованным **синим** вектором в том же пространстве
- Будем использовать косинусную меру (скалярное произведение) для соответствия новостей пользователям
- Вычислительная задача: провести (пред)вычисления на **синих** векторах так,чтобы для каждого входящего **красного** вектора можно было бы быстро определить десять ближайших **синих** соседей



## 1.3. Вовлеченные направления

- Классификация текстов, алгоритмы поиска ближайших соседей
- Вычислительная геометрия
- Структуры данных
- Архивирование (редкие векторы)
- Линейная алгебра (сингулярное разложение)
- Модели распределенных вычислений
- Могут ли помочь квантовые алгоритмы?



## 1.4. Алгоритм Клейнберга (1/2)

- Есть $n$ точек в $d$-мерном пространстве
- Нужно записать их в структуру данных
- С вероятностью $1 - \varepsilon$ быстро находить ближайшую точку к любой данной



## 1.4. Алгоритм Клейнберга (2/2)

**Алгоритм по шагам:**

1. Выбираем $k$ случайных нормализованных "контрольных" вектора
2. Составляем скалярные произведения между векторами из коллекции и контрольными
3. Запоминаем ближайшую точку для каждой **системы согласованных интервалов**
4. Обработка запроса: берем входную точку, определяем соответствующую ей систему интервалов, смотри в структуру данных



## 1.5. План исследований

1. Построить быстрый алгоритм фильтрации всех новостей для всех пользователей
2. Найти наиболее эффективные структуры данных для хранения пользователей/новостей
3. Изучить динамические аспекты: описания новостей и пользователей быстро меняются
4. Разработать систему предотвращения спама в системе персонального сбора новостей
5. Как уравнять в правах свежие и старые новости?



## 1.5. Ваши конструктивные идеи

Какие вопросы необходимо решить в представленной модели?

Как сделать формализацию лучше?



ЗАДАЧА 2

**Распространение меток**
Tag Propagation

Как распространить начальное распределение ключевых слов на весь Веб?

## 2.1. Технологическая задача

**Классификация веба:**
Люди используют миллионы меток (ключевых слов)
Веб состоит из миллиардов страниц
**Разреженная** коллекция пар (вебсайт,метка)

**Цель:**
Построить быстрый алгоритм подбора меток для произвольной страницы

**Приложения:**
Настройка рекламных объявлений
Аннотирование результатов поисковых систем
Автоматические каталоги



## 2.2. Формализация

- Есть граф ссылок
- Зафиксируем метку. Пусть для исходно-помеченных страниц $T_0(i) = 1$, для остальных $T_0(i) = 0$
- Возьмем предел по рекурсивным соотношениям:

$$T_k(i) = T_{k-1}(i) + \alpha \sum_{j \text{ links to } i} \frac{T_{k-1}(j) - T_{k-2}(j)}{Out(j)}$$

- Вычислительная задача: найти алгоритм (пред)вычислений по исходному распределению меток, с помощью которого можно быстро находить десять меток с наибольшим рангом для произвольного запрашиваемого сайта



## 2.3. Вовлеченные направления

- Структуры данных
- Архивирование (разреженные множества)
- Численные методы (скорость сходимости)
- Алгоритмы для PageRank



## 2.4. PageRank vs. TagRank

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

vs.

$$T_k(i) = T_{k-1}(i) + \alpha \sum_{j \text{ links to } i} \frac{T_{k-1}(j) - T_{k-2}(j)}{Out(j)}$$



## 2.5. План исследований

1. Определить формулы распространения меток
2. Построить алгоритм быстрой предварительной обработки учебной коллекции и on-line аннотирования



## 1.5. Ваши конструктивные идеи

Какие вопросы необходимо решить в представленной модели?

Как сделать формализацию лучше?



ЗАДАЧА 3

**Выявление структур**
Structure Discovery

Посмотрим на ключевые слова (метки), которые мы используем. Как подобрать наиболее точную систему отношений (иерархию?) между ними?



## 3.1. Технологическая задача

Можно собрать огромные коллекции данных:
истории покупок и поисковых запросов,
граф звонков и RSS подписок, социальные сети.
КАК ПОЛУЧИТЬ ПОЛЬЗУ ОТ ЭТИХ ДАННЫХ?

Пример: **обнаружение иерархии**
У нас есть некоторая **фолксономия**
Как вычислить "оптимальную" иерархию меток?

**Приложения:**
Улучшение визуализации данных
упрощение навигации
Решение проблемы синонимов

## 3.2. Формализация

- Каждая метка характеризуется множеством соответствующих ей сайтов
- Мы хотим построить "оптимальное" **AND-OR** дерево меток
- Оптимальное = минимальное отклонение от идеала
- Идеал: дети OR-вершины не должны пересекаться, множество родителя содержит множества всех детей, и т.д.



## 3.3. Вовлеченные направления

- Вычислительная биология (алгоритмы филогенетики)
- Приближенные алгоритмы
- Добыча данных (data mining, web mining)



## 3.4. Алгоритм Фитча (1/2)

- Есть бинарное дерево
- На листьях ДНК
- Нужно найти правдоподобные ДНК внутренних вершин



## 3.4. Алгоритм Фитча (2/2)

**Алгоритм по шагам:**

1. Отдельно работаем для каждой позиции
2. Для каждой внутренней вершины составим список разумных кандидатов $S_v$
3. Проход снизу вверх: если $w$ — родитель $u$ и $v$, и $S_u \cap S_v =$, то $S_w = S_u \cup S_v$, иначе $S_w = S_u \cap S_v$
4. Проход сверху вниз: выбираем символ для корня, дальше берем символ родителя если он входит в множество ребенка, иначе произвольный символ из множества ребенка



## 3.5. План исследований

1. Выбрать формат представления метки и определить критерии идеальной иерархии меток
2. Найти быстрый алгоритм построения оптимальной иерархии
3. Изучить взаимосвязи с алгоритмами филогенетики.



## 3.5. Ваши конструктивные идеи

Какие вопросы необходимо решить в представленной модели?

Как сделать формализацию лучше?



## Голосование

Мы обсудили три задачи. Какая из них вам лично кажется наиболее привлекательной?

1. Крупномасштабная фильтрация
2. Распространение меток
3. Выявление структуры



## Задача на дом

Пусть $|v| < |u|$, докажите что с вероятностью не менее $\frac{1}{2}$ для случайного вектора $r$ выполнено $r \cdot v < r \cdot u$

## Основные моменты

**Сегодня мы узнали:**

- Технологические задачи: персональный сбор новостей, использование больших объемов данных, автоматическое аннотирование
- Ключевая алгоритмическая проблематика: алгоритмы на миллиардах объектов: эффективные структуры данных и быстрая обработка запросов. Нужно ускорить наивные "каждый-каждый" алгоритмы
- Следующий шаг: (1) обзор литературы, (2) формализации и модели, (3) публичное обсуждение

Спасибо! Вопросы?



## Источники

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

Использованные материалы:



Jon Kleinberg
Two algorithms for nearest-neighbor search in high dimensions
http://citeseer.ist.psu.edu/kleinberg97two.html


Ron Shamir
Phylogenetics
http://www.cs.tau.ac.il/~rshamir/algmb/01/scribe08/lec08.pdf


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