

Алгоритм прогнозирования структуры локально-оптимальных моделей.

Михаил Лепехин

МФТИ ФИВТ

lepehin.mn@phystech.edu

21 марта 2019 г.

Цель исследования

Цель

На основе генетического алгоритма, а также с использованием нейронных сетей построить метод для предсказания структуры нелинейной ранжирующей функции и сравнить полученные результаты с результатами сообщества TREC.

Проблема

Предсказание структуры нелинейной модели по имеющимся данным - тяжёлая задача.

Решение

Использование генетического алгоритма для построения ранжирующей функции в виде дерева с покрашенными вершинами с разбиением выборки на кластеры.

Существующие методы

Простые методы

- Salton, Gerard and McGill, Michael J. Introduction to Modern Information Retrieval // McGraw- Hill, Inc. New York, NY, USA, 1986

Перебор суперпозиций

- P. Goswami, S. Moura, E. Gaussier, M.-R. Amini, F. Maes Exploring the space of ir functions // ECIR'14, 2014, pp. 372–384.

Использование генетического алгоритма

- Fan, Weiguo and Gordon, Michael D. and Pathak, Praveen Personalization of Search Engine Services for Effective Retrieval and Knowledge Management // In Proceedings of the twenty first international conference on I

- A.S. Kulunchakov, V.V. Strijov Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection, Expert Systems with Applications: An International Journal (2017).

Постановка задачи порождения модели

Дано

Коллекция текстовых документов C , состоящая из документов $\{d_i\}_{i=1}^{|C|}$ и множество поисковых запросов $Q = \{q_j\}_{j=1}^{|Q|}$.

Часть документов оценена экспертами. Таким образом задана функция $r(d, q) \rightarrow \{0, 1\}$, где оценка 1 ставится в случае релевантности документа d запросу q .

Обозначения

$\text{count}(w, C)$ - количество документов $d \in C$, в которые входит слово w ,

$\text{freq}(w, d)$ - количество вхождений слова w в документ d ,

size_{avg} - среднее количество слов в документах коллекции,

$\text{size}(d)$ - количество слов в документе d .

Постановка задачи порождения модели

Рассматриваемые характеристики

$$\text{idf}(w, C) := \frac{\text{count}(w, C)}{|C|}$$

$$\text{tf}(w, d, C) := \text{freq}(w, d) * \log \left(1 + \frac{\text{size}_{\text{avg}}}{\text{size}(d)} \right)$$

Пусть $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ - функция 2 переменных. Тогда её значение на паре (d, q) определяется как сумма её значений на парах (d, w) , где $w \in q$ - слово из запроса:

$$f(d, q) := \sum_{w \in q} f(\text{tf}(w, d), \text{idf}(w))$$

$$\text{MAP}(f, C, Q) = \frac{1}{|Q|} \sum_{q \in Q} \text{AvgP}(f, q),$$

Постановка задачи порождения модели

Метрика качества ранжирующей функции

$$\text{MAP}(f, C, Q) = \frac{1}{|Q|} \sum_{q \in Q} \text{AvgP}(f, q),$$

где

$$\text{AvgP}(f, q) = \frac{\sum_{k=1}^{|C_q|} \text{Prec}(k) \times r(q, k)}{\sum_{k=1}^{|C_q|} r(q, k)},$$

$$\text{Prec}(k) = \frac{\sum_{s=1}^k r(q, s)}{k}$$

Постановка задачи порождения модели

Пространство исследуемых функций

В качестве математических примитивов $h(x, y)$ будем использовать функции \sqrt{x} , $x + y$, $x - y$, $x * y$, x / y , $\log x$, e^x . Будем исследовать пространство всех суперпозиций этих примитивов. Обозначим его \mathcal{F} .

Оптимизируемая функция

$$f^* = \arg \max_{f \in \mathcal{F}} \text{MAP}(f, C, Q) - R(f),$$

где R - регуляризатор, штрафующий за структурную сложность порождаемой суперпозиции.

Постановка задачи на кластерах документов

Обозначим L множество всех рассматриваемых слов в документах, $|L| = n$.

Определим $tf - idf$ для всей коллекции документов. Рассмотрим отображение $V : C \rightarrow \mathbb{R}^n$, в котором каждому документу сопоставляется вектор $tf - idf$ представления всех слов в нем. Для кластеризации документов будем использовать их представление в пространстве \mathbb{R}^n . Расстояние между документами считаем при помощи стандартной евклидовой метрики.

Получаем множество кластеров D , $|D| = m$. Разбиение на кластеры выполним с использованием метода k ближайших соседей. Для каждого кластера при помощи генетического алгоритма построим семейство ранжирующих функций $F_{d_i}^* = \{f_i^1, \dots, f_i^n\}$. В каждом семействе i выделим наилучшую по описанной выше метрике ранжирующую функцию $f_i^* \in F_{d_i}$.

Определим ранжирующую функцию на кластерах:

Базовый алгоритм

Используется генетический алгоритм со следующими процедурами:

- мутация — замена произвольной вершины на заново сгенерированную.
- скрещивание (crossover) — обмен местами двух произвольных вершин деревьев.

Регуляризация

$$R(f) = ||f||^2,$$

где $||f||$ - число вершин в дереве функции f .

Цель эксперимента

Цель эксперимента

Проверить работоспособность метода. Улучшить результаты по сравнению с работами сообщества TREC.

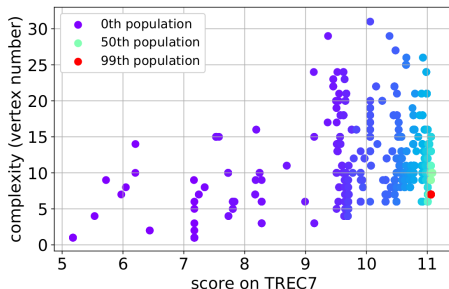
Данные

Коллекция TREC (датасеты 5-8).

<https://trec.nist.gov/data.html>

Результаты эксперимента

Зависимость сложности модели от значения целевой метрики.



Результаты эксперимента

Результаты при сравнении на корпусах TREC-5, TREC-6, TREC-7.

Superposition	TREC-5	TREC-6	TREC-7
Функции сообщества			
f_1	8.785	13.715	10.038
f_2	8.908	13.615	9.905
f_3	8.908	13.615	9.905
Найденные наилучшие функции			
h_5^*	9.537	13.762	10.584
h_6^*	8.903	13.967	10.771
h_7^*	8.526	13.424	11.060

Заключение

- Показана работоспособность метода
- Для каждого корпуса была получена функция наилучшим образом ранжирующая документы для данного запроса

Планируется

Улучшить текущий метод путём деления набора коллекций на 3 части:

- На первой части генерируется ансамбль моделей
- На второй части подбираются оптимальные веса для данных моделей
- Последняя часть используется для проверки итогового качества.