Ранжирующие модели для систем информационного поиска. Прогнозирование структуры локально-оптимальных моделей.

Поповкин Андрей Алексеевич
Романенко Илья Игоревич

Московский физико-технический институт

Курс: Численные методы обучения по прецедентам (практика, В. В. Стрижов)/Группа 594, весна 2018

Цель исследования

Цель исследования

Исследовать возможность порождения ранжирующий функции при помощи генетического алгоритма и при помощи нейронный сети, сравнить полученные результаты с результатами сообщества.

Проблема

Сложно исследовать пространство существенно нелинейных функций.

Список литературы

1. *Kulunchakov A. S., Strijov V. V.* Generation of simple structured IR functions by genetic algorithm without stagnation // http://strijov.com/papers/Kulunchakov2014RankingBySimpleFun.pdf
2. *Salton, Gerard and McGill, Michael J.* Introduction to Modern Information Retrieval // McGraw-Hill, Inc., New York, NY, USA, 1986
3. *Gordon, M.* Probabilistic and Genetic Algorithms in Document Retrieval // Commun. ACM 31, 10 (October 1988), 1208-1218.

Постановка задачи

Пусть $\mathbf{C} = \{d_i\}$ - коллекция документов, $\mathbf{Q}$ - пользовательских запросов, $q = \{w_j\}$. Определена функция релевантности $r(d, q) \to \{0, 1\}$.
Рассматриваются характеристики пары документ-слово: $(d, w, \mathbf{C}) \to (\text{tf}, \text{idf})$.

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

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

$\mathcal{T}$ - множество суперпозиций функций от tf, idf. Будем аппроксимировать функцию $r(d, q)$, как функцию $f(d, q) = \sum_{w \in d} f'(\text{tf}, \text{idf})$,где $f' \in \mathcal{T}$.

$$f^* = argmax_{f \in \mathcal{T}} \left(\text{MAP}(f, \mathbf{C}, \mathbf{Q}) - \| f \|^2\right)$$

Функционал качества

Качеством аппроксимационной функции будем считать MAP.

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

$$\mathrm{AvgP}(f, q, \mathbf{C}) = \frac{\sum_{i=0}^{|C_q|} \mathrm{PrefSum}(r(d_i, q), k) \cdot r(d_i, q)}{\sum_{d \in C_q} r(d)}$$

Цели эксперимента

Цель эксперимента

Получение результатов, сравнимых с предыдущими работами в этой сфере. Улучшение этих результатов.

Используемые данные

Коллекция TREC (датасеты 5-8).
`https://trec.nist.gov/data.html`

Вычислительный эксперимент

Используется генетический алгоритм с регуляризацией по числу узлов в дереве.
Со следующими процедурами:

1. мутация — замена произвольной вершины на заново сгенерированную.
2. crossover — обмен местами двух произвольных вершин деревьев.

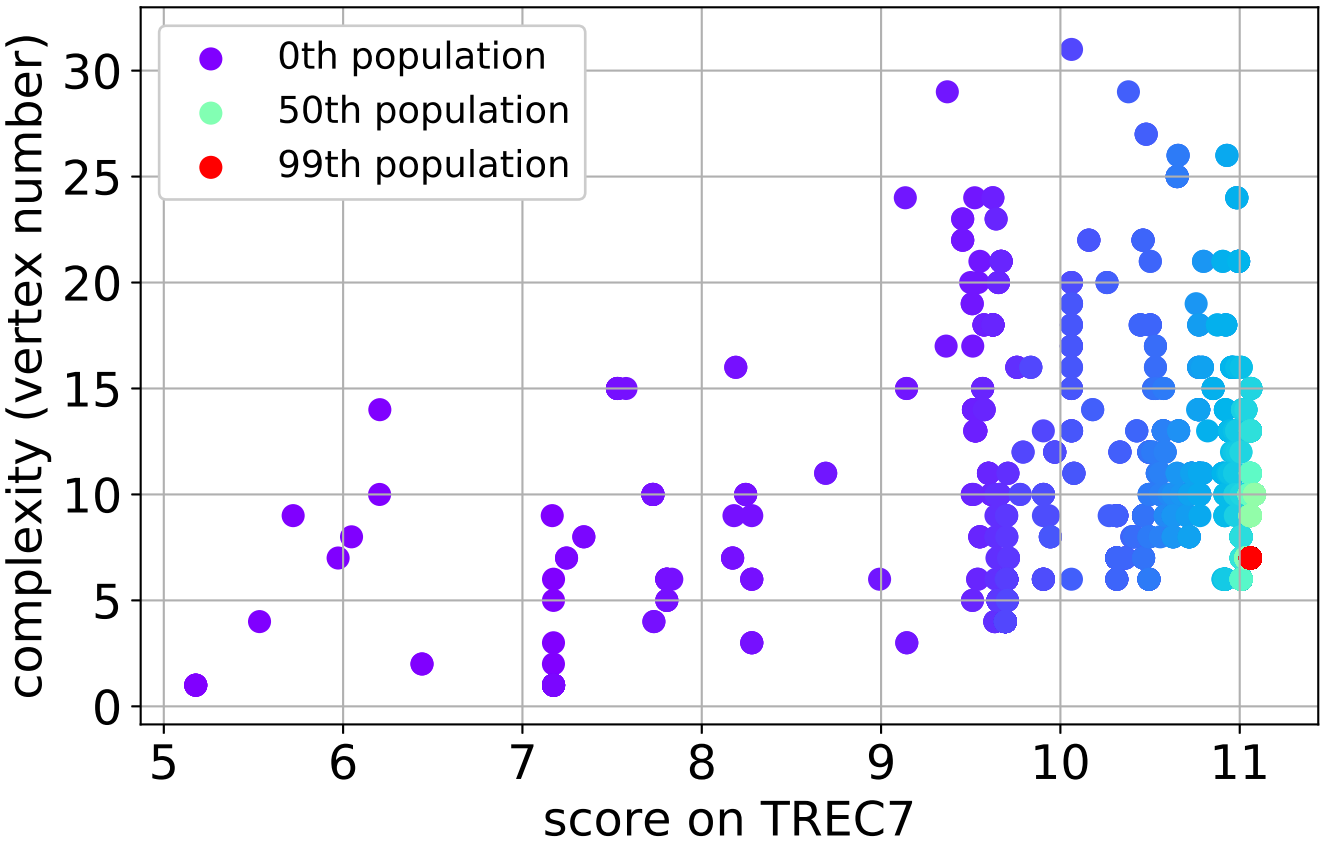
Вычислительный эксперимент

Используется генетический алгоритм с регуляризацией по числу узлов в дереве.
Со следующими процедурами:

1. мутация — замена произвольной вершины на заново сгенерированную.
2. crossover — обмен местами двух произвольных вершин деревьев.

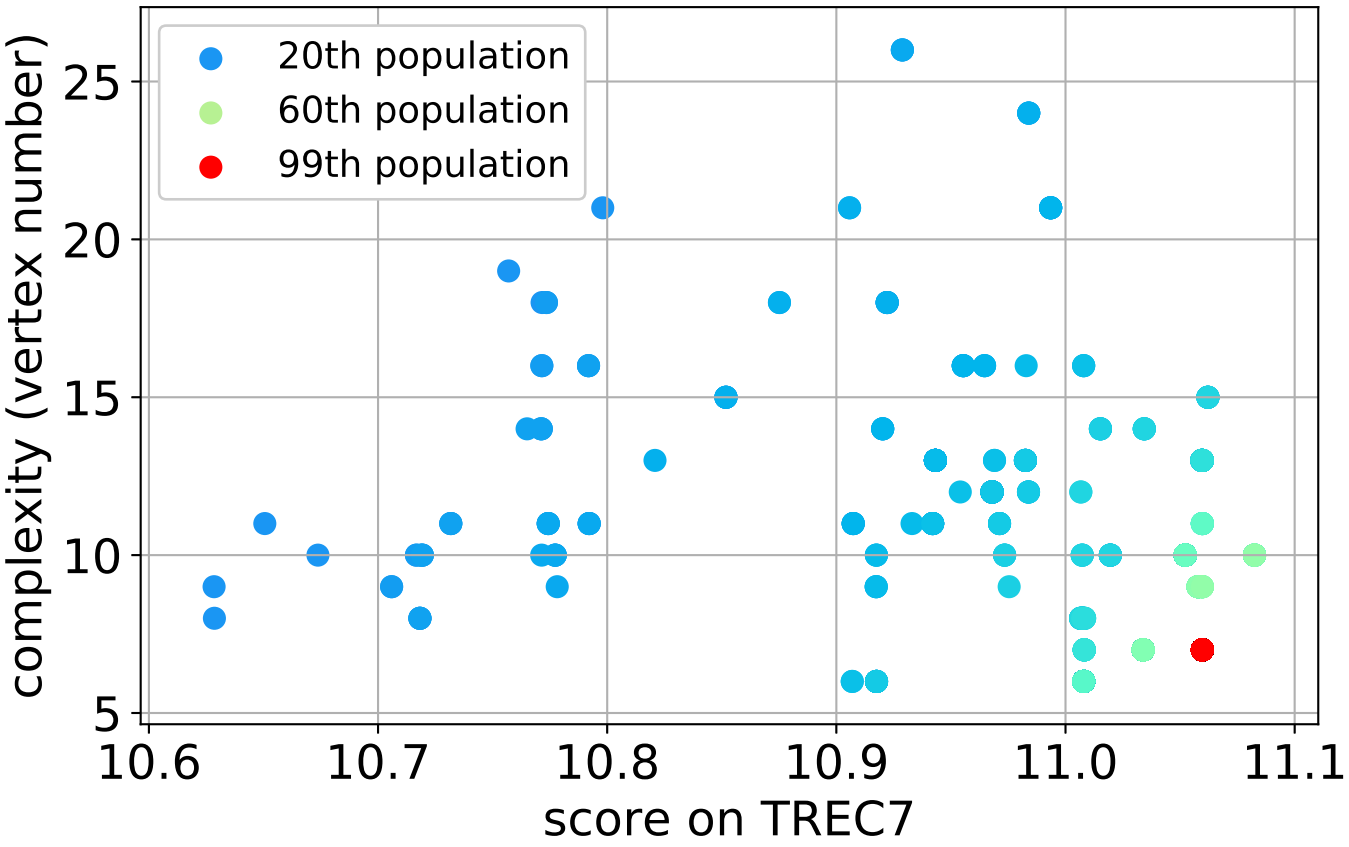
Вычислительный эксперимент

Сложность моделей в зависимости от целевой метрики.



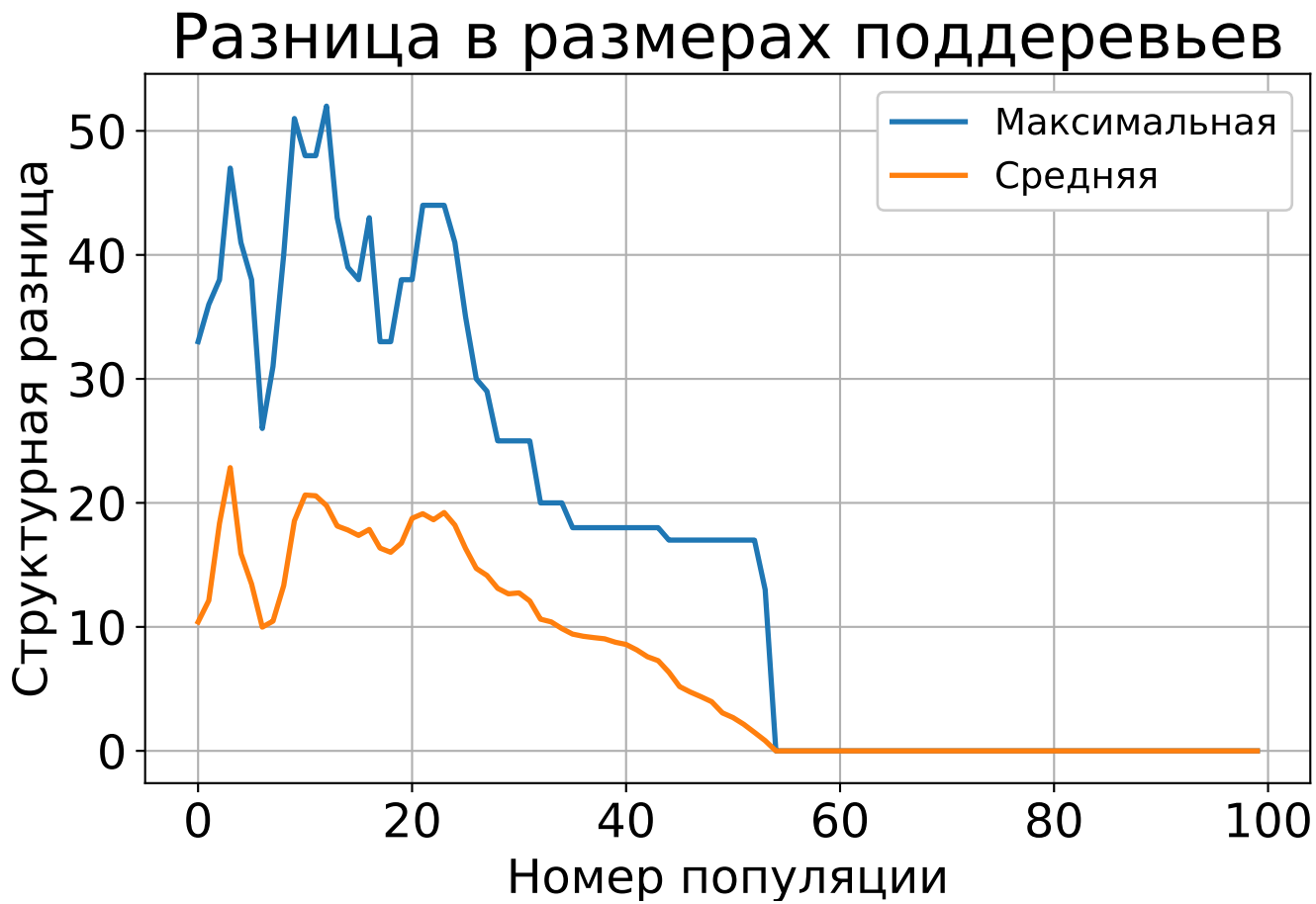
Вычислительный эксперимент

Сложность моделей в зависимости от целевой метрики, начиная с 20-ого поколения



Вычислительный эксперимент

Метрика максимального общего поддерева для определения стагнации.



Результаты эксперимента

Экспертные функции:

$$f_1 = e^{\sqrt{\log(1+\frac{\text{tf}}{\text{idf}})}}$$

$$f_2 = \sqrt[4]{\frac{\text{tf}}{\text{idf}}}$$

$$f_3 = \sqrt{\text{idf} + \sqrt{\frac{\text{tf}}{\text{idf}}}}$$

Найденные наилучшие функции :

$$h_5^* = \log\Big(1 + \frac{\log(1 + \log(1 + \log(1 + \sqrt{\log(1 + \text{tf}) - \sqrt{\text{idf}}})))}{2 \cdot \text{idf}}\Big)$$

$$h_6^* = \sqrt{\frac{\sqrt[4]{\text{tf}}}{2 \cdot \text{idf}}}$$

$$h_7^* = \frac{\sqrt[8]{\log(1 + \text{tf})}}{\text{idf}}$$

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

1. Для каждого корпуса получили функцию наилучшим образом ранжирующую документы для данного запроса.
2. Улучшили существующие результаты сообщества.
3. Весь код доступен в репозитории на GitHub