

Отличие двух графов

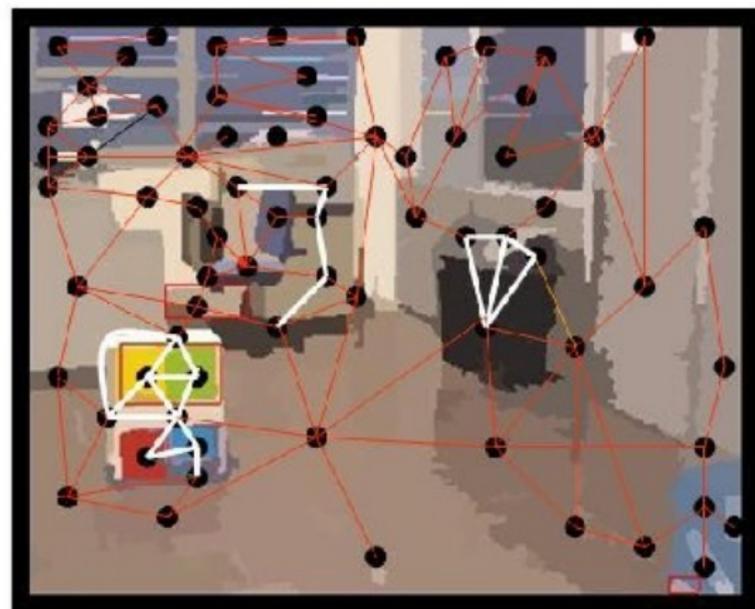
Бзикадзе Александр Важевич
группа 344

Научный руководитель
к.т.н., доц. Литвинов Юрий Викторович

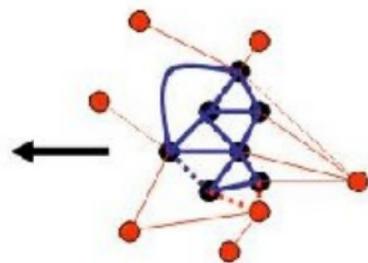
СПбГУ

22 мая 2018 г.

Применение (1)



Recognition

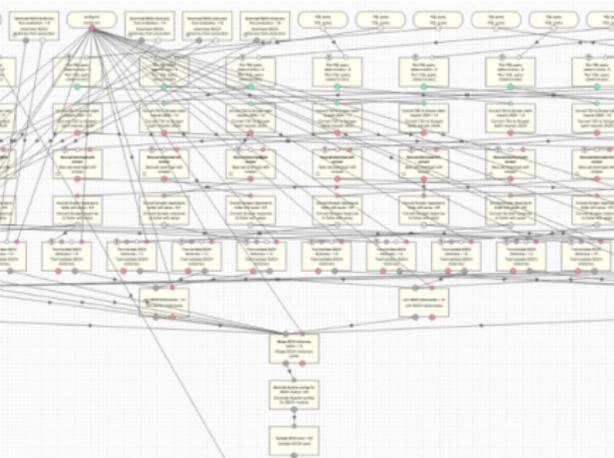


Search this sub-graph
in the complete graph

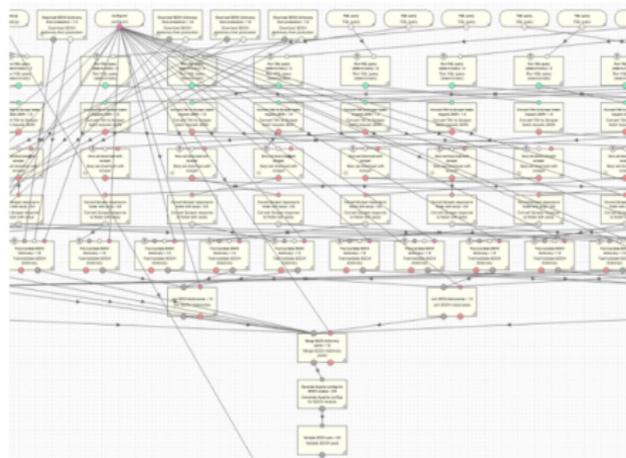
Рис.: Теория распознавания образов

Применение (3)

• •



До изменений



После изменений

Рис.: Нирвана

Цели работы

- ▶ Изучение алгоритмов отличия графов
- ▶ Реализация и оптимизация данных алгоритмов
- ▶ Применение алгоритмов для "Нирваны"
- ▶ Визуализация

Технологии

- ▶ Python3
- ▶ Graphvis (pydot)
- ▶ C++
- ▶ Pybind11 (cppimport)
- ▶ OMP

Рассматриваемые алгоритмы

- ▶ **Полный перебор**
- ▶ Перебор с отсечениями
- ▶ Генетические алгоритмы
- ▶ Муравьиный алгоритм
- ▶ Алгоритм имитации отжига

Рассматриваемые алгоритмы

- ▶ Полный перебор
- ▶ Перебор с отсечениями
- ▶ Генетические алгоритмы
- ▶ Муравьиный алгоритм
- ▶ Алгоритм имитации отжига

Рассматриваемые алгоритмы

- ▶ Полный перебор
- ▶ Перебор с отсечениями
- ▶ Генетические алгоритмы
- ▶ Муравьиный алгоритм
- ▶ Алгоритм имитации отжига

Рассматриваемые алгоритмы

- ▶ Полный перебор
- ▶ Перебор с отсечениями
- ▶ Генетические алгоритмы
- ▶ Муравьиный алгоритм
- ▶ Алгоритм имитации отжига

Рассматриваемые алгоритмы

- ▶ Полный перебор
- ▶ Перебор с отсечениями
- ▶ Генетические алгоритмы
- ▶ Муравьиный алгоритм
- ▶ Алгоритм имитации отжига

Муравьиный алгоритм

Краткие сведения

- ▶ Таблица феромонов на основе прошлых выборов
- ▶ Опираемся на текущую привлекательность выбора
- ▶ Вероятностно выбираем отображение графов
- ▶ Обновляем таблицу феромонов

Муравьиный алгоритм

By Olfa Sammoud, Christine Solnon, and Khaled Ghedira

- ▶ Неупорядоченный выбор вершин для отображения
- ▶ Оптимизируемо до кубической сложности
- ▶ Предлагается 1000 итераций

$$P(u, u') = \text{ScoreFactor}(u, u') \sum_{(v, v') \in CH} \text{PherTable}(u, u', v, v')$$

Муравьиный алгоритм

Модификация

- ▶ Упорядоченный выбор вершин для отображения
- ▶ Квадратическая сложность
- ▶ Завершаем работу после 300 одинаковых итераций
- ▶ (Опционально) добавляем вероятность матча "похожим" вершинам

$$P(u, u') = \text{ScoreFactor}(u, u') \text{PherTable}(u, u') \text{Stat}(u, u')$$

Алгоритм имитации отжига

Краткие сведения

- ▶ Имеем начальное решение
- ▶ Незначительно изменяем решение
- ▶ Вероятностно принимаем новое решение
- ▶ Со временем реже ухудшаем решение

Алгоритм имитации отжига

- ▶ Жадно выбираем начальное решение
- ▶ Для случайной вершины выбираем случайную пару
- ▶ Используется "быстрый" отжиг: $T(k) = \frac{T_0}{k}$
- ▶ Линейная сложность
- ▶ Требуется порядка 10000 итераций
- ▶ Возможность использовать в связанных вычислениях

Графы в "Нирване"

- ▶ Блоки: название операции и входные/выходные данные
- ▶ Ребра: связи по данным или по исполнению
- ▶ Иногда неважны детали, а только факт изменений

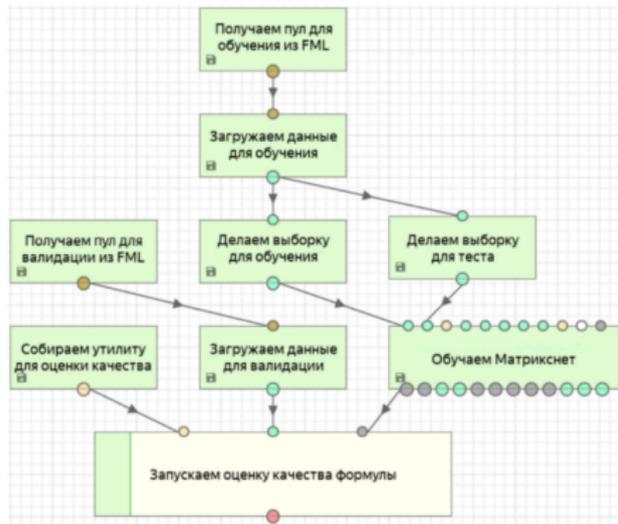


Рис.: Граф "Нирваны"

Результаты в "Нирване"

- ▶ Полное/простейшее преобразование блоков
- ▶ Шаблонный Pipeline: алгоритм, способ преобразования
- ▶ Визуализация

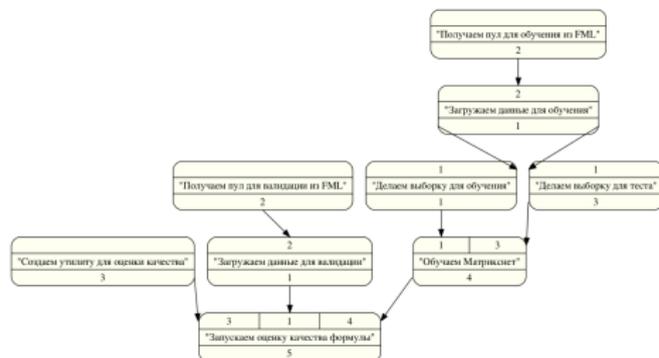


Рис.: Граф "Нирваны"

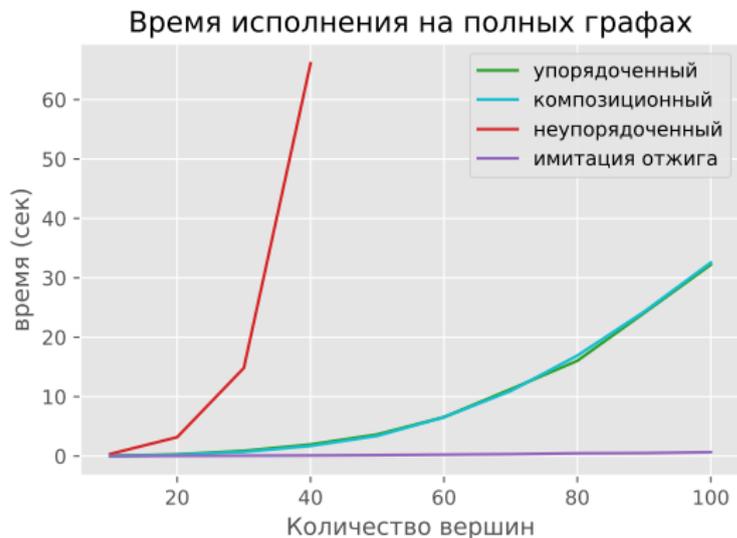
Результаты

- ▶ Изучены алгоритмы отличия графов
- ▶ Алгоритмы реализованы (5 и вариации)
- ▶ Алгоритмы применены для графов "Нирваны"
- ▶ Сделана визуализация
- ▶ Проект выпущен на PyPI

Полезные ссылки

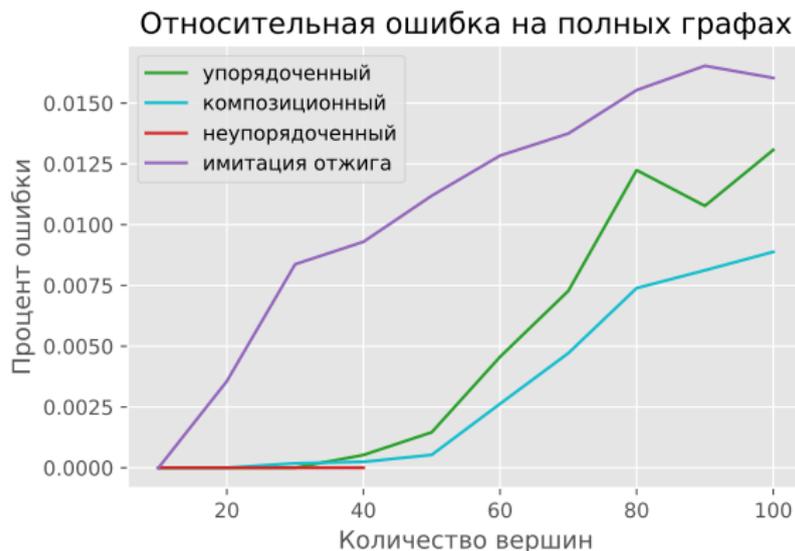
- ▶ alexander.bzikadze@gmail.com
- ▶ yurii.litvinov@gmail.com
- ▶ amosov-f@yandex-team.ru
- ▶ Репозиторий
- ▶ PyPI

Эксперименты (1)



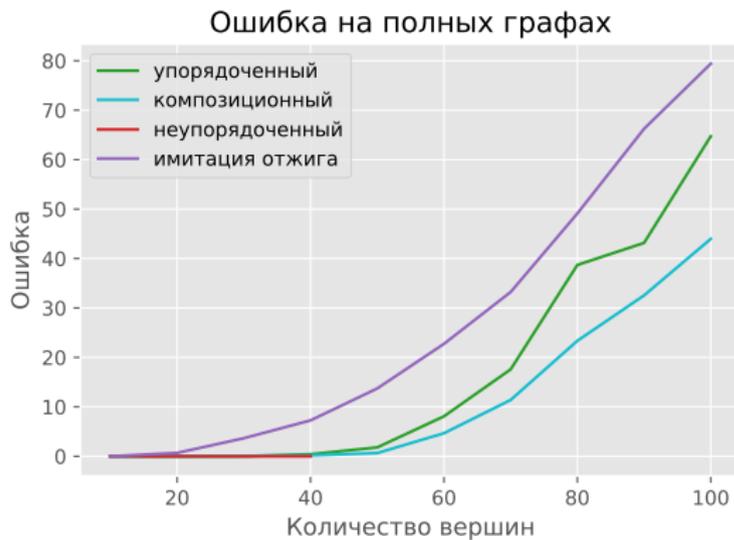
Алгоритм	S^2	$S_{0.1}^2$
(1) Unordered	56411.105570062	10.080270244697
(2) Ordered	12.223421995	6.958452753547
(3) Composed	11.757216180	4.896001973837
(4) Annealing	0.034229817	0.00621532293

Эксперименты (2)



Алгоритм	S^2	$S_{0.1}^2$
(1) Unordered	0.0	0.0
(2) Ordered	0.001325	0.000057
(3) Composed	0.000045	0.000025
(4) Annealing	0.000035	0.000016

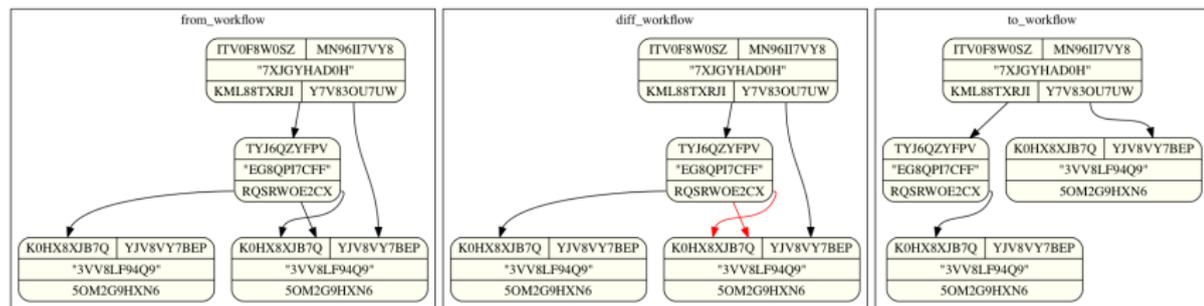
Эксперименты (3)



Галерея (1)



Галерея (2)



Галерея (3)

