Автоматическая кластеризация изменений кода, основанная на сценариях редактирования

Ерохина Алина Сергеевна

Научный руководитель: к.т.н., Брыксин Тимофей Александрович

Национальный исследовательский университет
«Высшая школа экономики»

Санкт-Петербург, 2020 год

Контекст

Задачи, решаемые анализом типовых изменений кода

- Улучшение качества программного обеспечения
  - Изучение процесса эволюции кода
  - Автоматизация частых изменений
  - Автодополнение кода
  - Рекомендации исправлений ошибок
  - Адаптация клиентского кода к изменениям в библиотеке
- Анализ учебного кода
  - Внедрение обратной связи на учебных онлайн-платформах путём анализа частых ошибок

Существующие подходы к поиску схожих изменений

- Подход, основанный на построчной разнице (diff)[1]
  - чувствительность к форматированию
  - не учитывает структуру кода
- Подход, основанный на длине НОП сценариев редактирования AST[2]
  - время работы
- Подход, основанный на графах программных зависимостей (CPatMiner)[3]
  - требует индивидуальных правил построения графов для нового языка

[1] S. Wang et al., Understanding Widespread Changes: A Taxonomic Study. CSMR'13.

[2] P. Kreutzer et al., Automatic clustering of code changes. MSR'16.

[3] H.A. Nguyen et al., Graph-based mining of in-the-wild, fine-grained, semantic code change patterns. ICSE'19.

Цель и задачи

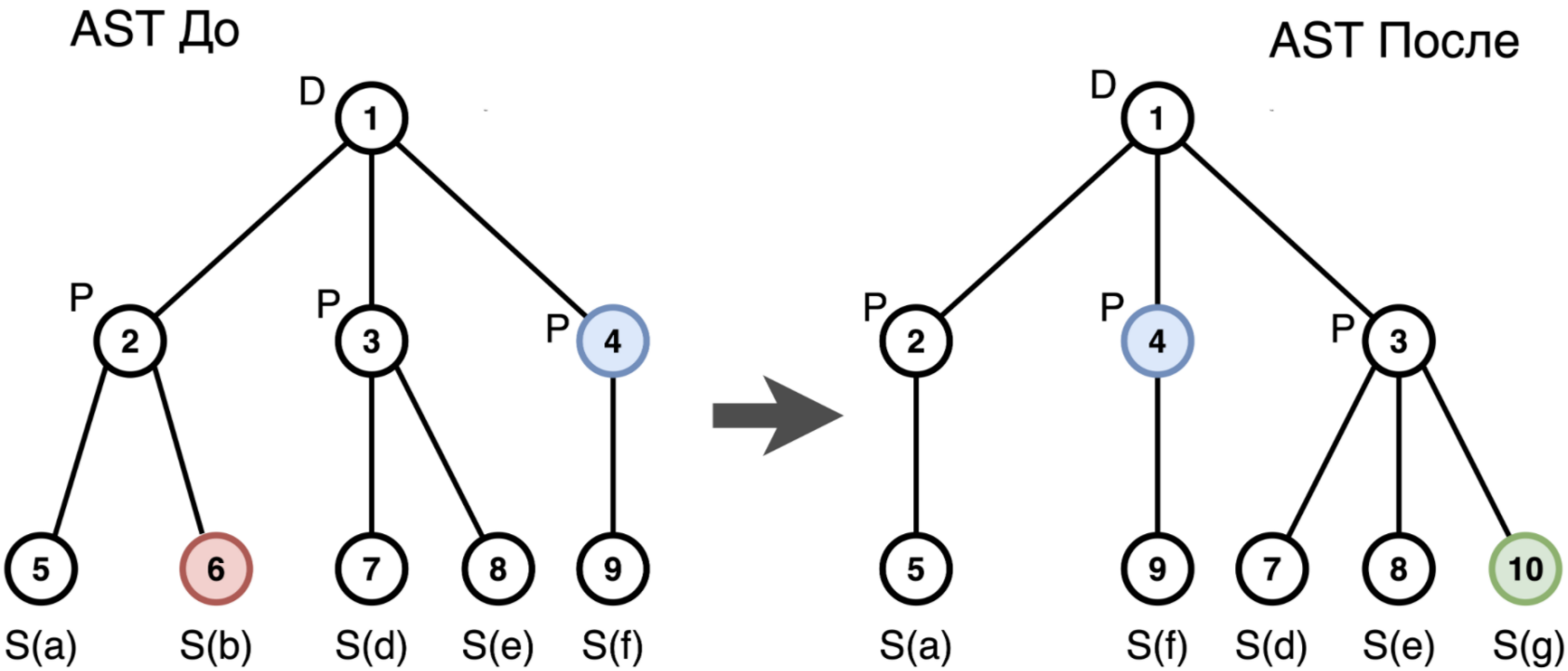
Цель:

- Разработка легковесного и эффективного подхода к представлению и кластеризации изменений кода

Задачи:

- Разработка нового подхода к представлению изменений
- Выбор алгоритма кластеризации и критерия схожести
- Реализация предложенного подхода
- Апробация на существующих датасетах и реальных задачах

Сценарий редактирования



Сценарий редактирования:

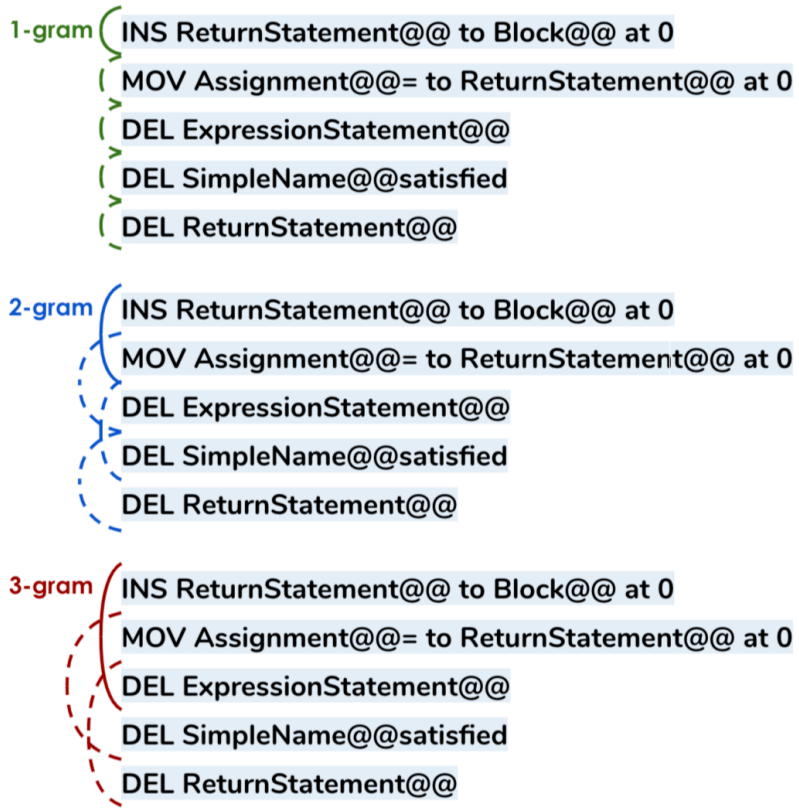
MOV(4, 1, 2)

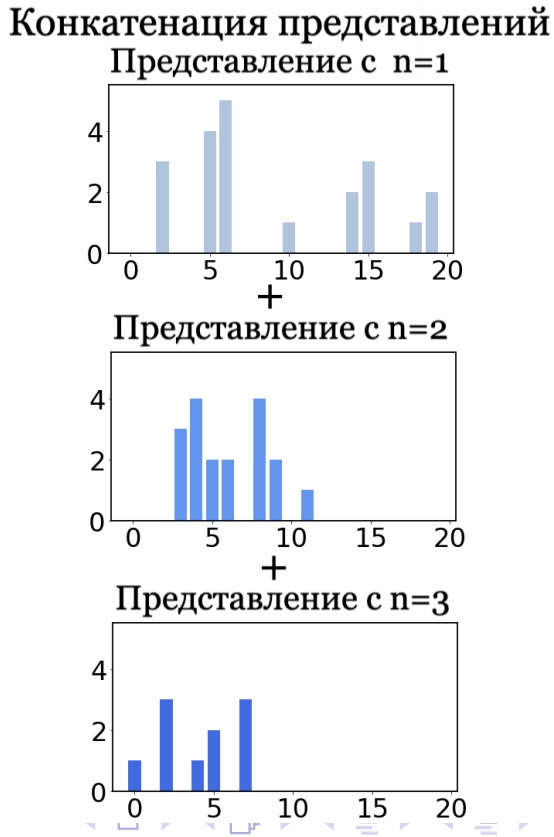
INS((10, S, g), 3, 3)

DEL(6)

Выбор способа представления

- Гипотеза: схожие изменения имеют общие непрерывные подпоследовательности модификаций (n-граммы)
- Идея: строить вектора частот встречаемости n-грамм сценариев редактирования





Выбор способа кластеризации

- Алгоритм кластеризации
  - Критерии: учёт выбросов, достаточно матрицы расстояний, оптимальное время работы
    - DBSCAN и Agglomerative Hierarchical Clustering (HAC)
- Функция схожести
  - Критерии: нормируемость, независимость от длины представления
  - Рассмотрение различных вариантов схожести
    - Расстояние Жаккара
    - Канберрское расстояние
    - Косинусное расстояние
    - Расстояние Пирсона

Особенности реализации

- Использование библиотеки GumTree для построения AST и сценариев редактирования
- Метки вершин игнорируются
- Использование разреженных списков
- Код: github.com/JetBrains-Research/code-changes-clustering

Апробация

Сравнение функций расстояния

- Датасет из работы Tufano et al.[4]
- 627 изменения, 58 кластеров, средний размер класса: 10.8

Алгоритм	N	Функция расстояния	Число кластеров	Число выбросов	Энтропия	Чистота	F-мера
DBSCAN	1	Jaccard	10	23 (3.7%)	4.503	0.200	0.133
		Canberra	80	129 (20.6%)	0.674	0.815	**0.542**
		Cosine	77	126 (20.1%)	0.774	0.806	0.534
		Pearson	77	126 (20.1%)	0.774	0.806	0.534
DBSCAN	1-5	Jaccard	10	22 (3.5%)	4.502	0.200	0.134
		Canberra	71	103 (16.4%)	0.979	0.744	**0.511**
		Cosine	85	147 (23.5%)	0.622	0.831	0.510
		Pearson	85	148 (23.6%)	0.623	0.831	0.510
DBSCAN	5	Jaccard	91	155 (24.7%)	0.575	0.816	0.464
		Canberra	91	155 (24.7%)	0.575	0.816	0.464
		Cosine	91	155 (24.7%)	0.575	0.816	0.464
		Pearson	91	155 (24.7%)	0.575	0.816	0.464

[4] M. Tufano et al. On learning meaningful code changes via neural machine translation. ICSE '19.

Тестирование

Сравнение различных представлений и значений N

- Датасет из работы Tufano et al.
- 627 изменения, 58 кластеров, средний размер класса: 10.8

Алгоритм	N	Функция расстояния	Число кластеров	Число выбросов	Энтропия	Чистота	F-мера
DBSCAN	1	Canberra	80	129 (20.6%)	0.674	0.815	**0.542**
	3	Canberra	86	166 (26.5%)	0.562	0.835	0.494
	1-3	Canberra	73	94 (15.0%)	1.101	0.702	0.473
	5	Canberra	91	155 (24.7%)	0.575	0.816	0.464
	1-5	Canberra	71	103 (16.4%)	0.979	0.744	0.511
		LCS	76	107 (17.1%)	0.807	0.773	0.527
HAC complete	1	Canberra	54	29 (4.6%)	1.323	0.669	0.562
	3	Canberra	98	113 (18.0%)	0.651	0.794	0.502
	1-3	Canberra	60	30 (4.8%)	1.284	0.673	0.555
	5	Canberra	92	158 (25.2%)	0.555	0.823	0.464
	1-5	Canberra	60	30 (4.8%)	1.263	0.673	0.551
		LCS	63	45 (7.2%)	1.097	0.713	**0.570**

Тестирование

Запуск на размеченных датасетах

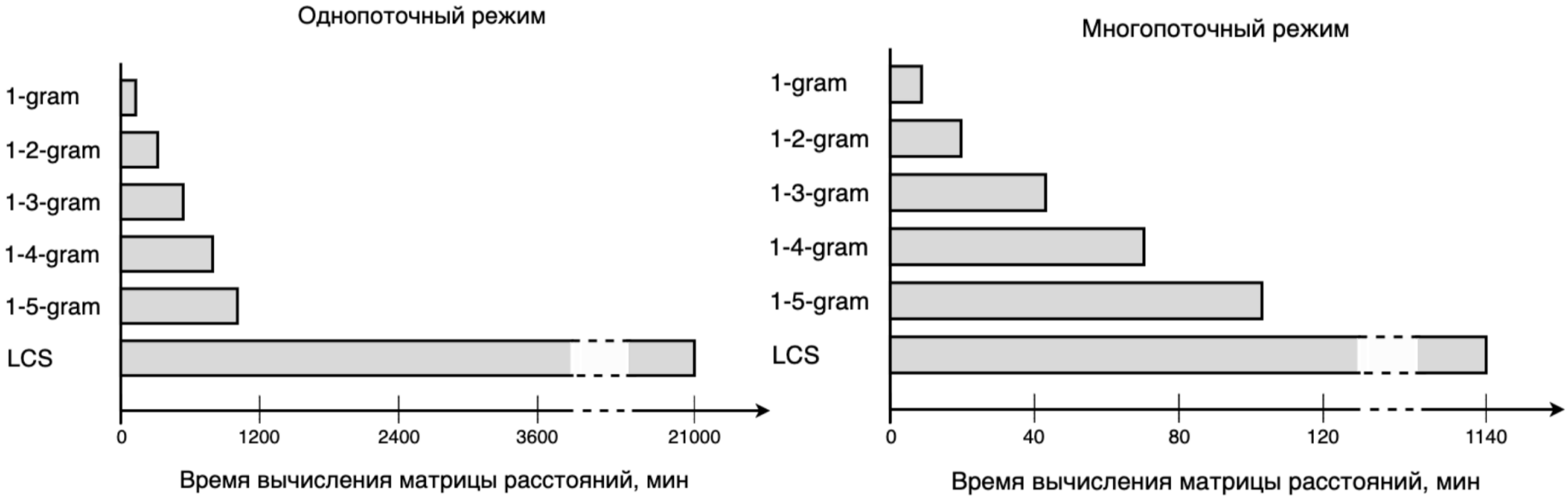
- Датасет для тестирования LASE[5]: 149 элементов, 24 класса, средний размер класса: 6.2
- Более крупные изменения, более схожие изменения в группе

Алгоритм	N	Функция расстояния	Число кластеров	Число выбросов	Энтропия	Чистота	F-мера
DBSCAN	1	Canberra	29	4 (2.7%)	0.000	1.000	0.933
	1-2	Canberra	26	3 (2.0%)	0.037	0.993	0.955
	1-3	Canberra	27	3 (2.0%)	0.037	0.993	0.955
	1-4	Canberra	27	3 (2.0%)	0.037	0.993	0.955
	1-5	Canberra	25	3 (2.0%)	0.037	0.993	0.975
	1-6	Canberra	25	2 (1.3%)	0.037	0.993	**0.978**
	1-7	Canberra	25	2 (1.3%)	0.037	0.993	**0.978**
	1-8	Canberra	26	2 (1.3%)	0.037	0.993	0.971
	LCS		29	5 (3.4%)	0.000	1.000	0.926

[5] N. Meng et al., LASE: Locating and Applying Systematic Edits by Learning from Examples.

Сравнение времени работы

- Вычисление матрицы расстояний на датасете из $\approx$ 23к элементов из неразмеченного датасета из работы Tufano et al.[6]



- Вычисление элемента матрицы:
  - LCS: $\mathcal{O}(s_1 s_2)$, где s_1 и s_2 — длины сценариев редактирования
  - 1-k-gram: $\mathcal{O}(s_1 + s_2)$, считая, что $\mathcal{O}(k) = \mathcal{O}(1)$

[6] M. Tufano et al. On learning meaningful code changes via neural machine translation. ICSE '19.

Применение в задаче определения типов ошибок в решениях задач по программированию

- В предыдущей работе[7] использовались подходы на основе:
  - коэффициента Жаккара и разных вариантов схожести модификаций (def_jac, ext_jac, ful_jac), нечёткой модификации расстояния Жаккара (fuz_jac), модели bag-of-words (bow)
- Для оценки качества решений использовалась метрика PR-AUC

Подход к кластеризации	Подход к классификации	Классификатор	A	B	C	D	Среднее
BOW20000	fuz_jac	k-nearest-15	**0.851**	0.720	0.796	0.772	**0.785**
BOW20000	fuz_jac	k-nearest-10	0.845	**0.725**	0.788	0.770	0.782
ful_jac	fuz_jac	k-nearest-3	0.803	0.674	0.802	0.752	0.758
BOW20000	def_jac	k-nearest-15	0.817	0.675	0.752	0.779	0.756
1-2-gram	fuz_jac	k-nearest-10	0.803	0.677	**0.803**	0.752	0.759
BOW20000	1-3-gram	k-nearest-15	0.845	0.704	0.764	0.782	0.774
1-gram	1-2-gram	k-nearest-10	0.801	0.679	0.766	**0.783**	0.757

[7] Lobanov A., Bryksin T., Shpilman A., Automatic Classification of Error Types in Solutions to Programming Assignments at Online Learning Platform. AIED'19.

Результаты

- Разработан новый легковесный и эффективный подход к представлению изменений кода, основанный на n-граммах сценариев редактирования
- Проанализированы и выбраны алгоритмы кластеризации и критерии схожести
- Произведена реализация предложенного подхода
- Проведена апробация подхода
  - Проведён анализ качества работы подхода на датасетах с различными характеристиками, выполнено сравнение с аналогичным подходом
  - Подход был применён в практической задаче, и в нескольких случаях наблюдалось улучшение результата
  - Эффективность работы нового подхода, достигнутая не в убыток качеству, позволяет предположить, что он может иметь широкое практическое применение

Выбор способа представления и кластеризации

Функция расстояния

- Расстояние Жаккара

$$JD(v_1, v_2) = 1 - \frac{1}{|intersect(v_1,v_2)|} \sum_{i \in intersect(v_1,v_2)} \frac{min(v_1[i],v_2[i])}{max(v_1[i],v_2[i])}$$

- Канберрское расстояние

$$KD(v_1, v_2) = \frac{1}{|union(v_1,v_2)|} \sum_i \frac{|v_1[i]-v_2[i]|}{|v_1[i]|+|v_2[i]|}$$

- Косинусное расстояние

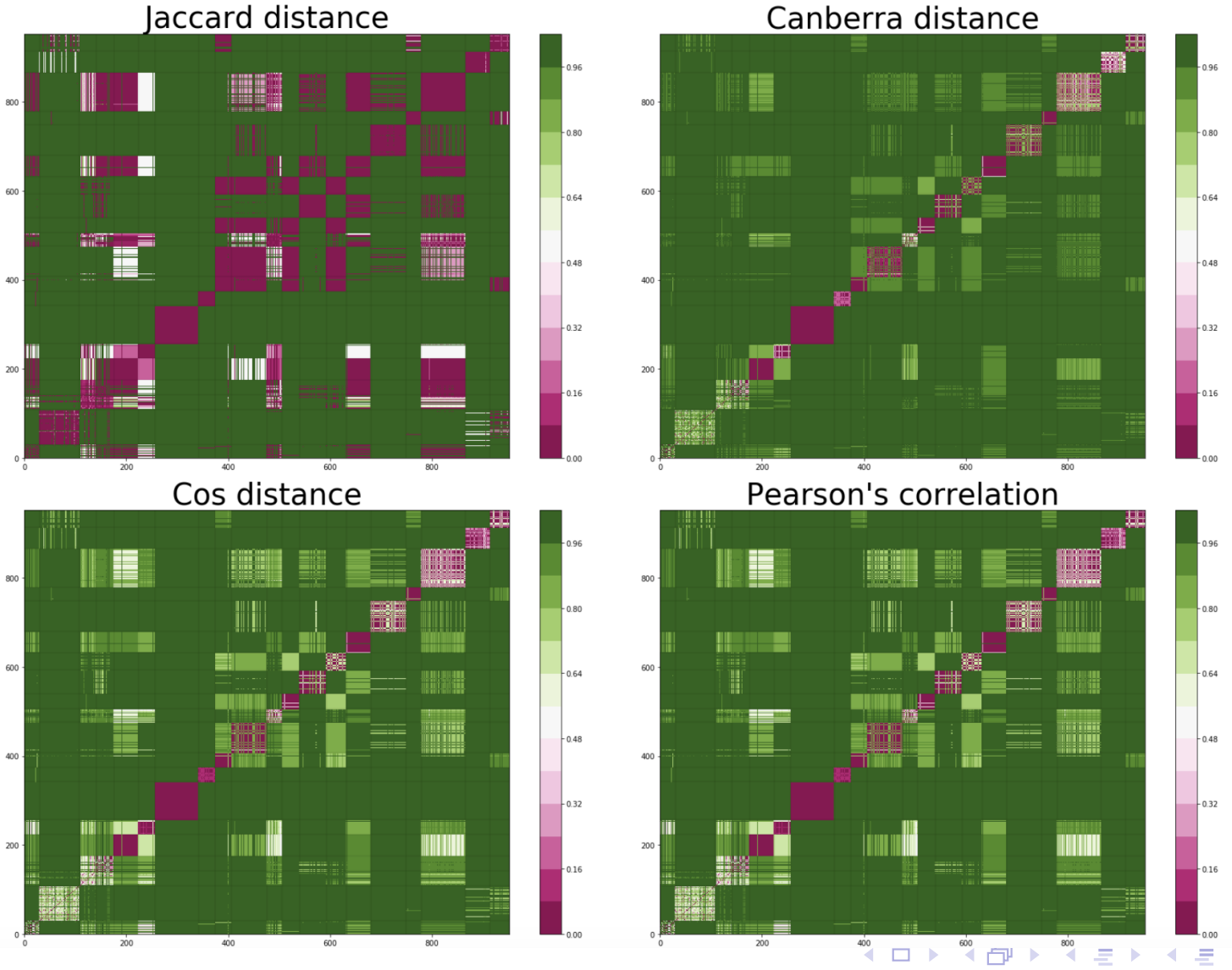
$$CD(v_1, v_2) = 1 - \frac{\sum_i v_1[i]*v_2[i]}{\sqrt{\sum_i v_1[i]^2}\sqrt{\sum_i v_2[i]^2}}$$

- Коэффициент корреляции Пирсона

$$PD(v_1, v_2) = 1 - \frac{\sum_i (v_1(i)-\overline{v_1})(v_2(i)-\overline{v_2})}{\sqrt{\sum_i (v_1(i)-\overline{v_1})^2 \sum_i (v_2(i)-\overline{v_2})^2}}$$

Реализация

Визуализация матриц расстояний



Оценка качества кластеризации

- Внешние
  - есть эталонная кластеризация
  - **Энтропия:** $e = \sum_i^K \frac{m_i}{m} e_i$, где e_i — энтропия для кластера, $0 \leq e \leq \log L$, где L — кол-во классов
  - **Чистота:** $purity = \sum_i^K \frac{m_i}{m} purity(i)$, где $purity(i) = max_j p_{ij}$
  - **F-мера:** $\sum_j \frac{m_j}{m} max_i F(i,j)$, где $F(i,j) = \frac{2precision(i,j)recall(i,j)}{precision(i,j)+recall(i,j)}$
  - **Rand Index:** $\frac{TP+FN}{TP+TN+FP+FN}$, **Jaccard Index:** $\frac{TP}{TP+TN+FP}$
- Внутренние
  - на основе расстояний внутри кластеров и между кластерами