Вариант 3: Кластеризация. Отчёт

Кудисов Артём Аркадьевич, группа 325

November 7, 2021

---

1 Постановка задачи

Необходимо было написать программу кластеризации неразмеченного набора данных и, используя 2 разные метрики схожести и 2 метода кластеризации разных видов, провести сравнение между ними на основании 2-ух внутренних и 2-ух внешних мер качества.

В работе использовались оценки качества в том виде, в котором они описаны на сайтах:

- https://neerc.ifmo.ru/wiki/index.php?title=Оценка_качества_в_задаче_кластеризации
- https://scikit-learn.org/stable/modules/clustering.html
- https://www.mathworks.com/help/stats/clustering.evaluation.calinskiharabaszevaluation-class.html

2 Описание датасета

Данные были взяты с сайта http://cs.joensuu.fi/sipu/datasets/. (раздел S-sets, S2) и представляют из себя 5000 двумерных векторов, искуственно сгенерированных из распределения Гаусса на основании 15 кластеров, немного пересекающихся друг с другом (рис. 1).

Чтобы программа работала быстрее (особенно при использовании агломеративного метода), я проредил количество примеров, удалив кластеры 1, 3, 8, 9, 11, 13, 14 (рис. 2). В итоге осталось 2666 примеров.

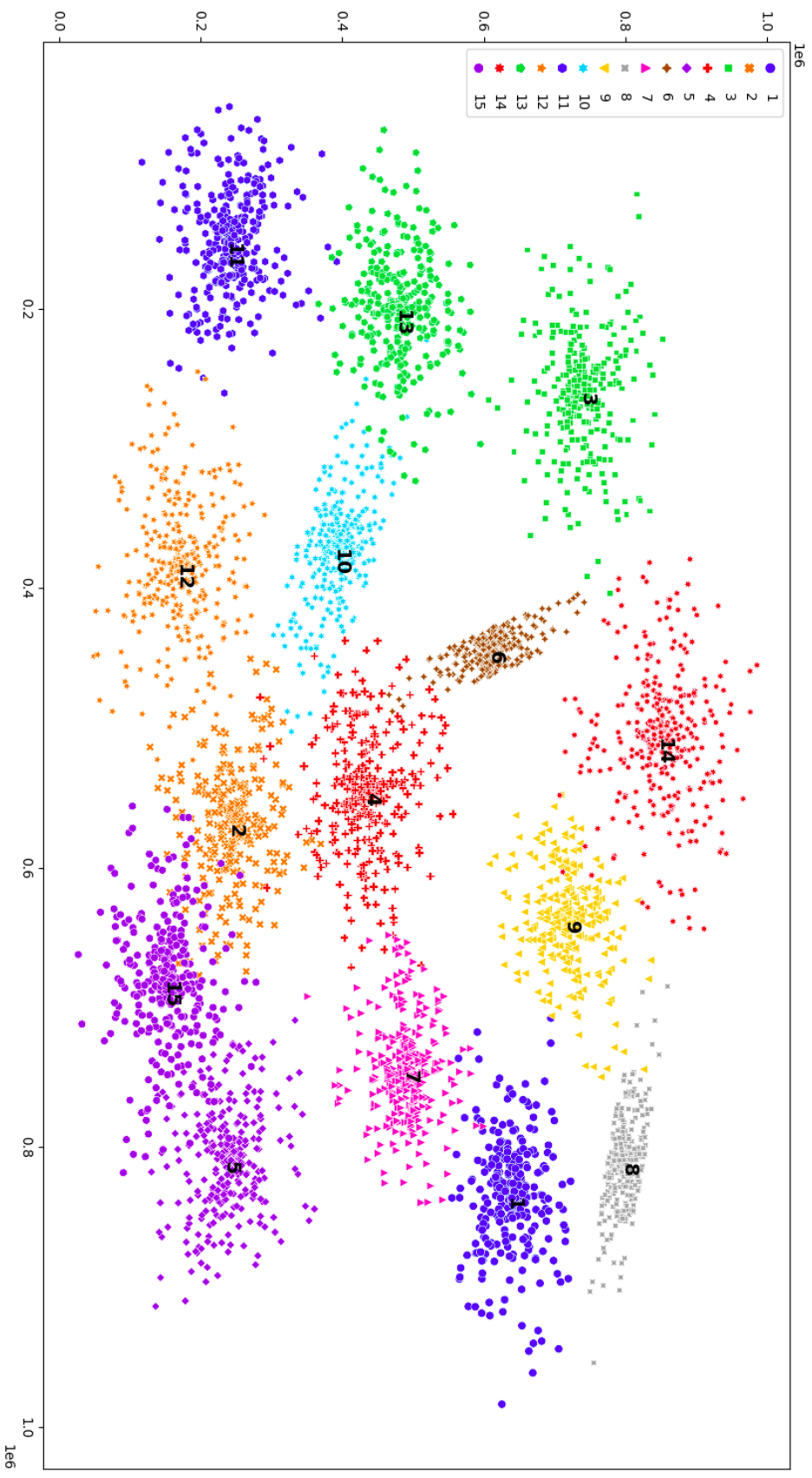

Рис. 1: Оригинальные данные. 5000 примеров. 15 кластеров

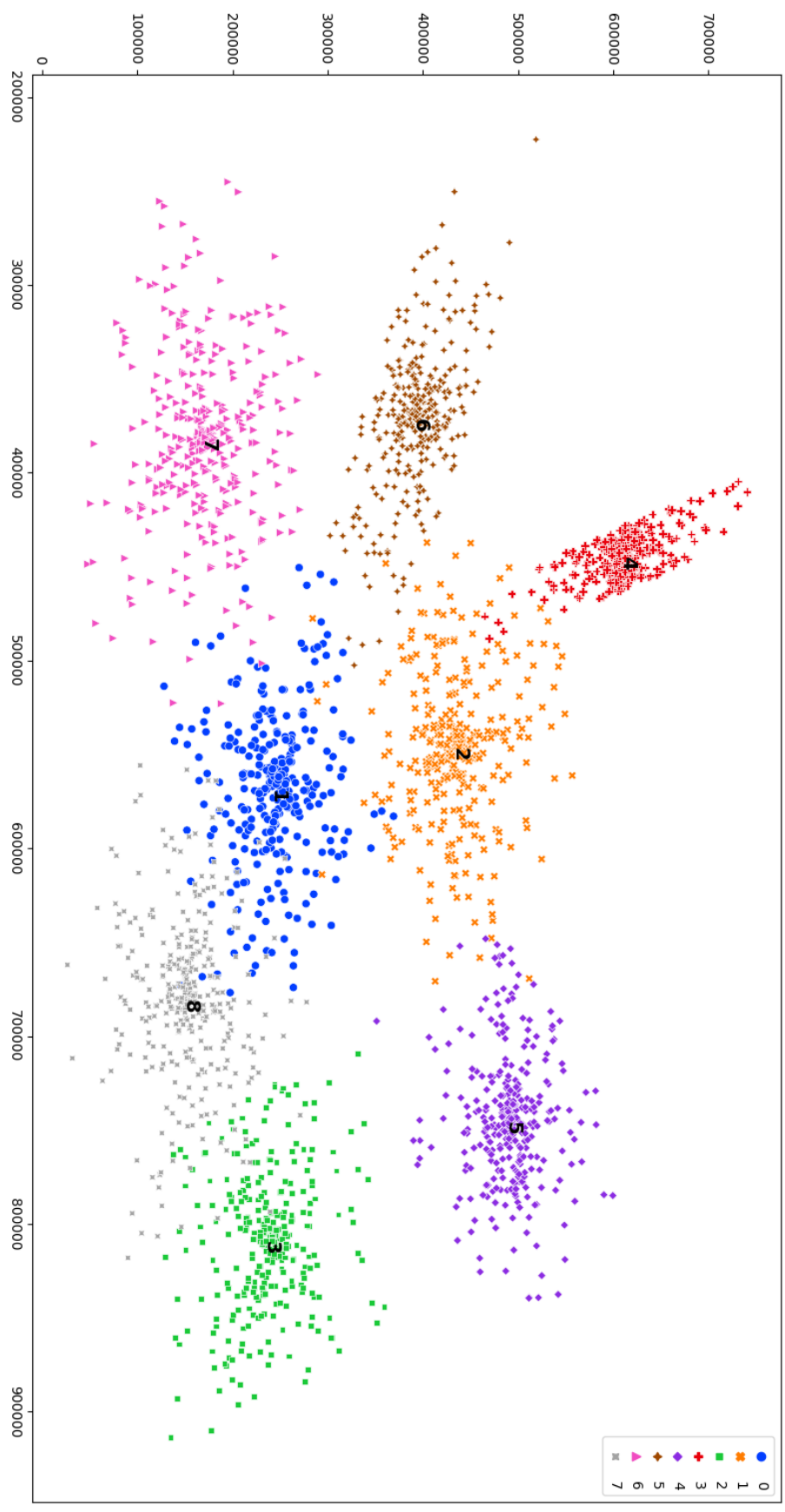

Рис. 2: Оставшиеся данные. 2666 примеров. 8 кластеров

3 Описание программы

- Самостоятельно реализовал все требующиеся элементы:
  - 2 алгоритма кластеризации:
    - ◇ **KMeans** (используется расстояние между центроидами кластеров)
    - ◇ **Agglomerative** (используется расстояние между центроидами кластеров)
  - 3 метрики схожести:
    - ◇ **l1** (Манхэттенское расстояние)
    - ◇ **l2** (Евклидова норма)
  - 2 внутренние метрики качества:
    - ◇ **Silhouette Score**
    - ◇ **Calinski-Harabasz Index**
  - 3 внешние метрики качества:
    - ◇ **Rand Index**
    - ◇ **Adjusted Rand Index**
    - ◇ **Jaccard Index**
- В работе использую следующие сторонние библиотеки:
  - **sklearn**: для того, чтобы убедиться, что реализованные оценки верни (путем сравнения с уже написанными функцями из этой библиотеки)
  - **matplotlib.pyplot**: для вывода графиков
  - **numpy**: для работы с матрицами
  - **seaborn**: для вывода графиков
  - **tqdm**: для отслеживания выполнения работы
- Сам же проект состоит из следующих файлов и директорий:
  - **data**
    - ◇ **s2.txt**
    - ◇ **s2-label.txt**
  - **utils**:
    - ◇ **clustering.py** (Методы кластеризации)

◇ **general_utils.py** (Вспомогательные функции)

◇ **metrics.py** (Меры близости)

◇ **scores.py** (Внешние и внутренние оценки)

– **run.py** (Основной скрипт)

Важные замечания:

1. При вызове KMeans этот метод запускается несколько раз (10). Выбирается то разбиение, при котором получается наименьшая сумма квадратов расстояния от точек до соответсвующих ближайших соседних класетеров.
2. Внутренняя оценка Silhouette использует в своей работе указанную меру близости, в то время как Calinski-Harabasz всегда считает $||l2||_2^2$ (ровно как в библиотеке sklearn)
3. При работе алгоритмы кластеризации перебирают число кластеров от 4 до 9 включительно - выбирают то из них, при котором мы получили наилушую внутреннюю оценку

4 Сравнение

		KMeans		Agglomerative	
		silhouette	calinski	silhouette	calinski
l1	RI	0.980	0.980	0.964	0.964
	ARI	0.912	0.912	0.840	0.840
	JI	0.857	0.857	0.755	0.755
l2	RI	0.980	0.980	0.949	0.964
	ARI	0.912	0.912	0.785	0.840
	JI	0.857	0.857	0.687	0.755

5 Выводы

Как хорошо видно, на данном датасете лучшего всего себя показывает алгоритм Kmeans, причем независимо от используемой внутренней оценки и меры близости. Уверен, что это всё благодаря хорошей стуктуре данных в датасете - довольно легко выделять кластеры.

Но и агломеративная кластеризация показывает себя очень хорошо. Однако она в данном случае всё-таки немного проигрывает KMeans и по качеству и по скорости работы.

Делать какие-то глобальные выводы о том, какой из методов лучше не стану. Потому что для каждой задачи подходит свой собственный алгоритм.

Но по крайней мере могу сказать, что мне удалось выполнить это задание и реализовать с нуля весь основной функционал программы, будь то меры близости, оценки, алгоритмы кластеризации (и надеюсь, что даже без ошибок).