2025-02-19

Задачи классификации и кластеризации

Обзор класса задач

Оба класса задач подразумевают, что есть набор некоторых объектов (в широком смысле), которые обладают некоторым набором характеристик (общим для всех объектов). На основании того, какие характеристики у этих объектов есть и в какой мере они выражены, мы эти объекты хотим сгруппировать.

Классификация предполагает, что группы уже выделены заранее: для каждого объекта в обучающей выборке уже есть свой класс (т.е. для каждого класса всегда есть заранее подготовленные примеры).

Классификация относится к задачам на обучение с учителем (supervised learning) и с точки зрения машинного обучения сводится к обучению модели признакам, характеризующим некоторый класс, и обобщению наборов признаков отдельных объектов класса для характеристики класса в целом.

Далее обученная модель относит любые новые данные и встреченные в них новые объекты (не входившие в обучающую выборку) к тому или иному классу, сличая их свойства с обобщенным внутренним представлением свойств (признаков) каждого класса.

Кластеризация относится к методам обучения без учителя (unsupervised learning) и ставит своей целью выяснение того, какие группы можно выделить в изначально несгруппированном наборе данных. Признаки, по которым производится разделение на группы, также выявляются в ходе обучения модели.

Модель должна разбить входные данные на N групп, при этом само число групп чаще всего задается заранее (хотя и не всегда — например, алгоритм affinity propagation сам итеративно подбирает число кластеров, пока не стабилизируется). Необходимое число групп можно приблизительно определить в ходе предварительного анализа данных, используя, например, ранее рассмотренный метод локтя.

Применительно к кластерному анализу это часто также подразумевает итеративное выполнение алгоритма с разным целевым числом кластеров и сравнением метрик качества кластеризации для разного числа кластеров.

Одной из наиболее популярных метрик является коэффициент силуэта — мера пересечения кластеров друг с другом.

Info

The Silhouette Coefficient is calculated using the mean intra-cluster distance `(a)` and the mean nearest-cluster distance `(b)` for each sample. The Silhouette Coefficient for a sample is `(b - a) / max(a, b)`. To clarify, b is the distance between a sample and the nearest cluster that the sample is not a part of. We can compute the mean Silhouette Coefficient over all samples and use this as a metric to judge the number of clusters.

Коэффициент силуэт — рассчитывается с использованием среднего расстояния внутри кластера (`a`) и среднего расстояния до ближайшего кластера (`b`) для каждого образца, т.е. насколько каждый образец похож на другие в своем кластере и непохож на образцы в другом ближайшем кластере

Идеальное значение — 1 (внутри группы все похожи, на другую группу никто не похож); худшее — -1 (кластеры подобраны неверно); 0 — кластеры перекрывают друг друга (описанная выше ситуация с объектами на границах и компромиссных решениях)

scikit-learn

Документация: [User Guide — scikit-learn 1.6.1 documentation]

Официальный сайт: https://scikit-learn.org/

Страница библиотеки в PyPI: scikit-learn · PyPI

```
pip install scikit-learn
```

Исходный код: GitHub - scikit-learn/scikit-learn: scikit-learn: machine learning in Python

Изначально появилась в 2007-м году как библиотека предиктивной аналитики. Само название «Scikit» появилось от сокращения «SciPy Toolkit» — т.е. задумана она была как расширения инструментов SciPy для предиктивной аналитики.

Соответственно, построена на NumPy, SciPy, matplotlib и легко интегрируется с ними.

Автор — французский исследователь, аналитик данных и разработик **David Cournapeau** (Давид Курнапо). В дальнейшем разработку продолжили другие исследователи и программисты, например, Александр Грамфор (agramfort (Alexandre Gramfort) · GitHub)

Библиотека написана на Python, Cython, C++ и чистом C.

- включает в себя множество классических датасетов для демонстрации и обучения (например, Ирисы Фишера и т.д.)
- пермиссивная (разрешающая) лицензия BSD и открытый исходный код — можно свободно использовать все модели и алгоритмы в любых (в том числе коммерческих) проектах

Из минусов:

- не подходит для deep learning (глубокого обучения), нужно использовать более специализированные инструменты
- в последнее время выявились проблемы масштабирования (по-настоящему большие датасеты сложно обрабатывать, не хватает инструментария для эффективного скейлинга)
- нет возможности легко перевести данные из датафрейма pandas в вид, пригодный для модели из scikit-learn — придется использовать в качестве

промежуточных структур ndarray из NumPy

Структура и основные модели и алгоритмы библиотеки

1. Обучение с учителем (Supervised learning)
2. Обучение без учителя (Unsupervised learning)
3. Выбор моделей и их оценка
4. Инспекция данных
5. Визуализация данных
6. Преобразования данных
7. Инструменты загрузки готовых датасетов

Модели и алгоритмы обучения с учителем

[1. Supervised learning — scikit-learn 1.6.1 documentation]

- Линейные модели, включая ElasticNet, логистическую регрессию (прогнозирует обычно вероятности), перцептроны, стохастический градиентный спуск (Stochastic Gradient Descent, SGD), гребневая регрессия (ридж-регрессия, ridge regression, регрессия Тихонова) — относится к методам снижения размерности
- Машины опорных векторов (Support Vector Machines, SVM) — модель для классификации
- Метод k-ближайших соседей — тоже классификация
- Наивный Байес (наивный байесовский классификатор) — классификация
- Деревья решений (decision tree) — регрессия или классификация
- Ансамблевые методы — случайный лес (random forest) — ансамбль на базе дерева решений; деревья с градиентным бустингом (дерево решений + градиентный бустинг), голосующий классификатор (Voting Classifier) — объединение нескольких моделей машинного обучения (можно разных) в один ансамбль, каждая модель производит свою классификацию, после чего производится голосование

Больший плюс документации — есть возможность скачать примеры как в виде .py-файла (plot_adaboost_regression.py), так и в виде Jupyter-ноутбука (plot_adaboost_regression.ipynb). Весь пример реализации такого алгоритма, включая исходные данные, доступен для изучения и экспериментов.

Модели и алгоритмы обучения без учителя

2. Unsupervised learning — scikit-learn 1.6.1 documentation

Обучение без учителя не предполагает какой-либо разметки данных до начала обучения модели.

Некоторые алгоритмы кластерного анализа

Алгоритм k-средних (иногда — алгоритм Ллойда)

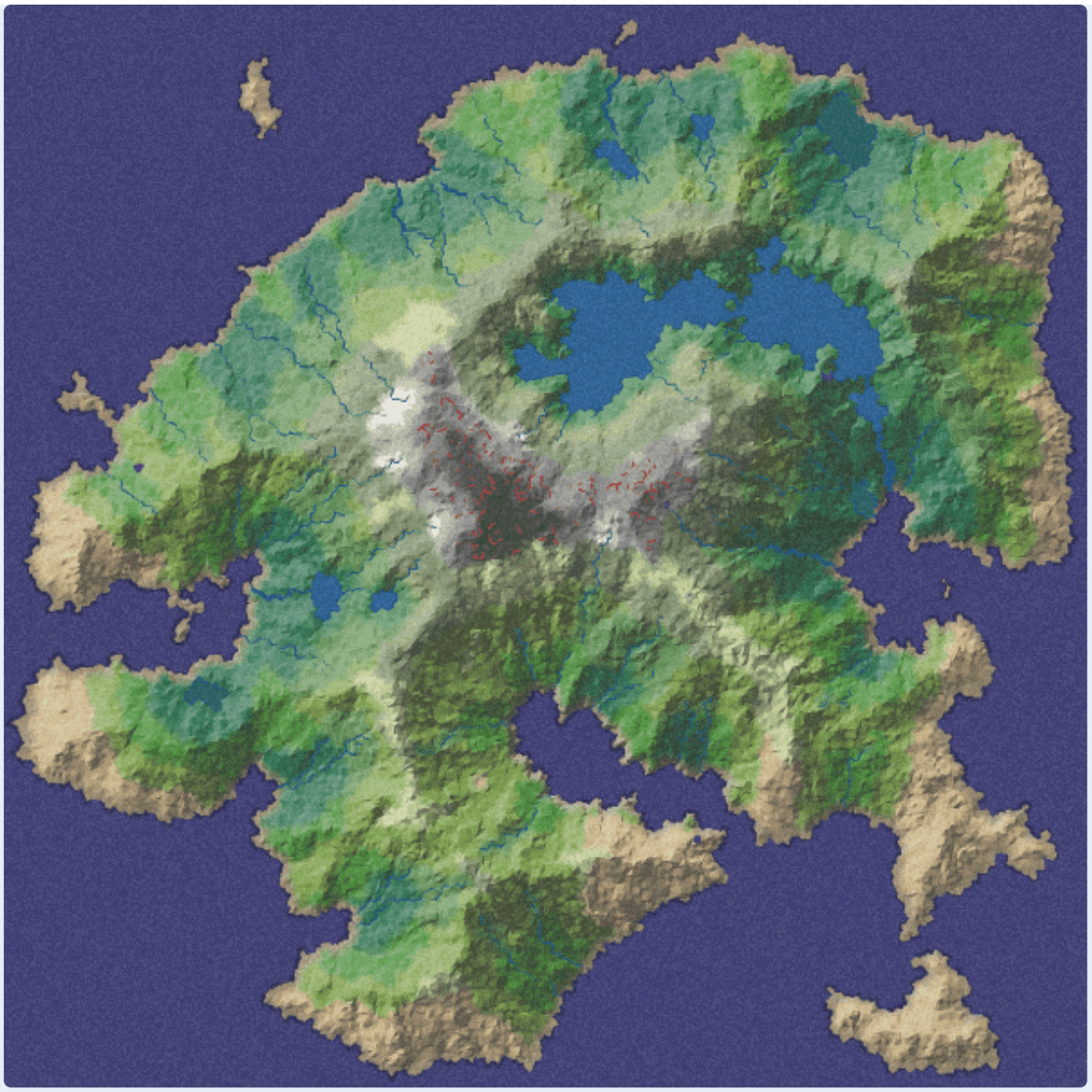
стремится минимизировать инерцию (intertia), выбирая такие центры кластеров, для которых сумма квадратов расстояний от центра до точек минимальна.

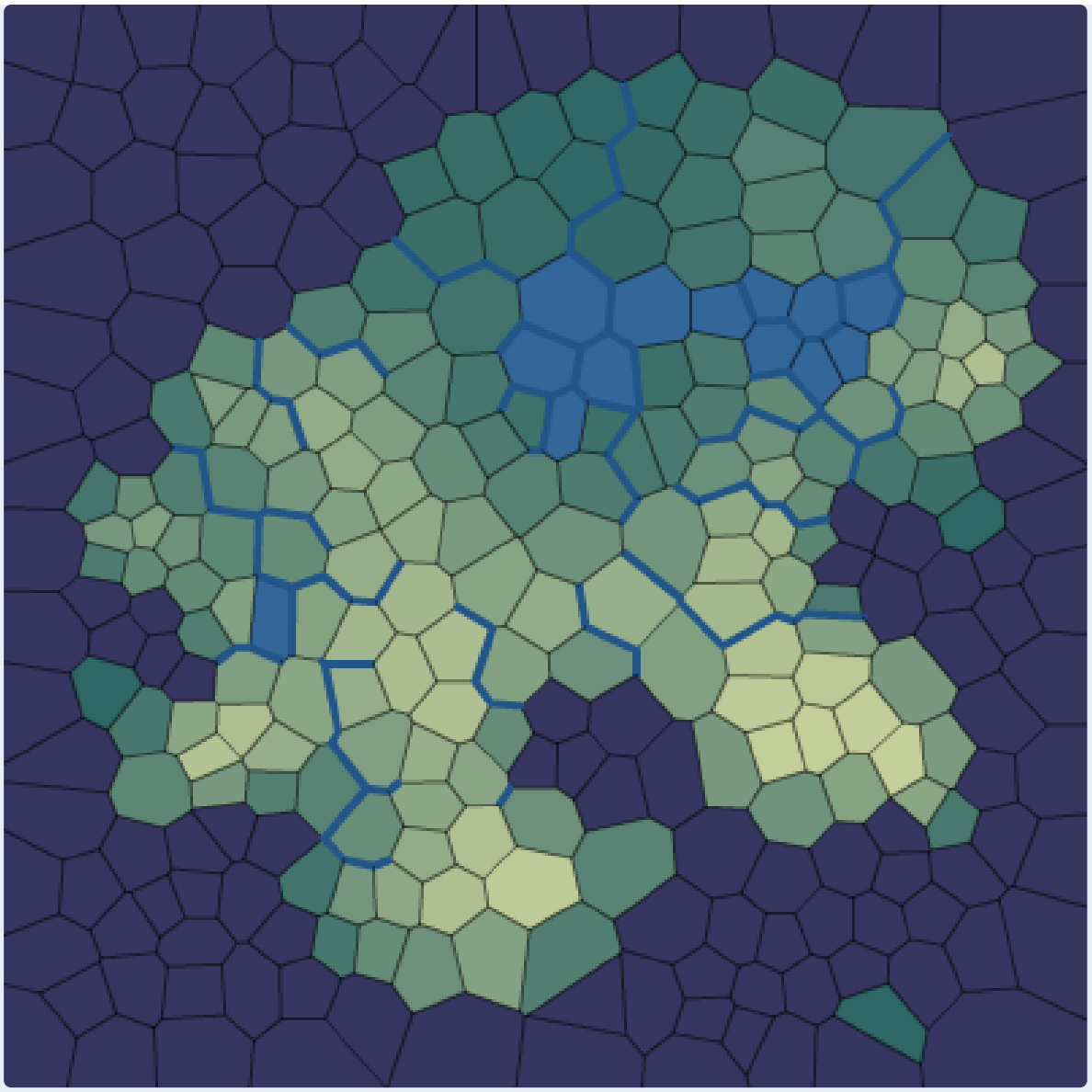
Страдает от т.н. «проклятия размерности» — т.к. инерция не является нормализованной метрикой, то Евклидовы расстояния для пространств высоких размерностей быстро возрастают. Поэтому часто метод k-средних используют в связке с методами снижения размерности (например, методом главных компонент).

Алгоритм требует задания числа кластеров, поэтому также нужно применять метод локтя или аналогичные способы поиска оптимального числа кластеров.

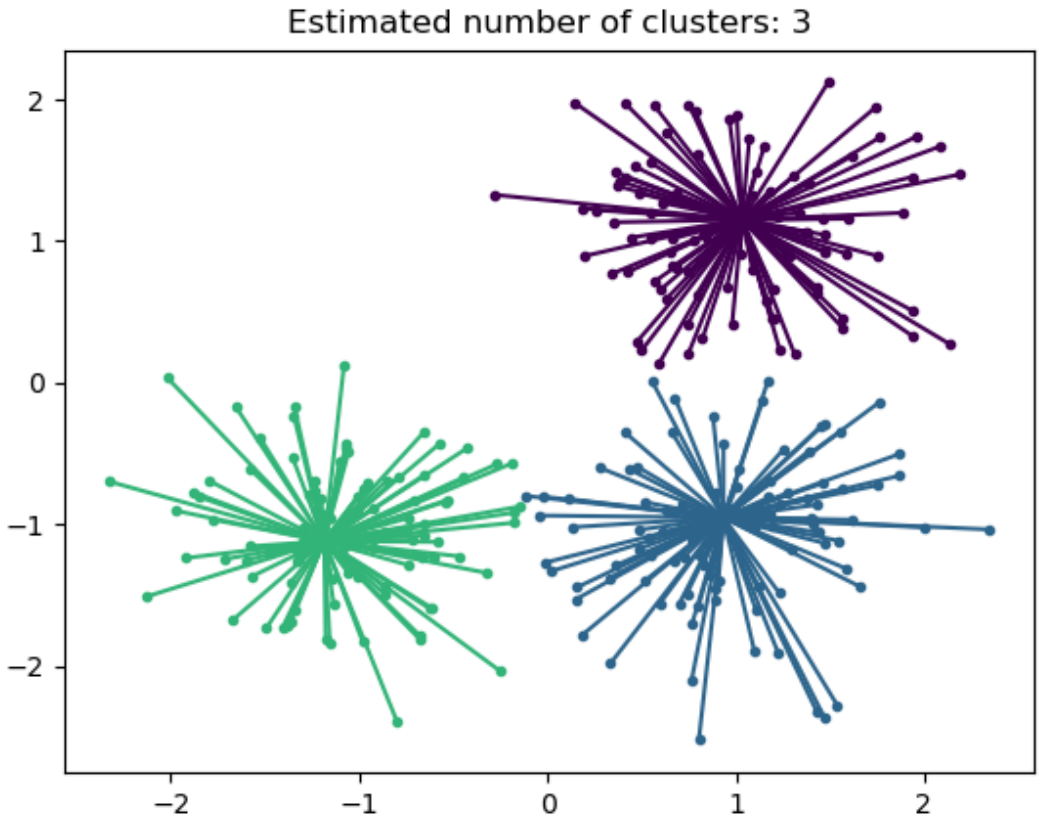
Info

Похожим на k-средние образом считаются диаграммы Вороного http://www-cs-students.stanford.edu/~amitp/game-programming/polygon-map-generation/





Алгоритм распространения близости (Affinity Propagation)



В отличие от многих других кластерных алгоритмов не требует задания числа кластеров.

Алгоритм итеративно посылает «сообщения» между парами образцов (объектов) пока не наступит эквилибриум — нельзя будет переместить образец из одной группы в другую. Сообщения несут смысл того, насколько один элемент в паре может служить образцом для другого (т.е. насколько они похожи между собой)

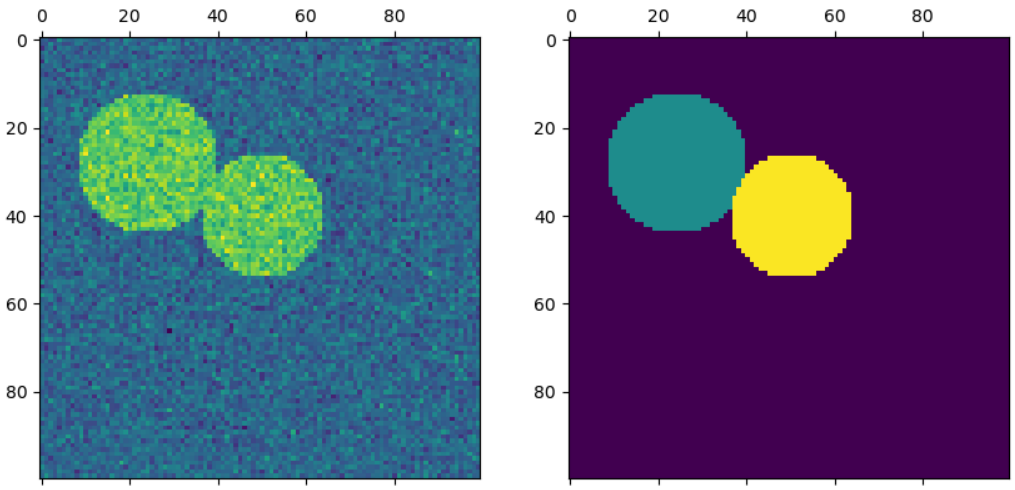
Иногда выходят компромиссные варианты (т.е. объекты на границах кластеров не имеют четко выраженной принадлежности ни к одной из двух групп и оказались в какой-либо из них просто по достижению предела и общего эквилибриума)

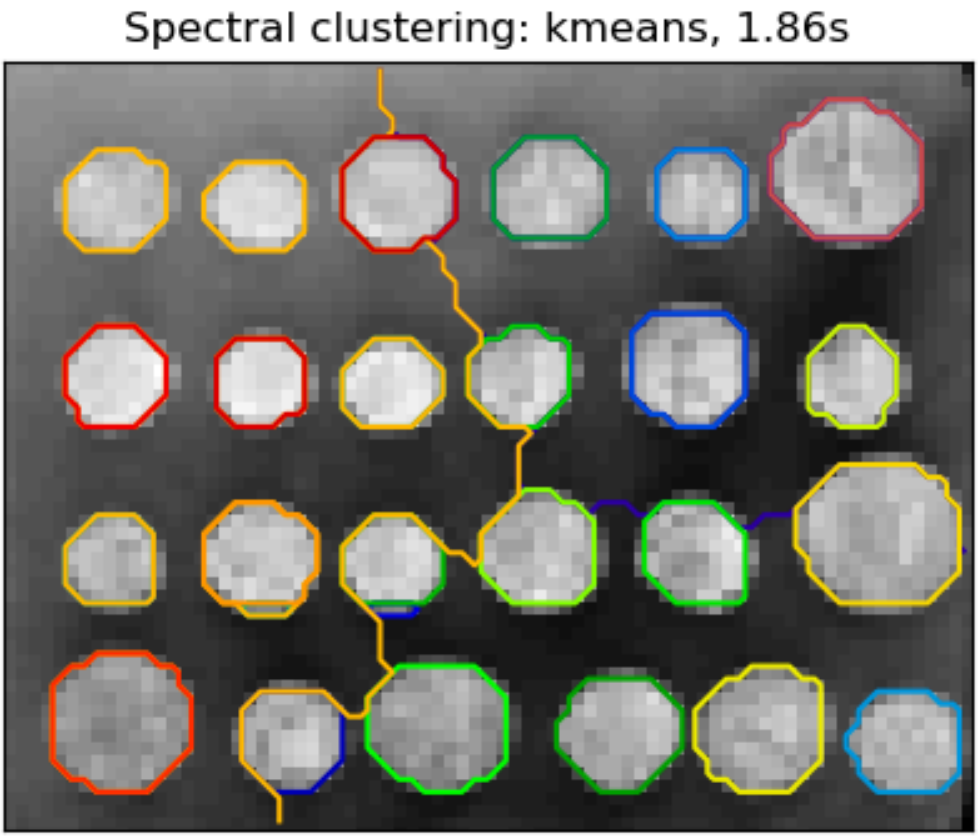
Спектральная кластеризация (spectral clustering)

- работает с матрицами сходства (affinity/similarity matrices) в отличие от матриц расстояния
- производится свертка (embedding) матрицы, после чего к уже этой свертке применяется другой алгоритм кластеризации (чаще всего — k-средние) —

поэтому спектральной кластеризации тоже нужно число кластеров (см. метод локтя)

- хорошо работает для разреженных матриц
- хорошо подходит для решения задачи сегментации (например, изображений)





Note

«Красивые» сходства:

$$similarity = np.exp(-beta * distance / distance.std())$$

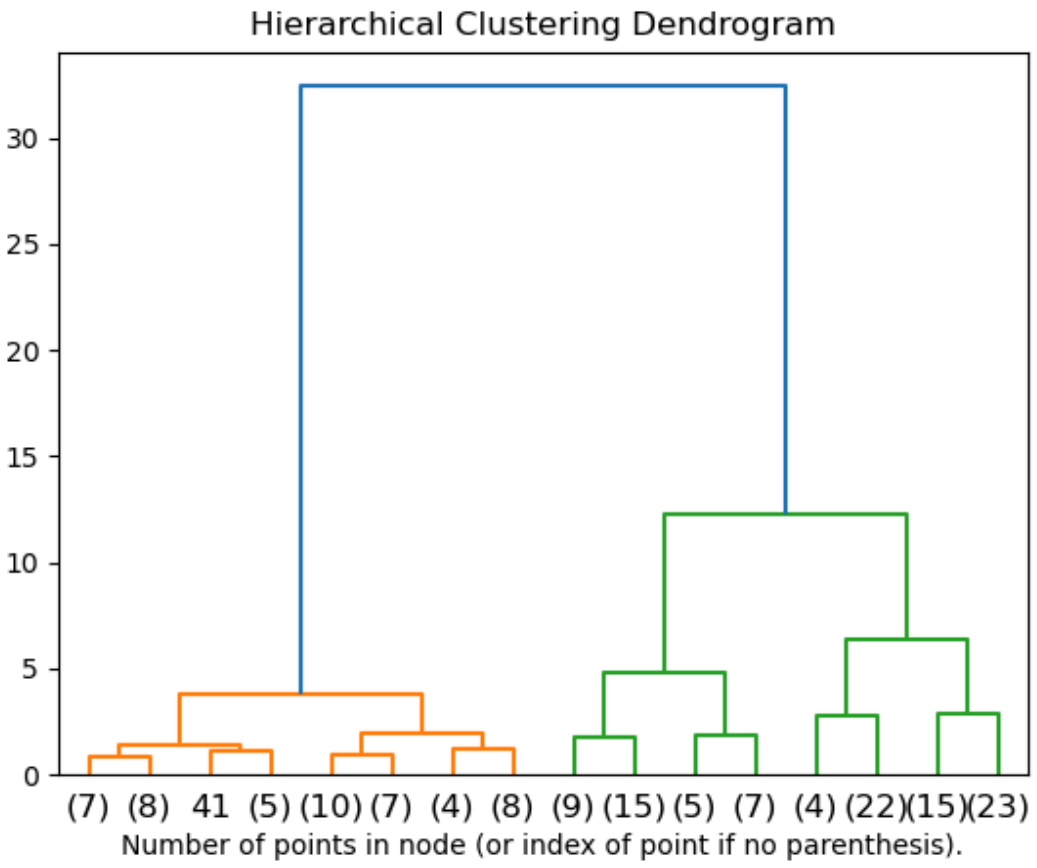
```
delta = 2 # delta is a free parameter representing the width of the
Gaussian kernel.
similarity_matrix = np.exp(- matrix ** 2 / (2. * delta ** 2))
```

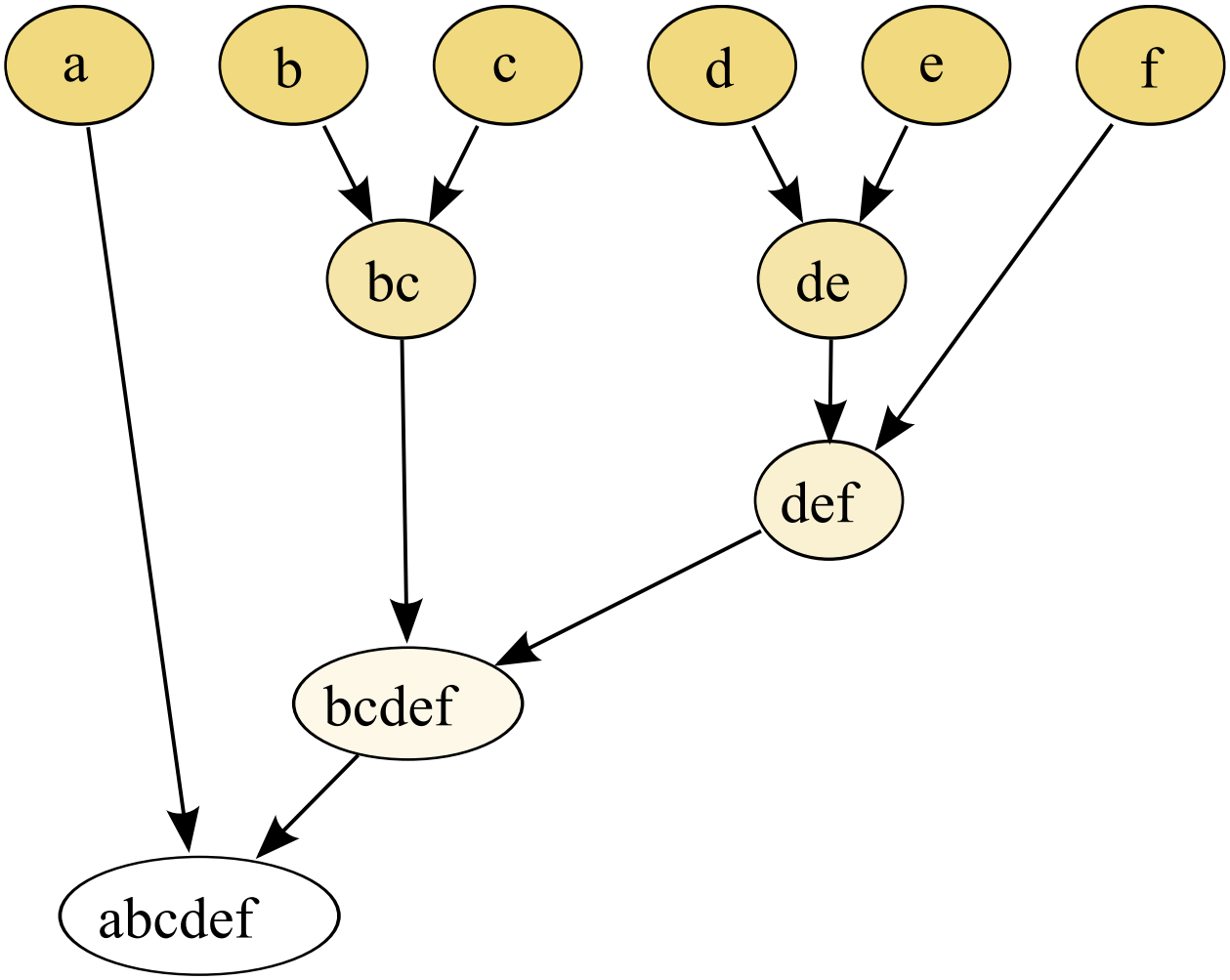
«Удобные» сходства (когда надо решить задачу быстро, но жертвуя «смыслом» меры — quick and dirty):

$$similarity = 1 - distance$$

Иерархическая кластеризация (hierarchical) / агломеративная (agglomerative)

- семейство алгоритмов, образующих иерархическую структуру из кластеров (более мелкие кластеры входят в более крупные)
- такая иерархия чаще всего визуализируется при помощи дендрограммы — древовидной диаграммы, причем «корневым элементом» такого дерева является единый общий кластер, объединяющий все элементы, тогда как ветви — кластеры, включающие в себя только некоторые элементы, а листья — непосредственно сами образцы (элементы)

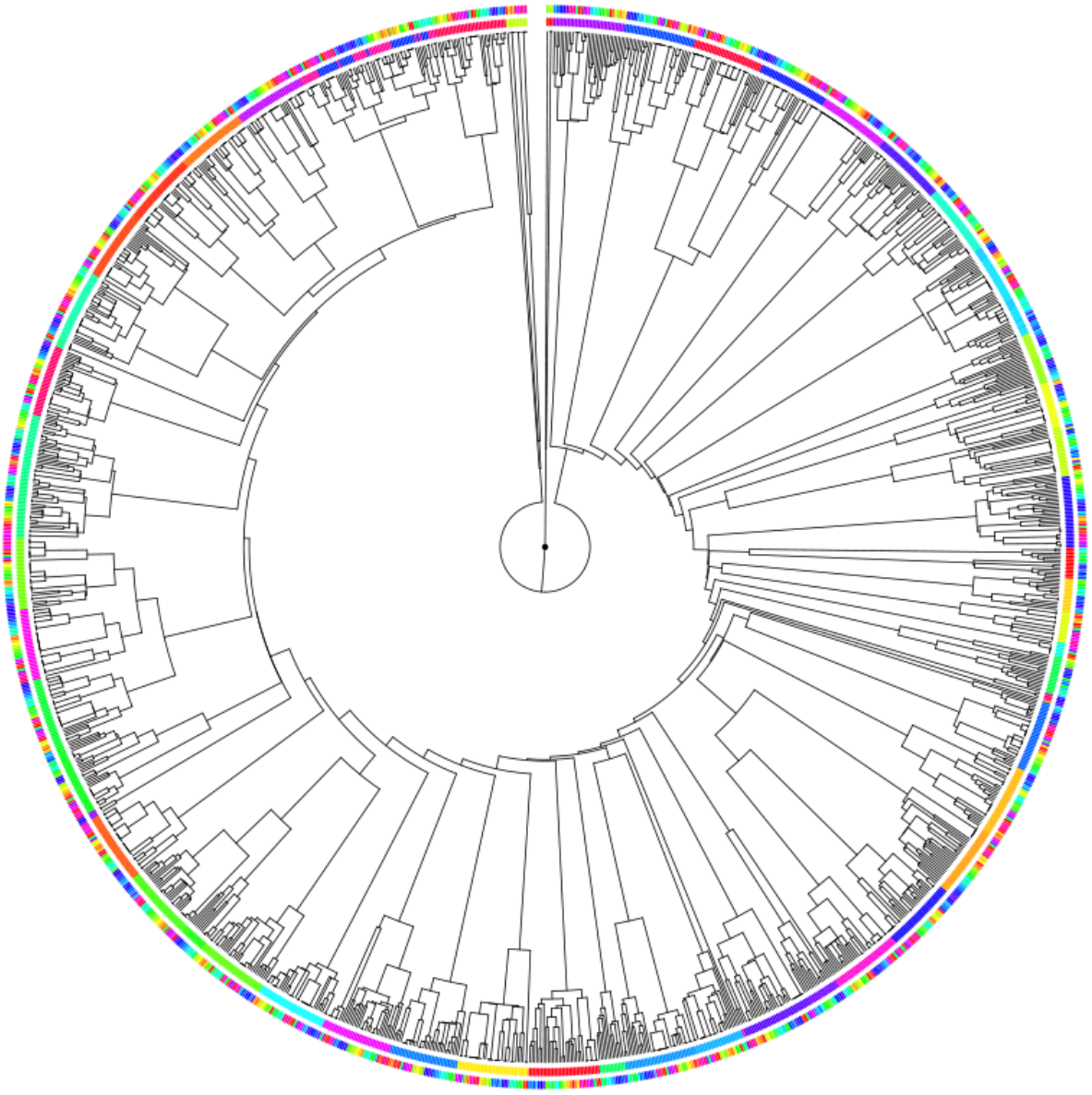




A Visual Expedition Inside the Linux File Systems - Linux Kernel 2.6.x — в примере несколько экспериментов с разными мерами расстояния (на иллюстрации — расстояние Канберры)

Info

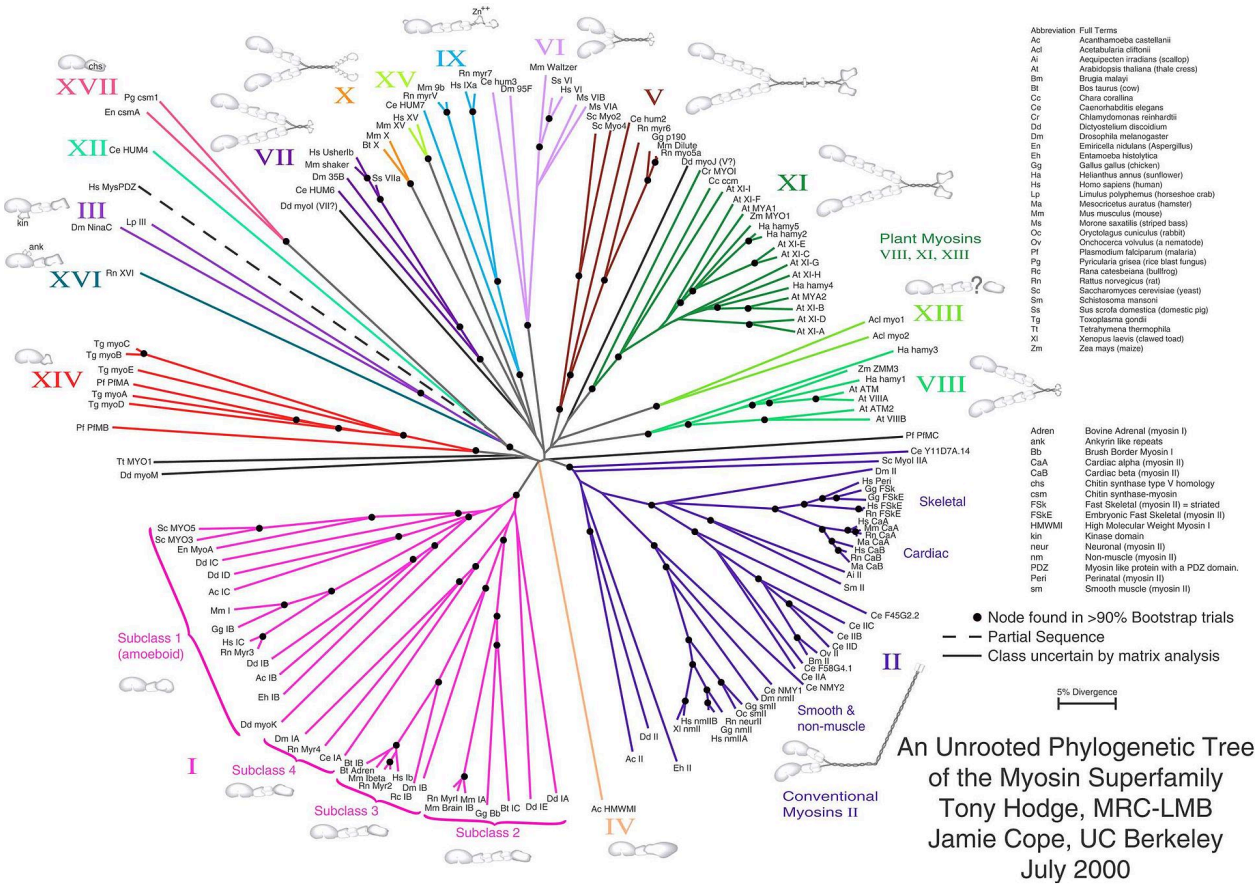
Расстояние Канберры — взвешенная версия Манхэттенского расстояния, представлена данная мера была в 1960-х годах



Аггломеративная кластеризация — частный случай иерархической кластеризации, когда мы движемся «снизу вверх»: от отдельных элементов к объединяющим их кластерам.

Если мы движемся «сверху вниз», то такая иерархическая кластеризация называется **разделительной** (divisive) — мы начинаем с «корня» и рекурсивно разделяем элементы по иерархии (ступеням)

Данные алгоритмы активно применяются в биологии и связанных отраслях научного знания, например, филогенетическое древо:

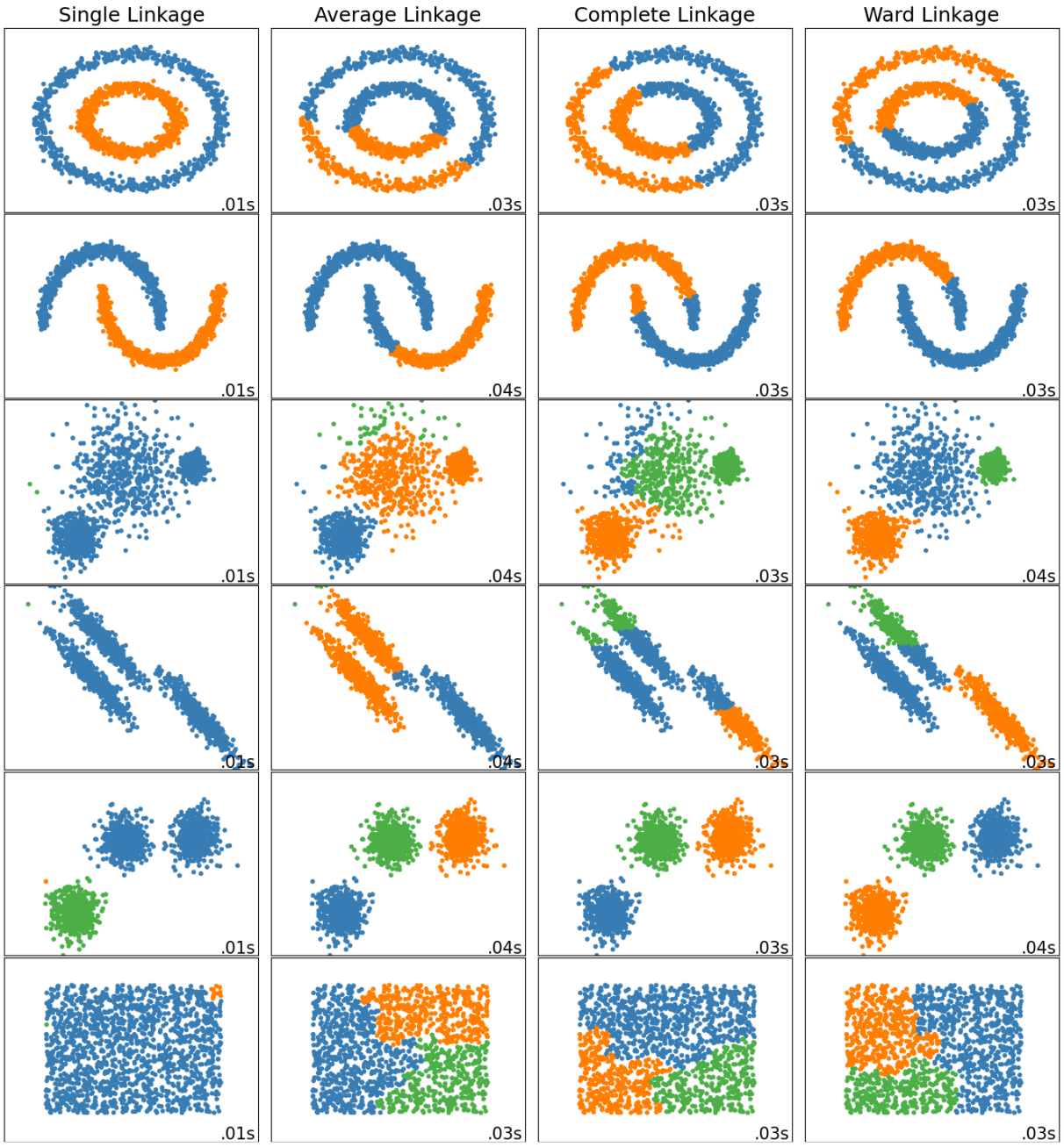


Алгоритмы иерархической кластеризации различаются тем, какие формулы для расчета связи между элементами используются:

- Метод Уорда (Минимальное увеличение суммы квадратов, MISSQ) (Ward clustering) — В качестве расстояния между кластерами берётся прирост суммы квадратов расстояний объектов до центра кластера, получаемого в результате их объединения. На каждом шаге алгоритма объединяются такие два кластера, которые приводят к минимальному увеличению дисперсии. Этот метод применяется для задач с близко расположенными кластерами.
- Метод полной связи (maximum/complete linkage) — расстояние между кластерами равно максимальному расстоянию между двумя элементами из данных кластеров, «метод дальнего соседа»
- Метод одиночной связи (single linkage) — расстояние между кластерами равно минимальному расстоянию между двумя элементами из данных кластеров, «метод ближайшего соседа»
- Метод средней связи (average linkage) — расстояние между кластерами равно средней дистанции между всеми элементами этих двух кластеров, бывает взвешенный (WPGMA) и невзвешенный (UPGMA) — отличаются тем, что в одном случае связи могут иметь коэффициент веса, что увеличивает или уменьшает их влияние на итоговое расстояние между кластерами

Сравнение разных походов к связям есть в примере в документации `scikit-learn`:

[Comparing different hierarchical linkage methods on toy datasets — scikit-learn 1.6.1 documentation]



Лабораторная работа №3. Решение задач классификации и кластеризации

1. Загрузить из наборов данных Scikit-learn набор «Ирисы Фишера» (https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html)

2. Ознакомиться с алгоритмами классификации (https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html) и кластеризации (https://scikit-learn.org/stable/modules/clustering.html, https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html).
3. Провести классификацию набора алгоритмом k-ближайших соседей. Визуализировать результат.
4. Провести классификацию набора алгоритмом «случайный лес». Визуализировать результат.
5. Провести классификацию набора машинами опорных векторов (SVM). Визуализировать результат.
6. Провести кластеризацию набора алгоритмом k-средних. Визуализировать результат.
7. Провести иерархическую кластеризацию методом Уорда. Визуализировать дендрограмму.
8. Провести спектральную кластеризацию. Визуализировать результат.
9. Используя набор данных о пульсарах HTRU2 (приложенный архив или https://archive.ics.uci.edu/ml/datasets/HTRU2), исследовать набор данных и реализовать классифицирующую модель любым подходящим методом (но не ИНС!).
    - рекомендуется в числе прочего использовать `pairplot` (библиотека `seaborn`) для визуального исследования датасета
    - при решении важно отметить и описать особенности набора данных
    - в ходе решения задачи нужно также ответить на вопрос, достаточна ли точность в 98% для такого набора данных
10. Используя набор публикаций о здоровье и медицине (приложенный архив или https://archive.ics.uci.edu/ml/datasets/Health+News+in+Twitter), исследовать набор данных и реализовать кластеризующую модель любым подходящим методом (но не ИНС!).