Лекция 8. Задачи кластеризации

Кластеризация является задачей разбиения набора данных на группы, называемые кластерами. Цель – разделить данные таким образом, чтобы точки, находящие в одном и том же кластере, были очень схожи друг с другом, а точки, находящиеся в разных кластерах, отличались друг от друга. Как и алгоритмы классификации, алгоритмы кластеризации присваивают (или прогнозируют) каждой точке данных номер кластера, которому она принадлежит.

Задача кластеризации относится к широкому классу задач обучения без учителя. Кластеризацию применяют для анализа и поиска признаков по которым можно объединить объекты, сжатия данных и поиска новизны (что не входит ни в один кластер)

В чем отличие классификации и кластеризации: при классификации у вас есть набор предопределенных классов, вы обучаете ИИ на наборе примеров и потом хотите знать, к какому классу принадлежит новый объект. При кластеризации вы используете алгоритм, который пытается сгруппировать набор объектов и определить, существует ли какая-либо взаимосвязь между объектами.

Интуитивная постановка задачи кластеризации довольно проста и представляет из себя наше желание сказать: "Вот тут у меня насыпаны точки. Я вижу, что они сваливаются в какие-то кучки вместе. Было бы здорово иметь возможность эти точки относить к кучкам и в случае появления новой точки на плоскости говорить, в какую кучку она падает." Из такой постановки видно, что пространства для фантазии получается много, и от этого возникает соответствующее множество алгоритмов решения этой задачи. Перечисленные алгоритмы ни в коем случае не описывают данное множество полностью, но являются примерами самых популярных методов решения задачи кластеризации.

Техника кластеризации применяется в самых разнообразных областях. Главное задача – разбить многомерный ряд исследуемых значений (объектов, переменных, признаков) на однородные группы, кластеры. То есть данные классифицируются и структурируются.

Вопрос, который задает исследователь при использовании кластерного анализа, – как организовать многомерную выборку в наглядные структуры.

Примеры использования кластерного анализа:

1. В биологии – для определения видов животных на Земле.
2. В медицине – для классификации заболеваний по группам симптомов и способам терапии.
3. В психологии – для определения типов поведения личности в определенных ситуациях.
4. В экономическом анализе – при изучении и прогнозировании экономической депрессии, исследовании конъюнктуры.
5. В разнообразных маркетинговых исследованиях.

Когда нужно преобразовать «горы» информации в пригодные для дальнейшего изучения группы, используют кластерный анализ. Далее приведем основные алгоритмы кластерного анализа.

Метод *k*-средних

Кластеризация *k*-средних – один из самых простых и наиболее часто используемых алгоритмов кластеризации. Сначала выбирается число кластеров *k*. После выбора значения *k* алгоритм *k*-средних отбирает точки, которые будут представлять ***центры кластеров (cluster centers)***. Затем для каждой точки данных вычисляется его евклидово расстояние до каждого центра кластера. Каждая точка назначается ближайшему центру кластера. Алгоритм вычисляет ***центроиды (centroids)*** – центры тяжести кластеров. Каждый центроид – это вектор, элементы которого представляют собой средние значения характеристик, вычисленные по всем точкам кластера. Центр кластера смещается в его центроид. Точки заново назначаются

ближайшему центру кластера. Этапы изменения центров кластеров и переназначения точек итеративно повторяются до тех пор, пока границы кластеров и расположение центроидов не перестанут изменяться, т.е. на каждой итерации в каждый кластер будут попадать одни и те же точки данных. Следующий пример иллюстрирует работу алгоритма на синтетическом наборе данных

Центры кластеров представлены в виде треугольников, в то время как точки данных отображаются в виде окружностей. Цвета указывают принадлежность к кластеру. Мы указали, что ищем три кластера, поэтому алгоритм был инициализирован с помощью случайного выбора трех точек данных в качестве центров кластеров (см. «Инициализация»). Затем запускается итерационный алгоритм. Во-первых, каждая точка данных назначается ближайшему центру кластера (см. «Назначение точек (1)»). Затем центры кластеров переносятся в центры тяжести кластеров (см. «Пересчет центров (1)»). Затем процесс повторяется еще два раза. После третьей итерации принадлежность точек кластерным центрам не изменилась, поэтому алгоритм останавливается (рисунок 1).

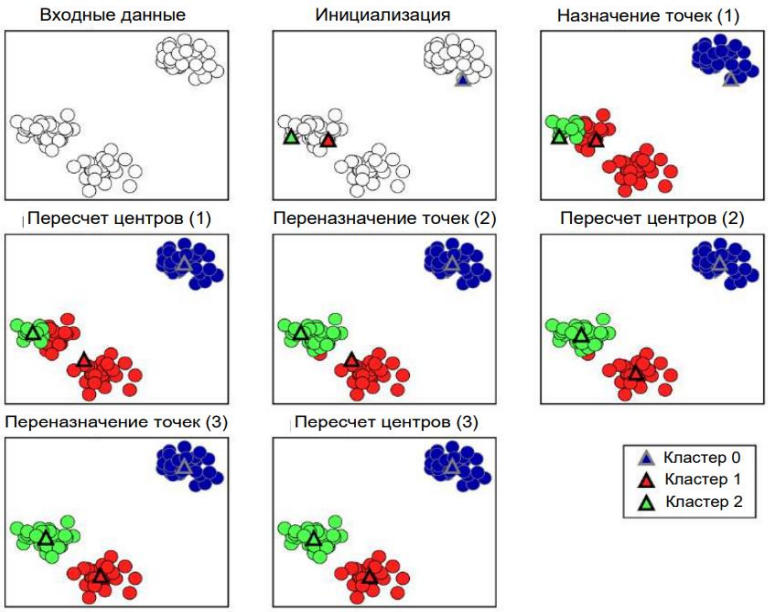

Рисунок 1 –Пример работы алгоритма метода k-средних

Получив новые точки данных, алгоритм k-средних будет присваивать каждую точку данных ближайшему центру кластера. Пример на рисунке 2 показывает границы центров кластеров, процесс вычисления которых был приведен на рисунке 1.

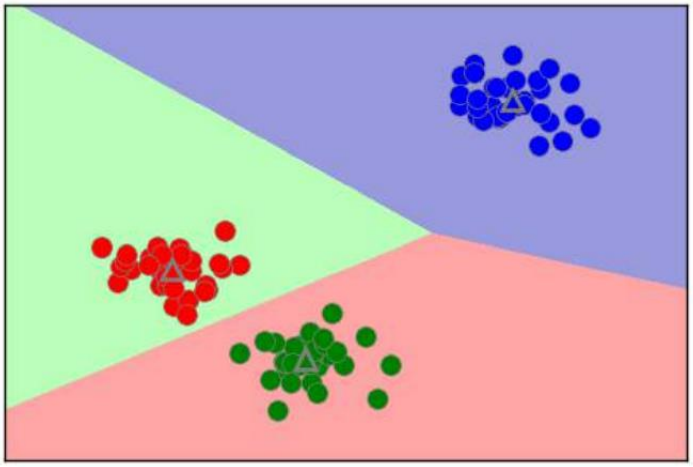

Рисунок 2 –Границы кластеров

Алгоритм *k*-средних, наверное, самый популярный и простой алгоритм кластеризации и очень легко представляется в виде простого псевдокода:

1. Выбрать количество кластеров *k*, которое нам кажется оптимальным для наших данных.
2. Высыпать случайным образом в пространство наших данных *k* точек (центроидов).
3. Для каждой точки нашего набора данных посчитать, к какому центроиду она ближе.
4. Переместить каждый центроид в центр выборки, которую мы отнесли к этому центроиду.
5. Повторять последние два шага фиксированное число раз, либо до тех пор пока центроиды не "сойдутся" (обычно это значит, что их смещение относительно предыдущего положения не превышает какого-то заранее заданного небольшого значения).

Стоит заметить, что можно рассчитывать расстояние между центройдами по любой метрике (Евклидовой, Хемминговой и т.д.).

К недостаткам алгоритма можно отнести следующее. Даже если вы знаете «правильное» количество кластеров для конкретного набора данных, алгоритм k-средних не всегда может выделить их. Каждый кластер определяется исключительно его центром, это означает, что каждый кластер имеет выпуклую форму. В результате этого алгоритм k-средних может описать относительно простые формы. Кроме того, алгоритм k-средних предполагает, что все кластеры в определенном смысле имеют одинаковый «диаметр», он всегда проводит границу между кластерами так, чтобы она проходила точно посередине между центрами кластеров. Это иногда может привести к неожиданным результатам.

Следующий график показывает двумерный набор данных с тремя четко обособленными группами данных. Однако эти группы вытянуты по диагонали. Поскольку алгоритм k-средних учитывает лишь расстояние до ближайшего центра кластера, он не может обработать данные такого рода. Кроме того, алгоритм k-средних плохо работает, когда кластеры имеют более сложную форму (рисунок 3).

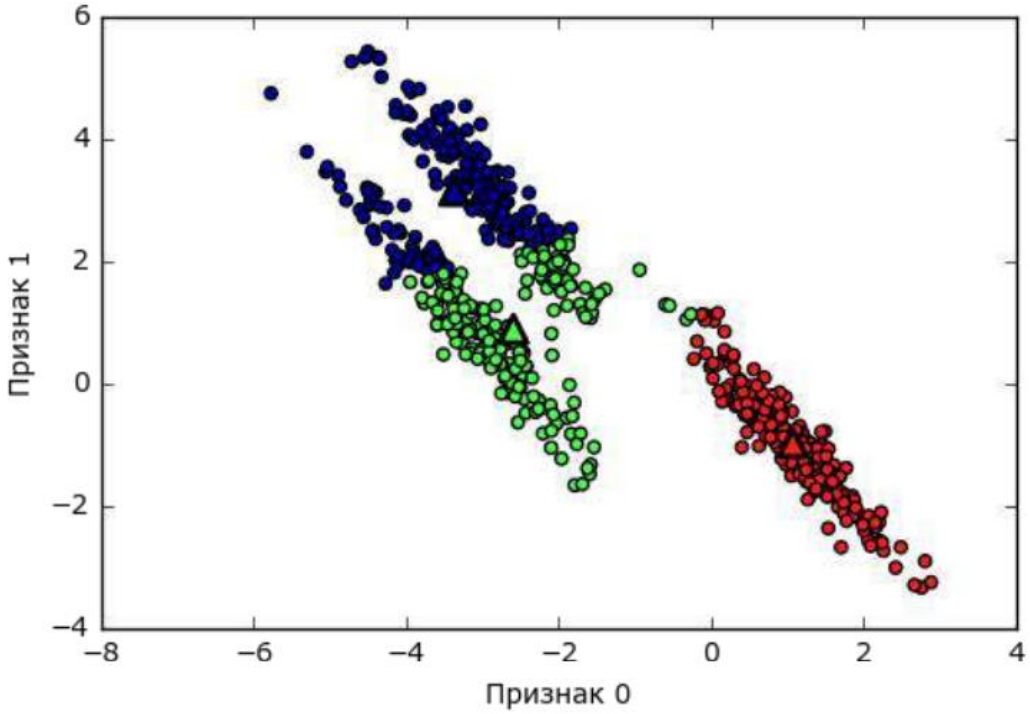

Рисунок 3 –Пример работы алгоритма метода k-средних

Пример

Для примера возьмем шесть объектов наблюдения. Каждый имеет два характеризующих его параметра.

	A	B	C
1	№ п/п	x	y
2	1	2	8
3	2	4	10
4	3	5	7
5	4	12	6
6	5	14	6
7	6	15	4

Рисунок 4 – Начальные данные

В качестве расстояния между объектами возьмем евклидовое расстояние. Формула расчета представлена на рисунке 5.

G2 =КОРЕНЬ((B3-B2)^2+(C3-C2)^2)

	A	B	C	D	E	F	G	H	I	J	K
1	№ п/п	x	y		№ п/п	1	2	3	4	5	6
2	1	2	8		1	0	2,83	3,16	10,20	12,17	13,60
3	2	4	10		2	2,83	0	3,16	8,94	10,77	12,53
4	3	5	7		3	3,16	3,16	0	7,07	9,06	10,44
5	4	12	6		4	10,20	8,94	7,07	0	2,00	3,61
6	5	14	6		5	12,17	10,77	9,06	2,00	0	2,24
7	6	15	4		6	13,60	12,53	10,44	3,61	2,24	0

Рисунок 5 – Формула расчета в Excel

Рассчитанные данные размещаем в матрице расстояний.

Самыми близкими друг к другу объектами являются объекты 4 и 5. Следовательно, их можно объединить в одну группу – при формировании новой матрицы оставляем наименьшее значение (рисунок 6).

E	F	G	H	I	J
№ п/п	1	2	3	[4,5]	6
1	0	2,83	3,16	10,20	13,60
2	2,83	0	3,16	8,94	12,53
3	3,16	3,16	0	7,07	10,44
[4,5]	10,20	8,94	7,07	0	2,24
6	13,60	12,53	10,44	2,24	0

Рисунок 6 – Объекты 4 и 5

Из новой матрицы видно, что можно объединить в один кластер объекты [4, 5] и 6 (как наиболее близкие друг к другу по значениям). Оставляем наименьшее значение и формируем новую матрицу (рисунок 7)

E	F	G	H	I
№ п/п	1	2	3	[4,5,6]
1	0	2,83	3,16	10,20
2	2,83	0	3,16	8,94
3	3,16	3,16	0	7,07
[4,5,6]	10,20	8,94	7,07	0

Рисунок 7 – Объединение в классы

Объекты 1 и 2 можно объединить в один кластер (как наиболее близкие из имеющихся). Выбираем наименьшее значение и формируем новую матрицу расстояний. В результате получаем три кластера (рисунок 8).

E	F	G	H
№ п/п	[1,2]	3	[4,5,6]
[1,2]	0	3,16	8,94
3	3,16	0	7,07
[4,5,6]	8,94	7,07	0

Рисунок 8 – Кластеры согласно матрице состоянии

Самые близкие объекты – 1, 2 и 3. Объединим их (рисунок 9).

E	F	G
№ п/п	[1,2,3]	[4,5,6]
[1,2,3]	0	7,07
[4,5,6]	7,07	0

Рисунок 9 – Объединение

Мы провели кластерный анализ по методу «ближайшего соседа». В результате получено два кластера, расстояние между которыми – 7,07. Метод можно распространить и на большее количество соседей.

Кластерный анализ представляет большое значение для экономического анализа. Инструмент позволяет вычленять из большой совокупности периоды, где значения соответствующих параметров максимально близки и где динамика наиболее схожа. Для исследования, к примеру, товарной и общехозяйственной конъюнктуры этот метод отлично подходит.

Агломеративная кластеризация

Алгомеративная кластеризация относится к семейству алгоритмов кластеризации, в основе которых лежат одинаковые принципы: алгоритм начинает свою работу с того, что каждую точку данных заносит в свой собственный кластер и по мере выполнения объединяет два наиболее схожих между собой кластера до тех пор, пока не будет удовлетворен определенный критерий остановки. Зачастую данным критерием выступает это количество кластеров, поэтому схожие между собой кластеры объединяются до тех пор, пока не останется заданное число кластеров. Следующий график иллюстрирует работу алгоритма агломеративной кластеризации на двумерном массиве данных, который ищет три кластера (рисунок 10).

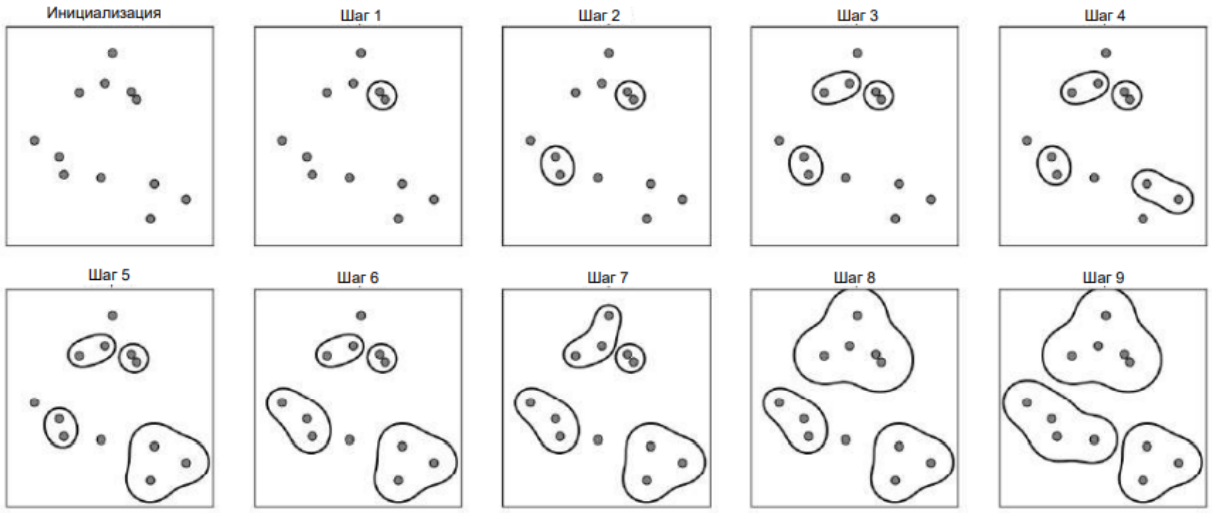

Рисунок 10 –Пример работы алгоритма иерархической кластеризации

Изначально количество кластеров равно количеству точек данных. Затем на каждом шаге объединяются два ближайших друг к другу кластера. На первых четырех шагах выбираются кластеры, состоящие из отдельных точек, и объединяются в кластеры, состоящие из двух точек. На шаге 5 один из 2-точечных кластеров вбирает в себя третью точку и т.д. На шаге 9 у нас остается три кластера. Поскольку мы установили количество кластеров равным 3, алгоритм останавливается.

Иерархическая кластеризация

Результатом агломеративной кластеризации является ***иерархическая кластеризация***. Кластеризация выполняется итеративно, и каждая точка совершает путь от отдельной точки-кластера до участника итогового кластера. На каждом промежуточном шаге происходит кластеризация данных (с разным количеством кластеров). Иногда полезно сразу взглянуть на все возможные кластеризации. Следующий пример показывает наложение всех возможных кластеризаций, показанных на рис. и дает некоторое представление о том, как каждый кластер распадается на более мелкие кластеры (рисунок 11).

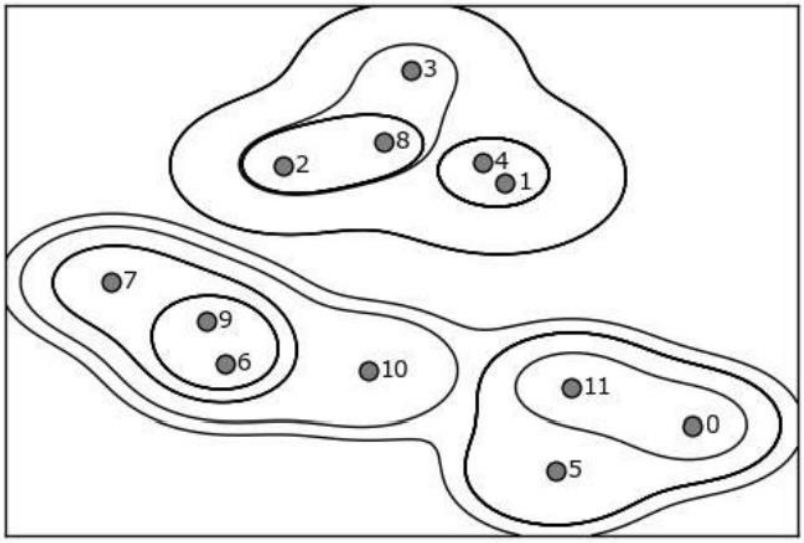

Рисунок 11 – Иерархическое присвоение кластеров (показаны в виде линий), полученное с помощью алгоритма агломеративной кластеризации, точки данных пронумерованы

Хотя эта визуализация дает достаточно детализированное представление о результатах иерархической кластеризации, она опирается на двумерную природу данных и не может быть использована для наборов данных, которые имеют более двух характеристик. Однако есть еще один инструмент для визуализации результатов иерархической кластеризации, называемый ***дендрограммой (dendrogram)*** и позволяющий обрабатывать многомерные массивы данных (рисунок 12).

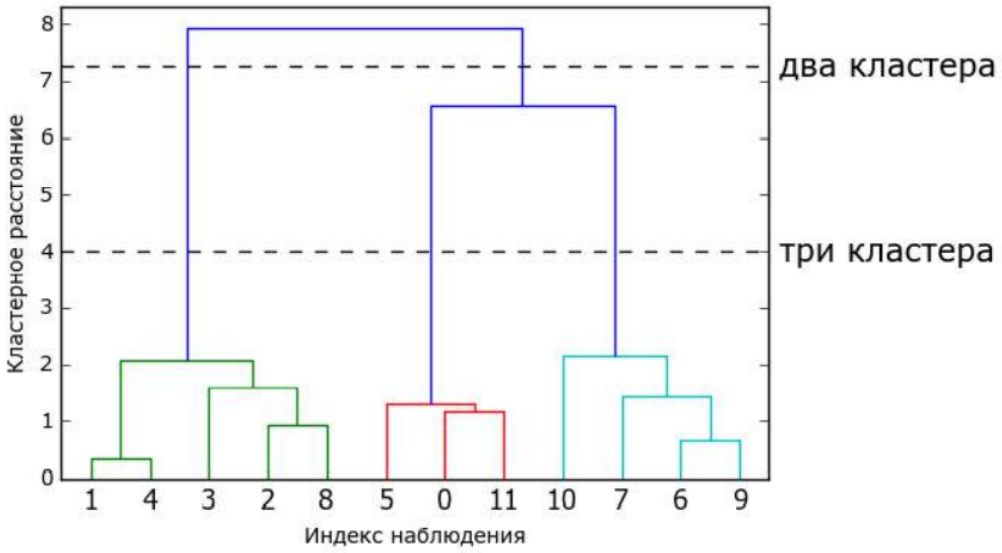

Рисунок 12 – Пример дендрограммы

Точки данных показаны в нижней части дендрограммы (пронумерованы от 0 до 11). Затем строится дерево с этими точками (представляющими собой кластеры-точки) в качестве листьев, и для каждых двух объединенных кластеров добавляется новый узел-родитель. Чтение дендрограммы происходит снизу вверх. Точки данных 1 и 4 объединяются первыми (как вы уже могли видеть на рис). Затем в кластер объединяются точки 6 и 9 и т.д. На самом верхнем уровне остаются две ветви, одна ветвь состоит из точек 11, 0, 5, 10, 7, 6 и 9, а вторая – из точек 1, 4, 3, 2 и 8. Они соответствуют двум крупнейшим кластерам.

Ось y в дендрограмме указывает не только момент объединения двух кластеров в ходе работы алгоритма агломеративной кластеризации. Длина каждой ветви показывает, насколько далеко друг от друга находятся объединенные кластеры. Самыми длинными ветвями в этой дендрограмме являются три линии, отмеченные пунктирной чертой с надписью «три кластера». Тот факт, что эти линии являются самыми длинными ветвями, указывает на то, что переход от трех кластеров к двум сопровождался объединением некоторых сильно удаленных друг от друга точек. Мы снова видим это в самой верхней части графика, когда объединение двух оставшихся

кластеров в единый кластер подразумевает относительно большое расстояние между точками.

Применение алгоритмов кластеризации с последующей оценкой их результатов является сложной и, как правило, очень полезной процедурой на исследовательском этапе анализа данных. Мы рассмотрели два алгоритма кластеризации: *k*-средние и агломеративную кластеризацию. Оба алгоритма имеют возможность настраивать гранулярность кластеризации. Алгоритмы k-средних и агломеративной кластеризации позволяют задать нужное количество кластеров. Оба метода могут быть использованы на больших реальных наборах данных, имеют относительно простую интерпретацию и допускают разбиение на большое количество кластеров.

Контрольные вопросы по теме

1. Для чего используется кластерный анализ?
2. В чем отличие классификации от кластеризации?
3. Что такое кластер?
4. Что получается в результате кластерного анализа?
5. В чем суть метода k-средних?
6. В чем заключается суть агломеративной кластеризации?
7. Каким образом можно определять расстояния между классами в кластерном анализе?
8. Характеристики близости объектов и показателей в кластерном анализе.
9. Иерархические кластер-процедуры.