***Алгоритм k*-*means***

Сегодня предложено несколько десятков алгоритмов кластеризации и еще больше их разновидностей. Несмотря на это, в *Data* *Mining* применяются в первую очередь понятные и простые в использовании алгоритмы. К таким относится алгоритм *k*-*means* — в русскоязычном варианте *k*-средних (от англ. *mean* — «среднее значение»). Его основная идея состоит в том, что для выборки данных, содержащей *n* записей (объектов), задается число кластеров — *k*, которое должно быть сформировано. Затем алгоритм разбивает все объекты выборки на *k* разделов (*k* < *n*), которые и представляют собой кластеры.

*Алгоритм выполняется в четыре шага:*

1. задается число кластеров — *k*, которое должно быть сформировано из объектов исходной выборки;
2. случайным образом выбирается *k* записей исходной выборки, которые будут служить начальными центрами кластеров. Начальные точки, из которых потом вырастает кластер, часто называют «семенами» (от англ. *seeds* — «семена», «посевы»). Каждая такая запись представляет собой своего рода «эмбрион» кластера, состоящий только из одного элемента;
3. для каждой записи исходной выборки определяется ближайший к ней центр кластера. Чтобы определить, в сферу влияния какого центра кластера входит та или иная запись, вычисляется расстояние от каждой записи до каждого центра в многомерном пространстве признаков и выбирается то «семя», для которого данное расстояние минимальное;
4. в анализе данных распространенной оценкой близости между объектами является *метрика*, или способ задания расстояния. Выбор конкретной метрики зависит от аналитика и конкретной задачи. Наиболее популярные метрики — евклидово расстояние и расстояние Манхэттена.

*Евклидово расстояние*, или метрика L2, применяется для вычисления расстояний следующее правило по формуле:

,

где *X* = (*x*1, *x*2, …, *xm*), *Y* = (*y*1, *y*2, ..., *ym*) — векторы значений признаков двух записей.

Поскольку множество точек, равноудаленных от некоторого центра, при использовании евклидовой метрики будут образовывать *сферу* (или круг в двумерном случае), то кластеры, полученные с использованием евклидова расстояния, также будут иметь форму, близкую к сферической.

Расстояние Манхеттена, или метрика L1, вычисляется по формуле:

.

Фактически *расстояние Манхеттена* — кратчайшее расстояние между двумя точками, пройденное по линиям, параллельным осям координатой системы. Преимущество метрики L1 заключается в том, что она позволяет снизить влияние аномальных значений на работу алгоритмов. Кластеры, построенные на основе расстояния Манхеттена, стремятся к *кубической* форме.

Используя метрики L1 или L2, для каждой записи исходной выборки определяется ближайший к ней центр (*центроид*) кластера.

Например, если в кластер вошли три записи с наборами признаков (*x*1, *y*1), (*x*2, *y*2), (*x*3, *y*3), то координаты его центроида по метрике L1 будут рассчитываться следующим образом:

;

1. старый центр кластера смещается в его центроид.

Таким образом, центроиды становятся новыми центрами кластеров для следующего итерации алгоритма. Шаги 3 и 4 повторяются до тех пор, пока выполнение алгоритма не будет прервано или пока не будет выполнено условие в соответствии с некоторым *критерием сходимости*.

Остановка алгоритма производится, когда границы кластеров и расположение центроидов перестают изменяться от итерации к итерации, т.е. на каждой итерации в каждом кластере остается один и тот же набор записей. Алгоритм *k*-*means* обычно позволяет находить набор стабильных кластеров за несколько десятков итераций.

Что касается *критерия сходимости*, то чаще всего используется сумма квадратов ошибок между центроидом кластера и всеми вошедшими в него записями:

,

где *p* ∈ *Ci* — произвольная точка данных, принадлежащих кластеру *Ci*; *mi* — центроид данного кластера.

Иными словами, алгоритм остановится тогда, когда ошибка *E* достигнет достаточно малого.

Один из основных недостатков, присущих алгоритму *k*-*means*, — отсутствие четких критериев выбора числа кластеров, целевой функции их инициализации и модификации. Кроме того, он очень чувствителен к шумам в данных и аномальным значениям, поскольку они способны существенно повлиять на среднее значение, используемое при вычислении положений центроидов. Чтобы снизить влияние таких факторов, как шумы и аномальные значения, иногда на каждой итерации используют не среднее значение признаков, а их медиану. Данная модификация алгоритма называется *k*-*mediods* (*k*-медиан).

***Пример работы алгоритма k-means***

Пусть имеется набор из восьми точек данных в двумерном пространстве, из которого требуется получить два кластера. Значения точек приведены в табл. 5.1 ина рис. 5.4.

*Таблица 5.1*

**Объекты для кластеризации**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *А* | *В* | *С* | *D* | *Е* | *F* | *G* | *Н* |
| (1;3) | (3;3) | (4;3) | (5;3) | (1;2) | (4; 2) | (1;1) | (2;1) |



*Рис. 5.1*. Процедура начальной инициализации: значения буллитов приведены в табл. 5.1

*Шаг 1.* Определим число кластеров, на которое требуется разбить исходное множество: *k* = 2.

*Шаг 2.* Случайным образом выберем две точки, которые будут начальными центрами кластеров. Пусть это будут точки *m*1 = (1;1) и *m*2 = (2:1). На рис. 5.1 они представлены ромбами.

*Шаг 3, проход 1*. Для каждой точки определим ближайший к ней центр кластера с помощью евклидова расстояния. В табл. 5.2 представлены вычисленные с помощью формулы расстояния между центрами кластеров *m*1 = (1;1) и *m*2 = (2:1) и каждой точкой исходного множества и указано, к какому кластеру принадлежит та или иная точка (табл. 5.2).

*Таблица 5.2*

**Нахождение ближайшего центра для каждой точки (первый проход)**

|  |  |  |  |
| --- | --- | --- | --- |
| Точка | Расстояние от *m*1 | Расстояние от *m*2 | Принадлежит кластеру |
| *А* | 2,00 | 2,24 | 1 |
| *В* | 2,83 | 2,24 | 2 |
| *С* | 3,61 | 2,83 | 2 |
| *D* | 4,47 | 3,61 | 2 |
| *Е* | 1,00 | 1,41 | 1 |
| *F* | 3,16 | 2,24 | 2 |
| *G* | 0,00 | 1,00 | 1 |
| *Н* | 1,00 | 0,00 | 2 |

Таким образом, кластер 1 содержит точки *А*, *Е*, *G*, а кластер 2 — точки *В*, *С*, *D*, *F*, *Н*. Как только определятся члены кластеров, может быть рассчитана сумма квадратов ошибок:

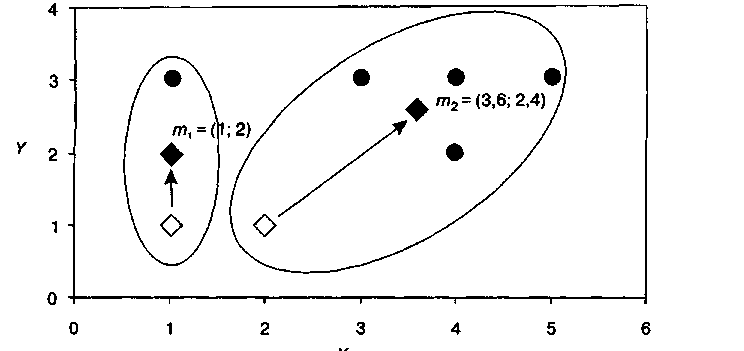
.

*Шаг 4, проход 1.*Для каждого кластера вычисляется центроид, и в него перемещается центр кластера.

Центроид для кластера 1: [(1 + 1 + 1) / 3, (3 + 2 + 1) / 3] = (1; 2).

Центроид для кластера 2: [(3 + 4 + 5 + 4 + 2) / 5, (3 + 3 + 3 + 2 + 1) / 5] = (3,6; 2,4).

Расположение кластеров и центроидов после первого прохода алгоритма представлено на рис. 5.2.



*Рис. 5.2*. Расположение кластеров и центроидов после первого прохода алгоритма

Здесь начальные центры кластеров представлены светлыми ромбами, а центроиды, вычисленные при первом проходе алгоритма, — темными ромбами. Они и станут новыми центрами кластеров, принадлежность точек данных к которым будет определяться на втором проходе.

*Шаг 3, проход 2.* После того, как найдены новые центры кластеров, для каждой точки снова определяется ближайший к ней центр и ее отношение к соответствующему кластеру. Для этого еще раз вычисляются евклидовы расстояния между точками и центрами кластеров. Результаты вычислений приведены в табл. 5.3.

*Таблица 5.3*

**Нахождение ближайшего центра для каждой точки (второй проход)**

|  |  |  |  |
| --- | --- | --- | --- |
| Точка | Расстояние от *m*1 | Расстояние от *m*2 | Принадлежит кластеру |
| *А* | 1,00 | 2,67 | 1 |
| *В* | 2,24 | 0,85 | 2 |
| *С* | 3,16 | 0,72 | 2 |
| *D* | 4,12 | 1,52 | 2 |
| *Е* | 0,00 | 2,63 | 1 |
| *F* | 3,00 | 0,57 | 2 |
| *G* | 1,00 | 2,95 | 1 |
| *Н* | 1,41 | 2,13 | 1 |

Относительно большое изменение *m*2 привело к тому, что запись *H* оказалась ближе к центру *m*1, что автоматически сделало ее членом кластера 1. Все остальные записи остались в тех же кластерах, что и на предыдущем проходе алгоритма. Таким образом, кластер 1 будет содержать точки *A*, *E*, *G*, *H*, а кластер 2 — *B*, *C*, *D*, *F*. Новая сумма квадратов ошибок составит:

Вычисление показывает уменьшение ошибки в сравнении с начальным состоянием центров кластеров (на первом проходе она составляла 36). Это говорит об улучшении качества кластеризации, т.е. о более высокой «кучности» объектов относительно центра кластера.

*Шаг 4, проход 2.*Для каждого кластера вновь вычисляется центроид, и в него перемещается центр кластера.

Новый центроид для кластера 1: [(1 + 1 + 1 + 2) / 4, (3 + 2 + 1 + 1) / 4] = (1,25; 1,75).

Новый центроид для кластера 2: [(3 + 4 + 5 + 4) /4, (3 + 3 + 3 + 2) / 4] = (4; 2,75).

Расположение кластеров и центроидов после второго прохода алгоритма представлено на рис. 5.3.



*Рис. 5.3*. Расположение кластеров и центроидов после второго прохода алгоритма

По сравнению с предыдущим проходом центры кластеров изменились незначительно.

*Шаг 3, проход 3.*Для каждой записи вновь ищется ближайший к ней центр кластера. Полученные на данном проходе расстояния представлены в табл. 5.4.

*Таблица 5.4*

**Нахождение ближайшего центра для каждой точки (третий проход)**

|  |  |  |  |
| --- | --- | --- | --- |
| Точка | Расстояние от *m*1 | Расстояние от *m*2 | Принадлежит кластеру |
| *А* | 1,27 | 3,01 | 1 |
| *В* | 2,15 | 1,03 | 2 |
| *С* | 3,02 | 0,25 | 2 |
| *D* | 3,95 | 1,03 | 2 |
| *Е* | 0,35 | 3,09 | 1 |
| *F* | 2,76 | 0,75 | 2 |
| *G* | 0,79 | 3,47 | 1 |
| *Н* | 1,06 | 2,66 | 1 |

Следует отметить, что записей, сменивших кластер на третьем проходе алгоритма, не было. Новая сумма квадратов ошибок составит:

Таким образом, сумма квадратов ошибок изменилась незначительно по сравнению с предыдущим проходом.

*Шаг 4, проход 3.*Для каждого кластера вновь вычисляется центроид, и центр кластера в него перемещается. Но поскольку на данном проходе ни одна запись не изменила своего членства в кластерах и положение центроидов не поменялось, алгоритм завершает работу.

Алгоритм *k*-*means* приобрел популярность благодаря следующим свойствам. Один из основных недостатков, присущих алгоритму *k*-*means*, — отсутствие четких критериев выбора числа кластеров, целевой функции их инициализации и модификации. Кроме того, он очень чувствителен к «шумам» в данных и аномальным значениям, поскольку они способны существенно повлиять на среднее значение, используемое при вычислении положений центройдов. Чтобы снизить влияние таких факторов, как шумы и аномальные значения, иногда на каждой итерации используют не среднее значение признаков, а их медиану. Данная модификация алгоритма называется *k*-*mediods* (*k*-медиан).