**9 слайд**

… после зачтения текста на слайде

Это делается путем простого определения среднего для значений каждого признака для всех записей в кластере.

**10 слайд**

Например, если в кластер вошли три записи с наборами признаков (х1, у1), (х2, у2), (х3, у3), то координаты его центроида будут рассчитываться следующим образом:

Затем старый центр кластера смещается в его центроид. Таким образом, центроиды становятся новыми центрами кластеров для следующей итерации алгоритма.

Шаги 3 и 4 повторяются до тех пор, пока выполнение алгоритма не будет прервано либо не будет выполнено условие в соответствии с некоторым критерием сходимости.

**11 слайд**

Остановка алгоритма производится тогда, когда границы кластеров и расположения центроидов не перестанут изменяться от итерации к итерации, т.е. на каждой итерации в каждом кластере будет оставаться один и тот же набор записей. На практике алгоритм k-means обычно находит набор стабильных кластеров за несколько десятков итераций.

**12 слайд**

Что касается критерия сходимости, то чаще всего используется критерий суммы квадратов ошибок между центроидом кластера и всеми вошедшими в него записями, т.е.

Где р ϵ Ci — произвольная точка данных, принадлежащая кластеру Сi, mi - центроид данного кластера. Иными словами, алгоритм остановится тогда, когда ошибка Е достигнет достаточно малого значения.

**13 слайд**

Свою популярность алгоритм *k-means* приобрел благодаря:

1. умеренные вычислительные затраты, которые растут линейно с увеличением числа записей исходной выборки. Вычислительной сложности: О(*n*) = *k* \* *п*\* *l*

где *k* - число кластеров, *п* - число записей и *l* - число итераций.

1. результаты его работы не зависят от порядка следования записей в исходной выборке

Одним из основных недостатков, является отсутствие четких критериев выбора числа кластеров. Кроме этого, он очень чувствительным к шумам и аномальным значениям в данных, поскольку они способны значительно повлиять на среднее значение. Чтобы снизить влияние таких факторов, на каждой итерации используют не среднее значение признаков, а их медиану. Данная модификация алгоритма называется k-mediods (k-медиан).

**14 слайд**Ключевым моментом алгоритма является вычисление на каждой итерации расстояния между записями и центрами кластеров, что необходимо для определения, к какому из кластеров принадлежит данная запись. Правило, по которому производится вычисление расстояния в многомерном пространстве признаков, называется метрикой. Наиболее часто используются метрики:

* Евклидово расстояние. Использует для вычисления расстояние между двумя точками в многомерном пространстве, вычисляемое по теореме Пифагора:

Где х = (х1, х2, ..., хm), у = (у1, y2,…, ym) наборы (векторы) значений признаков двух записей. Поскольку множество точек, равноудаленных от некоторого центра при использовании евклидовой метрики будет образовывать сферу (или круг в двумерном случае), то и кластеры, полученные с использованием евклидова расстояния, также будут иметь форму, близкую к сферической.

* Расстояние Манхэттена (метрика 2). Фактически это кратчайшее расстояние между двумя точками, пройденное по линиям, параллельным осям координатной системы (рис. 2).

Преимущество ее заключается в том, что ее использование позволяет снизить влияние аномальных значений на работу алгоритмов. Кластеры, построенные на основе расстояния Манхэттена, стремятся к кубической форме.

**16 слайд**

Пусть имеется набор из 8 точек данных в двумерном пространстве, из которого требуется получить два кластера. Значения точек приведены в таблице 1.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
| (1; 3) | (3; 3) | (4; 3) | (5; 3) | (1; 2) | (4; 2) | (1; 1) | (2; 1) |

**Рассмотрим действия по шагам:**

1. Определим число кластеров, на которое требуется разбить исходное множество *к* = 2.
2. Случайным образом выберем две точки, которые будут начальными центрами кластеров *ml*= (l; l) и *m2*= (2; 1).

**17 слайд**

1. Проход 1. Для каждой точки определим к ней ближайший центр кластера с помощью расстояния Евклида. Таким образом, получим расстояния между центрами кластеров *ml*= (l; 1), *m2*= (2; l) и каждой точкой исходного множества, а также в таблице 2 указано, к какому кластеру принадлежит та или иная точка.

|  |  |  |  |
| --- | --- | --- | --- |
| **Точка** | **Расстояние от m1** | **Расстояние от m2** | **Принадлежит к кластеру** |
| **А** | 2,00 | 2,24 | 1 |
| **В** | 2,83 | 2,24 | 2 |
| **С** | 3,61 | 2,83 | 9 |
| **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 |

**18 слайд**

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 квадратичных ошибок:

**19 слайд**

**4.** Проход 1. Для каждого кластера вычисляется его центроид, и центр кластера перемещается в него.

Центроид для первого кластера вычисляется как:

[(1 +1 +1) / 3, (3 + 2 +1) / 3)] = (1;2)

Центроид для кластера 2 будет равен:

[(3 + 4 + 5 + 4 + 2) / 5, (3 + 3 + 3 + 2 +1) / 5] = (3,6; 2,4)

20 слайд

**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 это еще раз вычисляются евклидовы расстояния между точками и центрами кластеров. Результаты вычислений приведены в таблице 3.

|  |  |  |  |
| --- | --- | --- | --- |
| **Точка** | **Расстояние**  **от m1** | **Расстояние**  **От m2** | **Принадлежит**  **к кластеру** |
| **А** | 1,00 | 2,67 | 1 |
| **В** | 2,24 | 0,75 | 2 |
| **С** | 3,16 | 0,75 | 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 т2 привело к тому, что запись Н оказалась ближе к центру *m1*, 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 А, Е, G, Н, а кластер 2 - В, С, D, F.

**21 слайд**

Новая сумма квадратов ошибок составит:

что показывает уменьшение ошибки относительно начального состояния центров кластеров (которая на первом проходе составляла 36). Это говорит об улучшении качества кластеризации, т.е. более высокую «кучность» объектов относительно центра кластера.

**4.** Проход 2. Для каждого кластера вновь вычисляется его центроид, и центр кластера перемещается в него. Новый центроид для 1-го кластера вычисляется как:

[(1 +1 +1 + 2) / 4, (3 + 2 +1 +1) / 4)] = (1,25; 1,75)

Центроид для кластера 2 будет:

[(3 **+** 4 **+** 5 **+** 4**) /** 4, (3 **+** 3 **+** 2 **+** 4**) /** 4)] **=** (4; 2,75)

**22 слайд**

**3.** Проход 3. Для каждой записи вновь ищется ближайший к ней центр кластера.

**ТАБЛИЦА**

Записей, сменивших кластер на данном проходе алгоритма, не было.

**23 слайд**

Новая сумма квадратов ошибок составит:

|  |
| --- |
|  |

Таким образом, изменение суммы квадратов ошибок является незначительным по сравнению с предыдущим проходом.

**24 слайд**

**4.** Проход 3. Для каждого кластера вновь вычисляется его центроид. Но поскольку на данном проходе ни одна запись не изменила своего членства в кластерах, то положение центроида не поменялось, и алгоритм завершает свою работу.

Рассчитаем вычислительную сложность проделанного алгоритма:

О(п) = к \* п \* l

Где *k* - число кластеров, *n* - число записей, *l* - число проходов:

О(п) = 2 \* 8 \* 3

O (n) = 48