Конспект лекции

А.Н. Шевляков

Машинное обучение

Кластеризация

# Постановка задачи кластеризации

**Кластеризация (clustering).** Дано множество объектов. Их нужно разбить на несколько групп (кластеров), состоящих из похожих друг на друга объектов.

**Для чего нужна кластеризация?**

1. Для вычисления степени сходства объектов. Например, содержание каких веб-страниц близко друг к другу, какие пользователи соцсети близки друг к другу по интересам.
2. Упростить дальнейшую обработку данных, разбить множество М на группы схожих объектов чтобы работать с каждой группой в отдельности.
3. Сократить объём хранимых данных, оставив по одному представителю (эталону) от каждого кластера (задачи сжатия данных).
4. Поиск выбросов.

Алгоритмы, разбивающие данные на заданное число кластеров (то есть число кластеров – это входной параметр алгоритма). Пример: алгоритм k-means.

Алгоритмы, в которых число кластеров не определено заранее, а вычисляется самим алгоритмом. Пример: алгоритм FOREL.

Если алгоритм кластеризации использует метрику на множестве объектов, то значения всех признаков необходимо предварительно нормировать.

# Кластеризация с помощью графов

Необходимо вычислить расстояние между всеми парами объектов.

Представить эти данные в виде графа.

**Описание алгоритма**

На вход алгоритма подается число R.

Удаляем все ребра в графе, метки которых > R. Например, для R=2 имеем картинку.

Кластеры – это **связные компоненты графа {A,B,C,D} и {E,F}.**

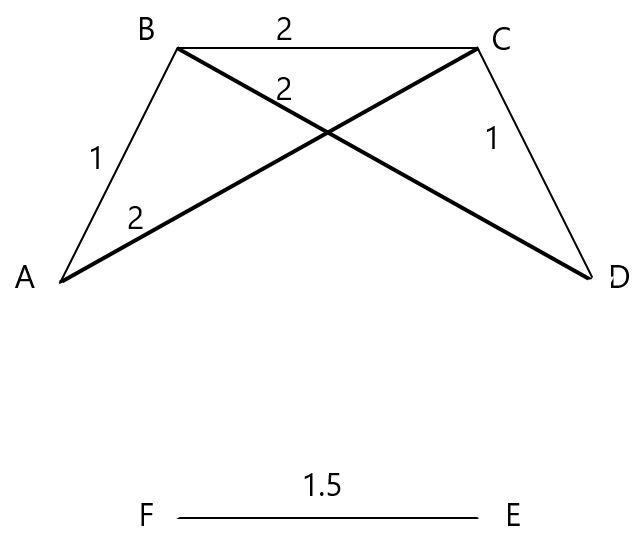


Рис.1

Если на вход алгоритма подать число 1.4, то получим 4 кластера: {A,B},

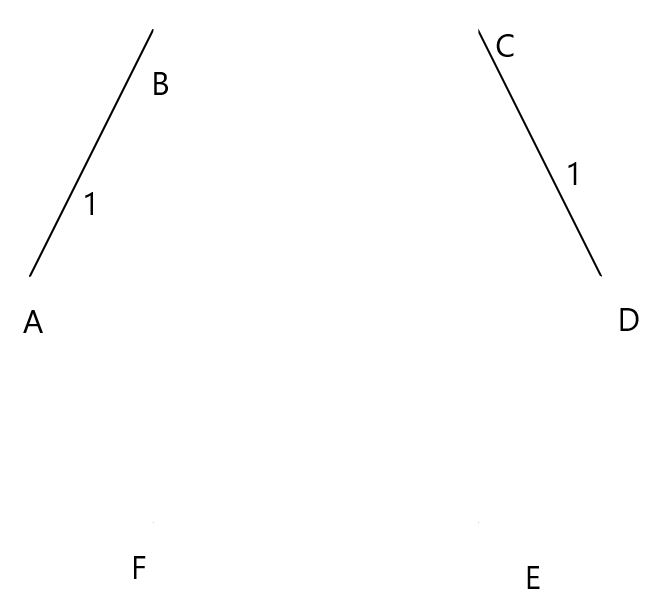
{C,D}, {E}, {F}.

Рис. 2

**Описание второго алгоритма**

На вход алгоритма подается число кластеров k.

1. Строим остовное дерево (это подграф, содержащий все вершины исходного графа и не имеющий циклов) минимальной суммарной длины.

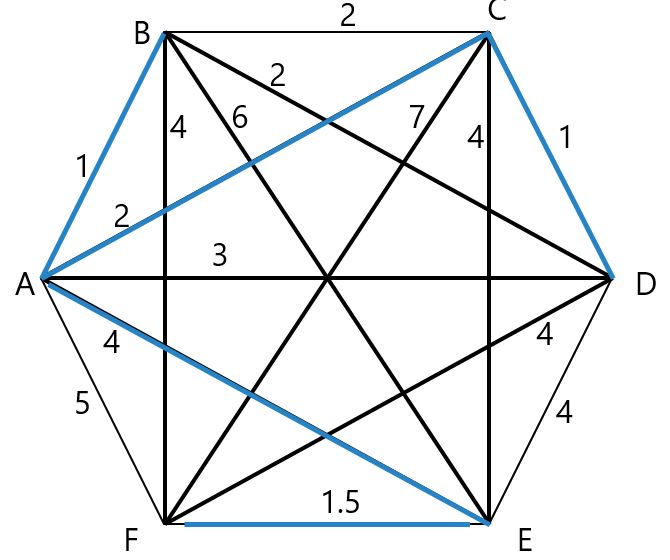


Рис. 3

1. Удаляем из дерева k-1 самых длинных ребер.

Например, для k=3 нужно удалить ребра AE и AC.

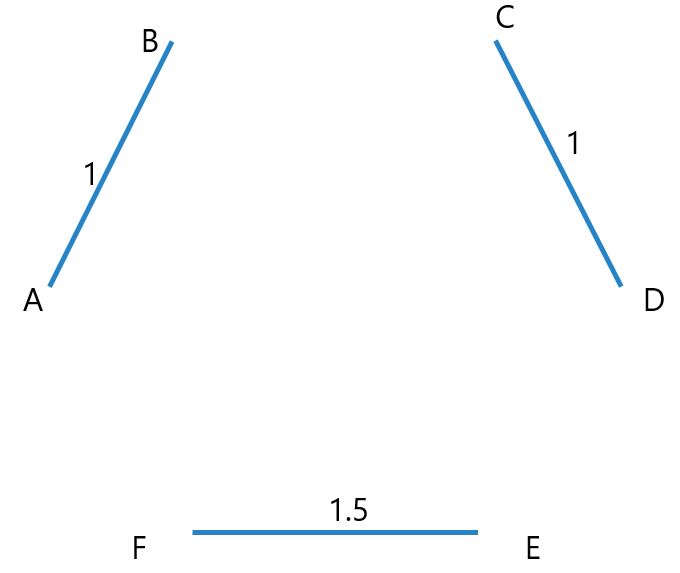
1. В один кластер попадают вершины из связных компонент.

Рис. 4

# Алгоритм FOREL

Главное свойство алгоритма: количество кластеров не определено заранее.

Идея: найти точки сгущения объектов, и эти сгущения объявить кластерами.

**Описание алгоритма forel**

Вход: число R.

Представление данных: объекты представляются точками в пространстве Rm.

**Шаг 1**: В произвольную точку пространства добавляем новый формальный объект F (отсюда и название алгоритма).

**Шаг 2**: Пусть К – все объекты, до которых расстояние от F меньше R.

**Шаг 3**: находим центр тяжести объектов из множества К. Переносим туда объект F. Переходим на шаг 2.

Нужно крутиться в цикле 2-3 до тех пор, пока множество К не стабилизируется.

**Шаг 4**: Когда множество К стабилизируется, оно объявляется новым кластером. Объекты, попавшие в К, из выборки удаляются.

**Шаг 5**: Возвращаемся на шаг 1, если выборка не пуста, иначе конец работы.

Этот алгоритм ищет области сгущения данных. Количество кластеров не определено заранее.

# Алгоритм k-means

Давайте разработаем алгоритм, который разбивает данные на заранее заданное число кластеров.

Главное свойство алгоритма: количество кластеров k определено заранее.

Идея реализации: одновременно происходит поиск всех центров кластеров.

**Описание алгоритма k-means (одна из реализаций)**

Вход: число кластеров k.

Представление данных: объекты представляются точками в пространстве Rm.

**Шаг 1**: Генерируем k случайных точек – центры кластеров.

**Шаг 2**: Объект будет отнесен к тому кластеру, чей центр расположен ближе всех к этому объекту.

**Шаг 3**: Пересчитываются центры кластеров, возврат на Шаг 2.

Цикл 2-3 крутится, пока изменяются центры кластеров.

Недостатки алгоритма k-means:результат зависит от выбора исходных центров кластеров, их оптимальный выбор неизвестен.

Количество кластеров для алгоритма k-means известно заранее.

# Выбор оптимального числа кластеров

Эта проблема актуальна для алгоритмов, в которых «число кластеров» является входным параметром. В частности, это актуально для k-means.

Идея: будем перебирать значения k=1,2… пока «качество кластеризации» не стабилизируется.

Что понимать под «качеством кластеризации»?

Пусть Sk – сумма расстояний от объектов до центров их кластеров (при условии, что объекты разбиты на k кластеров).

Тогда величину |Sk+1 -Sk| можно рассматривать как увеличение качества кластеризации при переходе от k кластеров к (k+1) кластеру.

Таким образом, «качество кластеризации» стабилизируется для такого k, где величина |Sk+1 -Sk| становится небольшой.

Выбор оптимального числа кластеров является нетривиальной проблемой.

Существуют методики, позволяющие найти оптимальное число кластеров.

# Кластеризация по столбцам

Дана таблица.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Вася** | **Петя** | **Маша** | **Даша** |
| Пол | 1 | 1 | 0 | 0 |
| Рост | 172 | 185 | 168 | 201 |
| Вес | 107 | 64 | 61 | 85 |
| Место | 3 | 4 | 2 | 1 |

Таблица 1

Ее можно перевернуть (транспонировать).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Студент** | **Пол** | **Рост** | **Вес** | **Место на олимпиаде** |
| Вася | 1 | 172 | 107 | 3 |
| Петя | 1 | 185 | 64 | 4 |
| Маша | 0 | 168 | 61 | 2 |
| Даша | 0 | 201 | 85 | 1 |

Таблица 2

Мы можем найти близкие (по значению) друг к другу признаки. Можно из каждого кластера оставить по одному признаку – и тем самым уменьшить размер данных.

Это иногда оправданно, так как огромное число признаков часто мешает анализу данных.

На кластеры можно разбивать не только объекты, но и их признаки.

Как правило, это нужно для отбора признаков и уменьшения объема данных.