ФГАОУ ВО «СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информатика»

Методы анализа данных

Практическая работа № 7

Кластеризация

Красноярск, 2024

Цель: знакомство с теоретическими основами кластеризации данных, формирование навыков решения задачи кластеризации с помощью языка программирования Python.

Задачи:

Выполнение практической работы предполагает решение следующий задач:

1. Предварительная обработка исходных данных
2. Построение моделей кластеризации данных
3. Подбор оптимальных параметров моделей
4. Оценка качества построенных моделей

Ссылки на данные

Данные для задачи классификации с kaggle

Общая последовательность действий

1. Загрузить данные. Удалить из набора данных метки классов.

2. Выполнить кластеризацию исходных данных с помощью алгоритма k-means. Количество кластеров выбрать равным числу классов. Оценить, насколько хорошо полученные классы согласуются с истинными метками классов.

3. Используя «правило локтя», подобрать оптимальное количество кластеров. Оценить качество решения задачи по выбранным метрикам оценки качества кластеризации.

4. Полученные результаты визуализировать с помощью одного из методов уменьшения размерности (PCA, t-SNE).

5. Дополнительно реализовать еще один из следующих методов: AffinityPropagation, MeanShift, SpectralClustering, AgglomerativeClustering, DBSCAN, Birch. Настроить гиперпараметры данного алгоритма. Сравнить результаты.

Примеры метрик качества кластеризации

1. Adjusted Rand index

Метрика применяется в том случае, если известны истинные метки классов. Для вычисления метрики используется функция *adjusted_rand_score*. Метрика возвращает результат в диапазоне $[-1; 1]$. Значение близкое к 1 говорит об очень хорошем качестве кластеризации. Значение близкое к 0 соответствует случайным разбиениям. Отрицательные значения говорят о плохом качестве кластеризации.

2. Adjusted Mutual Information

Для вычисления метрики используется функция *adjusted_mutual_info_score*.

Значение близкое к 1 говорит об очень хорошем качестве кластеризации. Значение близкое к 0 соответствует случайным разбиениям.

3. Homogeneity, completeness, V-measure

Для вычисления метрик используется функция *homogeneity_completeness_v_measure*.

- *Homogeneity* – каждый кластер содержит только представителей единственного класса (под классом понимается истинное значение метки кластера). Значение в диапазоне [0; 1], 1 говорит об очень хорошем качестве кластеризации.

- *Completeness* – все элементы одного класса помещены в один и тот же кластер. Значение в диапазоне [0; 1], 1 говорит об очень хорошем качестве кластеризации.

- *V-measure* – среднее гармоническое от Homogeneity и Completeness.

4. Коэффициент силуэта

Для вычисления метрики используется функция *silhouette_score*

Данный метод не требует знания истинных значений меток кластеров.

Пусть:

- a - среднее расстояние между текущей точкой и другими точками этого же кластера.
- b - среднее расстояние между текущей точкой и другими точками следующего ближайшего кластера.

Тогда коэффициент силуэта для точки (объекта) определяется как:

$$s = \frac{b - a}{max(a, b)}$$

Силуэтом выборки называется средняя величина силуэта объектов данной выборки. Таким образом, силуэт показывает, насколько среднее расстояние до объектов своего кластера отличается от среднего расстояния до объектов других кластеров. Данная величина лежит в диапазоне [-1;1]. Значения, близкие к -1, соответствуют плохим (разрозненным) кластеризациям, значения, близкие к нулю, говорят о том, что кластеры пересекаются и накладываются друг на друга, значения, близкие к 1, соответствуют "плотным" четко выделенным кластерам. Таким образом, чем больше силуэт,

тем более четко выделены кластеры, и они представляют собой компактные, плотно сгруппированные облака точек.

С помощью силуэта можно выбирать оптимальное число кластеров (если оно заранее неизвестно) – выбирается число кластеров, максимизирующее значение силуэта. В отличие от предыдущих метрик, силуэт зависит от формы кластеров, и достигает больших значений на более выпуклых кластерах, получаемых с помощью алгоритмов, основанных на восстановлении плотности распределения.

Требования к выполнению практической работы:

1. Написание программного кода и формирование результатов согласно заданию и установленному варианту.

2. Составление отчета, содержащего описание решаемых задач методов решения и полученных результатов.

3. Воспроизводимость полученного результата.

Программный код и отчет должны быть выполнены в среде Jupyter notebook.