Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Самарский национальный исследовательский университет имени академика С.П. Королева»

Лабораторная работа №1

АНАЛИЗ СОДЕРЖИМОГО БОЛЬШОГО КОЛИЧЕСТВА ДОКУМЕНТОВ МЕТОДАМИ КЛАСТЕРИЗАЦИИ

Методические указания
к лабораторным работам по курсу «Большие данные»

Самара 2020

Составитель В.В. Жидченко

УДК 681.3.016

Анализ содержимого большого количества документов методами кластеризации:
Метод. указания к лабораторным занятиям / Самар. ун-т; Сост. В.В. Жидченко.
Самара, 2020. 10 с.

В методических указаниях рассматриваются методы кластеризации, широко применяющиеся в анализе “больших данных”.
Методические указания подготовлены на кафедре программных систем.

ВВЕДЕНИЕ

Методы автоматического анализа текста предоставляют мощные средства для извлечения знаний из большого количества документов, относящихся к определенной теме. Например, они могут быть использованы для получения обзора содержимого некоторой темы и отслеживания тенденций развития темы с течением времени.

В настоящей лабораторной работе выполняется анализ содержимого большого количества научных статей по теме “Big Data”. Цель анализа состоит в том, чтобы определить основные темы статей. Для решения этой задачи используются методы кластерного анализа.

Кластерный анализ (Кластеризация)

Кластеризация означает выделение групп схожих объектов в заданном наборе объектов. Примером сходства является евклидово расстояние между точками. Точки в группе могут быть определены как схожие, если расстояние между любыми двумя из них меньше, чем расстояние между любой из этих двух точек и некоторой другой точкой, не принадлежащей группе.

Существует множество алгоритмов кластеризации (VanderPlas, 2016). Одним из широко применяемых алгоритмов является K-means clustering. В этом алгоритме количество кластеров nc, на которые разделяются исходные данные, должно быть задано до начала работы алгоритма. Алгоритм K- means пытается разделить данные на nc групп. Каждая группа описывается средним значением μi, которое обычно называют «центром кластера», «центром тяжести кластера», «центроидом кластера». В общем случае центроид не является точкой из набора данных. Это среднее арифметическое всех точек, принадлежащих кластеру. Каждая точка набора данных назначается кластеру, центр тяжести которого является ближайшим к этой точке. Алгоритм включает в себя три шага. На первом шаге для набора данных определяются nc центроидов с помощью некоторого алгоритма. Простейший подход - выбрать их случайным образом, например, в виде nc точек из набора данных. Более сложные решения пытаются распределить центроиды так, чтобы они были удалены друг от друга. Два других шага выполняются итеративно. На каждой итерации второй шаг

назначает каждую точку данных ближайшему к ней центроиду. На третьем шаге для каждой группы точек рассчитываются новые центроиды как среднее значение всех точек, присвоенных каждому центроиду на предыдущем шаге. Расстояние между новым и старым центроидом в каждой группе вычисляется, и алгоритм повторяет два последних шага, пока это расстояние не станет меньше предварительно заданного значения. Другими словами, алгоритм повторяется до тех пор, пока центроиды не перестанут существенно перемещаться. Алгоритм имеет некоторые недостатки [3]:

1) Количество кластеров должно быть предварительно задано. В зависимости от целей анализа это число может быть скорректировано, если получаются кластеры, содержащие слишком разрозненные данные. Автоматическое решение этой задачи называется анализом силуэта. Он используется для определения степени разделенности полученных кластеров друг от друга. Существуют альтернативные методы кластеризации, которые могут автоматически рассчитать подходящее количество кластеров (DBSCAN, среднее смещение, распространение сродства) (VanderPlas, 2016);

2) Алгоритм K-means существенно зависит от начального расположения центроидов. Из-за этого он может сходиться к конфигурации, которая не является оптимальной в глобальном масштабе. Поэтому обычно алгоритм запускается несколько раз с различными начальными положениями центроидов;

3) Границы между кластерами K-means всегда линейны. Алгоритм может быть неэффективным, если кластеры имеют сложную геометрию.

Для применения методов кластерного анализа к текстовым данным необходимо векторизовать текст, то есть преобразовать текстовые данные в векторы в евклидовом пространстве.

Векторизация текста

Самый простой способ векторизации текста — использование подхода «Bag of Words». Он рассматривает текст как неупорядоченный набор слов, не учитывая информацию об относительном положении слов в тексте. В этом подходе текст

характеризуется относительной частотой появления каждого слова (то есть отношением количества употреблений каждого слова к общему количеству слов в документе), но не порядком, в котором слова появляются в тексте. При таком подходе набор текстовых документов (обычно называемый «корпусом») может быть представлен матрицей, содержащей по одной строке на каждый документ и по одному столбцу на каждое слово в корпусе:

$$A = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1N} \\ f_{21} & f_{22} & \cdots & f_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M1} & f_{M2} & \cdots & f_{\mathrm{MN}} \end{bmatrix}$$

Подразумевается, что корпус состоит из M документов, и все документы корпуса вместе содержат N различных слов. Каждый элемент f_{ij} матрицы A равен частоте появления слова j в документе i. Если поставить в соответствие каждому слову ось в N-мерном евклидовом пространстве и рассматривать частоту появления этого слова как координату вдоль указанной оси, то каждый текстовый документ может быть представлен вектором в N-мерном пространстве, компоненты которого соответствуют частотам употребления слов в этом документе. В матрице A каждая строка описывает вектор, соответствующий документу с номером i. Элементы каждой строки равны координатам конца соответствующего вектора. Два документа считаются равными, если все их компоненты одинаковы. Это означает, что в рассматриваемых документах уотребляются одинаковые слова с одинаковой частотой.

Чтобы увеличить скорость обработки корпуса документов, можно уменьшить количество отличных друг от друга слов в документах корпуса и таким образом уменьшить размерность евклидова пространства, соответствующего корпусу. Часто применяемый для этого подход использует словарь «стоп-слов», т.е. часто используемых слов языка (например, «the», «a», «is» в английском языке). Такие слова встречаются во многих документах, но не несут значимой информации о фактическом содержании документа. Эти слова

исключаются из корпуса до обработки. Другой способ называется "стемминг" и представляет собой процесс приведения каждого слова к его неизменной форме, называемой "основа" (stem), путем удаления лишних морфем (аффиксов). Например, слова "compute", "computer", "computers", "computing" сводятся к одной основе - «comput». Это позволяет объединять разные формы одного и того же слова в общую сущность.

Дальнейшее улучшение процесса векторизации текста может быть достигнуто с помощью метода tf-idf (term frequency–inverse document frequency). Этот метод учитывает тот факт, что некоторые слова используются в языке чаще, чем другие. Для каждого слова вводится вес, чтобы сделать редкие слова более ценными. Если слово встречается в каждом документе корпуса, это не позволяет отличить документы друг от друга и затрудняет кластеризацию. Вместо использования частоты слова метод tf-idf умножает частоту на компонент idf, который вычисляется следующим образом:

$$\mathrm{idf}(j) = \log\left(\frac{1+M}{1+\mathrm{df}(j)}+1\right)$$

где j - номер столбца слова в матрице A, M - общее количество документов в корпусе, df(j) - количество документов, содержащих слово, соответствующее столбцу j.

Общий вес слова рассчитывается как частота слова, умноженная на idf:

$$\mathrm{tfidf}(i, j) = f_{\mathrm{ij}} \cdot \mathrm{idf}(j)$$

Полученные векторы с компонентами tfidf можно нормировать, используя евклидову норму:

$$V_{\mathrm{norm}} = \frac{V}{\|V\|} = \frac{V}{\sqrt{V_1^2+V_2^2+\ldots+V_N^2}}$$

Такая нормализация позволяет сбалансировать документы разного размера. Результирующая нормализованная матрица для совокупности документов может быть представлена следующим образом:

$$A = \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1N} \\ F_{21} & F_{22} & \cdots & F_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ F_{M1} & F_{M2} & \cdots & F_{MN} \end{bmatrix}$$

где $F_{ij}=tfidf_{norm}(i,j)$ [1].

Задание на лабораторную работу

В настоящей лабораторной работе необходимо кластеризовать корпус документов, состоящий из англоязычных статей, содержащих словосочетание «big data». Это позволит разбить множество статей на отдельные темы, использующие указанное словосочетание. В результате можно получить представление о различных направлениях исследований в области Big Data.

Анализируемый корпус документов состоит из 2141 статьи, которые были опубликованы в период с 2016 по 2020 г. и получены в результате поискового запроса «big data» на сайте модерируемого архива научных статей с открытым доступом arXiv.org (ARXIV, 2018).

Задание:

1) Скачать архив с содержимым статей. Архив содержит статьи в формате TXT, полученные автоматическим преобразованием из формата PDF для упрощения анализа. Имена файлов имеют формат YYMM.N.txt, где YY — год публикации статьи, MM — месяц публикации, N — порядковый номер статьи в пределах соответствующего месяца.

2) Выполнить кластеризацию корпуса документов, используя библиотеки scikit-learn и nltk языка Python. Для этого:

2.1) Создать dataset из файлов скачанного архива в формате TXT;

2.2) Используя библиотеку nltk, выполнить стемминг документов, образующих dataset.

2.3) Выполнить кластеризацию корпуса документов с использованием словаря стоп-слов английского языка и алгоритма

K-means. При необходимости дополнить словарь стоп-слов самыми часто используемыми словами из области Big Data. Сначала необходимо выполнить кластеризацию несколько раз, задав число кластеров равным 12, и изменяя различные параметры алгоритма K-means. Выбрать те параметры, которые обеспечили наилучшее разделение кластеров, используя один из критериев качества разделения, например, анализ силуэта. Затем необходимо отдельно выполнить кластеризацию того набора документов, который сформировал самый объемный кластер, т. е. тот из 12-ти найденных кластеров, который содержит максимальное количество документов. При этом количество искомых кластеров задать равным 6.

2.4) Определить тему статей, принадлежащих найденным кластерам. Для этого воспользоваться списком наиболее часто встречающихся слов в каждом кластере, а также просмотреть названия и бегло просмотреть содержание нескольких статей из каждого кластера, пользуясь файлами со статьями. При необходимости можно скачать любую статью с сайта arxiv.org в формате PDF.

Для выполнения данного пункта рекомендуется использовать программы analyse_input и doclustering.py, содержащиеся в архиве с документами.

3) Составить отчет по результатам работы.

Содержание отчета:

3.1) Постановка задачи, описание исходных данных и параметров задачи.

3.2) **Словесное описание** программы.

3.3) Описание (желательно графическое) зависимости качества разделения кластеров от используемых параметров алгоритмов.

3.4) Описание (желательно графическое) найденных кластеров (тем статей) с указанием наименования темы, количества документов в каждом кластере, списка самых часто употребляемых слов в документах кластера.

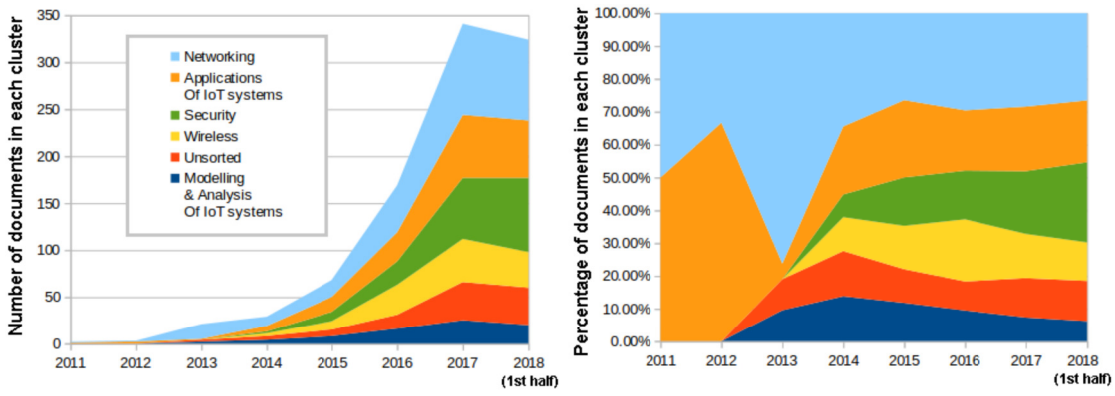
3.5) Графическое представление доли документов, образующих каждый кластер, по отношению к общему числу документов в корпусе (круговая диаграмма).

3.6) Анализ эволюции темы, соответствующей каждому кластеру, с течением времени. Следует использовать описанную выше систему наименования имен файлов для определения года публикации. Эволюция темы — это изменение количества документов, относящихся к кластеру, от года к году. Необходимо построить графики:

3.6.1) зависимости количества документов в каждом кластере от года публикации;

3.6.2) зависимости доли документов, образующих каждый кластер, по отношению к общему числу документов в корпусе, от года публикации.

Пример графиков приведен на рисунке:



Рекомендуемая литература:

1) Описание алгоритма K-Means на сайте библиотеки scikit-learn:

https://scikit-learn.org/stable/modules/clustering.html#k-means

2) Описание параметров функции KMeans на сайте библиотеки scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans
3) Описание способов векторизации текста на сайте библиотеки scikit-learn: https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction
4) VanderPlas J. (2016). Python Data Science Handbook. Essential Tools for Working with Data. O'Reilly Media, Inc.

Приложение
Установка Python и необходимых библиотек в Ubuntu

Как правило, интерпретатор Python устанавливается вместе с другими пакетами программ по умолчанию при установке ОС Ubuntu. Проверить наличие Python в Вашей системе можно, запустив программу эмуляции терминала (обычно она так и называется: «Терминал») и набрав в командной строке:

```
> python3
```

Должен открыться интерфейс интерпретатора Python. Если Вы видите сообщение об ошибке, нужно установить Python. Для этого перейдите в режим суперпользователя командой:

```
> sudo su
```

Установка Python и библиотек scikit-learn, nltk, matplotlib:

```
# apt-get update
# apt-get install python3
# apt-get install python3-pip
# pip3 install -U scikit-learn
# pip3 install -U nltk
# pip3 install -U matplotlib
```

После установки Вы можете запускать программы, содержащиеся в архиве с документами к настоящей лабораторной работе.