Тема «Кластеризация социальных сетей» в курсе «Машинное обучение и анализ данных»

Балашова Т.А.[1], Огнева М.В.[2]
[1]bala5howatatwork@yandex.ru, [2]ognevamv@mail.ru
Саратовский государственный университет имени Н.Г. Чернышевского

Аннотация. В данной статье рассмотрена методика преподавания темы кластеризации социальных сетей для студентов.

Ключевые слова: машинное обучение, кластеризация, социальная сеть, граф.

Одной из важных задач анализа больших данных является задача кластеризации, которая заключается в разбиении множества объектов на сообщества [1]. Данная задача сейчас очень актуальна, поскольку находит широкое применение в различных сферах человеческой жизни (медицина, спорт, музыка, социальная сфера и т.д.) и возникает тогда, когда, например,

нужно выделить дружественные сообщества по интересам, разбить страны мира на группы схожих по экономическому положению государств или по результатам социологических опросов выявить группы общественных проблем, вызывающих схожую реакцию у общества. Результаты данных разбиений могут помочь совершать прогнозы на будущее.

В связи с тем, что одну из лидирующих позиций по производству больших данных занимают в настоящее время социальные сети, отдельно выделяется задача кластеризации социальных сетей.

Структуру социальной сети могут образовывать различные виды отношений – дружба, взаимодействие, обмен контентом и т.д. Умение правильно разделять и соотносить пользователей в этой обширной структуре поможет давать верные рекомендации, определять грамотный контент, а также фильтровать получаемую информацию.

На факультете компьютерных наук и информационных технологий вопросы, посвященные данной тематике, рассматриваются в ходе изучения темы кластеризации для студентов направления «Математическое обеспечение и администрирование информационных систем» в рамках курса дисциплины «Машинное обучение и анализ данных», а также в ходе выполнения курсовых и выпускных квалификационных работ.

При проведении занятий по данной теме рассматриваются:

– актуальность решения задачи кластеризации социальных сетей, примеры;
– общая и формальная постановка задачи кластеризации социальных сетей;
– оценка качества решения задачи кластеризации социальных сетей;
– обзор алгоритмов решения задачи кластеризации социальных сетей, исходя из их особенностей, временной сложности, простоты понимания и т.д.
– подробный разбор одного-двух рассматриваемых алгоритмов;
– задание для самостоятельной работы.

Далее рассмотрим основные вопросы более подробно.

1. Социальная сеть. Постановка задачи кластеризации сетей

Структуру социальной сети очень удобно представлять в виде графа.

Под графом понимается упорядоченная пара $G = (V, E)$, где V – непустое множество объектов, называемых вершинами, $E \subset V^2$ – множество ребер.

Графы могут быть ориентированными и неориентированными.

Ориентированный граф представляет собой упорядоченную пару

$$G = (V, E): \forall (u, v) \in E \; \exists (v, u) \in E$$

Неориентированный граф представляет собой упорядоченную пару

$$G = (V, E): \forall (u, v) \in E \; !\exists (v, u) \in E$$

В дальнейшем будут рассматриваться неориентированные графы.

Формальная постановка задачи кластеризации графов выглядит следующим образом.

Дан неориентированный граф $G = (V, E)$, где V – множество вершин, E – множество ребер. Необходимо получить покрытие множества вершин, называемое кластеризацией, которое выглядит следующим образом:

$V = \bigcup_{C^i \in C} C^i$, где $C^i \neq \emptyset$ – покрытие множества вершин кластера с номером i, C – покрытие множества всех вершин.

2. Оценка качества

Оценка качества является очень важным этапом в анализе результатов кластеризации социальных сетей.

Принято выделять две группы методов оценки качества кластеризации:

А) Внешние методы – методы, основанные на сравнении результата с априори известным разделением на классы, так называемыми, ground-truth сообществами (индекс Rand, индекс Фоулкса-Мэллова).

Б) Внутренние методы – методы, определяющие качество кластеризации только по признаковым описаниям объектов, без итогового разбиения (индекс компактности, индекс разделимости, модулярность).

Для примера рассмотрим подробнее понятие «модулярность».

Модулярность оценивает плотность разбиения сети. Чем выше значение модулярности, тем более плотными являются связи вершин в одном сообществе и менее плотными между вершинами из разных сообществ.

Понятие «модулярность» впервые было введено Гирваном и Ньюманом и может определяться по следующей формуле:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{d_i d_j}{2m}\right) \delta(C_i, C_j), \text{ где}$$

A_{ij} – элемент -той строки и j-того столбца матрицы смежности графа,

C_i – номер сообщества, к которому относится вершина,

m – общее количество ребер в графе,

$\delta(C_i, C_j)$ – дельта-функция, равная единице, если $C_i = C_j$, в противном случае – нулю [2].

3. Обзор алгоритмов решения задачи кластеризации социальных сетей

Алгоритмы кластеризации социальных сетей можно классифицировать по механизмам, лежащим в основе их работы. Так, например,

– распространение близости (Labelpropagation);
– жадная оптимизация модулярности (Fastgreedy, Lovain);
– случайные блуждания (Infomap, Walktrap).

Алгоритм Infomap использует механизм случайных блужданий: задача поиска сообществ в графе сводится к минимизации длины кода пути, который проделает «случайный блуждатель».

Алгоритм Labelpropagation основывается на том принципе, что вершина относится к тому сообществу, что и большинство ее соседей. Изначально каждая вершина представляет собой отдельное сообщество. Затем для каждой вершины номер сообщества переопределяется в соответствии с номером того сообщества, к которому относятся большинство соседей этой вершины. Алгоритм завершается, когда в графе перестают происходить изменения.

Алгоритм Fastgreedy реализует механизм жадной оптимизации модулярности: сообщества объединяются таким образом, что это значение

достигает максимума.

Алгоритм Walktrap, также как и Infomap, обращается к случайным блужданиям. Основная идея алгоритма заключается в том, что короткие случайные блуждания не приводят к выходу из сообщества.

Алгоритм Lovain был разработан группой авторов из Лувенского католического университета (г.Лувен, Бельгия) в 2008 году. Он основан на жадной оптимизации модулярности и использует при этом механизм эвристики локального перемещения (LMH). Идея данного механизма заключается в переборе вершин в случайном порядке и перемещении этих вершин из одного сообщества в другое таким образом, чтобы модулярность увеличилась максимально.

4. Теоретический разбор алгоритмов на примере Lovain

Для более детального разбора был выбран алгоритм Lovain. Идея данного алгоритма состоит в жадной оптимизации модулярности.

Шаги алгоритма:

1. Инициализация – каждой вершине выделяется отдельное сообщество.
2. Оптимизация модулярности – для каждой вершины происходит попытка найти сообщество, перемещение в которое даст максимальное положительное изменение модулярности. Попытки продолжаются до тех пор, пока дальнейшее увеличение модулярности не станет невозможным.

Алгоритм продолжается до тех пор, пока граф не перестанет меняться [3].

Для примера рассмотрим граф и пошагово проиллюстрируем результат работы алгоритма Lovain. Пример исходного представлен на рисунке 1.

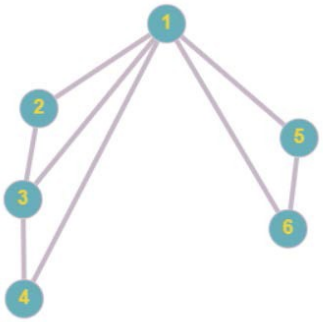

Рис. 1. Пример исходного графа

Для решения данной задачи воспользуемся следующим алгоритмом:

Шаг 1. Выделение констант

А) Матрица смежности исходного графа:

$$A = \begin{pmatrix} A_{11} & \cdots & A_{16} \\ \vdots & \ddots & \vdots \\ A_{61} & \cdots & A_{66} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Б) Вектор-столбец степеней вершин

$$D = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \\ 3 \\ 2 \\ 2 \\ 2 \end{pmatrix}$$

В) Матрица множителей (вспомогательная – для упрощения расчетов модулярностей). Определим ее следующим образом:

$$\forall i,j \in V: j > i \; M_{ij} = A_{ij} - \frac{d_i d_j}{2m}$$

Пример расчета множителя для одной пары:

$$M_{12} = A_{12} - \frac{d_1 d_2}{2*8} = 1 - \frac{5*2}{16} = 1 - \frac{10}{16} = \frac{6}{16} = 0{,}375$$

Множители для остальных пар рассчитываются аналогичным образом. Таким образом, получим следующую матрицу:

$$M = \begin{pmatrix} - & 0{,}375 & 0{,}0625 & 0{,}375 & 0{,}375 & 0{,}375 \\ - & - & 0{,}625 & -0{,}25 & -0{,}25 & -0{,}25 \\ - & - & - & 0{,}625 & -0{,}375 & -0{,}375 \\ - & - & - & - & -0{,}25 & -0{,}25 \\ - & - & - & - & - & 0{,}75 \\ - & - & - & - & - & - \end{pmatrix}$$

Шаг 2. Инициализация графа

Помещаем каждую вершину в уникальное сообщество. Выписываем пары вершин, находящихся в одном сообществе. В данном случае:

$$Pairs = \emptyset \Rightarrow \forall i,j \in V: i \neq j \; \delta(C_i, C_j) = 0 \Rightarrow Q_{нач} = 0$$

Шаг 3. Оптимизация модулярности

Обозначим множество существующих сообществ за Communities. На данный момент:

$$Communities = \{1,2,3,4,5,6\}$$

Выбираем любую стартовую вершину. Допустим, $v_{start} = 2$. Ищем для нее подходящее сообщество. Сообщество, в котором раньше находилась стартовая вершина, при этом исчезает. Таким образом:

$$Communities = \{1,3,4,5,6\}$$

Далее рассчитываем модулярности и выписываем пары вершин, оказавшихся в одном сообществе, при переходе этой вершины в сообщество (находящееся в Communities). Делаем это до тех пор, пока модулярность не прекратит увеличиваться.

Аналогичным образом, перебирая следующие вершины, определяем для них оптимальное сообщество. При этом если все вершины образовали одно сообщество с максимальной модулярностью или при выборе очередной вершины не удалось найти для нее сообщества оптимальнее, чем то, в котором

она изначально находилась, то это означает, что в графе перестали происходить изменения и алгоритм завершает работу.

Пример одного из вариантов искомого разбиения представлен на рисунке 2.

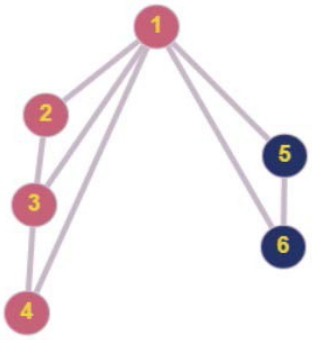

Рис. 2. Вариант искомого разбиения графа

В качестве домашнего задания можно предложить студентам сделать аналогичный разбор самостоятельно.

Искомое разбиение при этом может получиться разным, так как очень сильно зависит от последовательности рассматриваемых вершин, сообществ и т.д. Как следствие, работы получаются очень разными, что помогает выявить неизвестные ранее варианты искомых разбиений, а также способы оптимизации рассматриваемого алгоритма в рамках научно-исследовательской деятельности.

5. Практический анализ

В рамках курсовых работ рассматриваются готовые датасеты большого размера, а также самостоятельно сгенерированные датасеты, выполняется замер времени выполнения различных алгоритмов, а также оценка и анализ полученных кластеров. Рассмотрим более подробно пример такой работы.

Был проведен эксперимент на самостоятельно построенном графе социальной сети ВКонтакте. Вершинами графа являлись пользователи социальной сети ВКонтакте, ребро между двумя вершинами строилось, если соответствующие пользователи являются друзьями. Полученный граф был вручную разбит на ground-truth сообщества – одногруппники, одноклассники, спортсмены и т.д.

Рассмотрим общие результаты.

По времени выполнения самыми быстрыми оказались алгоритмы Labelprogation и Lovain.

Значение модулярности получилось небольшим у всех алгоритмов, поскольку анализируемый участок сети является очень узким, ибо рассматриваются друзья всего лишь одного пользователя. Однако результаты расчетов внешних оценок качества показывают, что достаточно большое количество объектов ground-truth сообществ в результате кластеризации алгоритмами также оказываются в одном сообществе.

Исходя из получившихся значений нормализованной взаимной информации, можно сказать о том, что в целом разбиения похожи друг на друга и не совсем похожи на ground-truth сообщества. Поскольку данные не

обезличены, можно посмотреть, как выглядят разбиения, полученные с помощью алгоритмов.

Рассмотрим три самых больших ground-truth сообщества – родственников, спортсменов и творческую молодежь.

Среди родственников алгоритм Labelpropagation выделяет отдельно родственников по линии матери и по линии отца, алгоритм Walktrap определяет в отдельные сообщества двоюродных братьев и сестер.

Членов творческой молодежи алгоритмы в основном относят к одному сообществу, причем оно формируется в соответствии с возрастным порогом (около 30 лет).

Среди спортсменов алгоритмы выделяют два крупных сообщества. Представители одного из них преимущественно городские, второго – деревенские жители.

Теперь проанализируем сообщества, определяемые алгоритмами. Для этого рассмотрим три крупнейших из них.

Первое сообщество можно назвать спортивным: в разбиениях всех алгоритмов их количество доминирует. В разбиение алгоритма Labelpropagation попало большее количество родственников, т.к. сам алгоритм является неустойчивым и даже в некоторых случаях может объединить все сообщества в одно.

Во втором сообществе алгоритм Labelpropagation объединяет всех одногруппников и преподавателей в одно сообщество. Это справедливо, поскольку данное сообщество можно было назвать университетом. Алгоритмы Infomap, Fastgreedy и Walktrap исключают из сообщества некоторых преподавателей, работающих в компаниях. Алгоритм Fastgreedy также присоединяет к данному сообществу трех членов творческой молодежи и одного родственника, которые учились в университете, чьи преподаватели и студенты объединены в одно сообщество.

Опираясь на результаты разбиений для третьего сообщества, можно сказать, что алгоритм Lovain в целом правильно разделяет преподавателей и студентов, что является очень хорошим результатом.

По времени выполнения самыми быстрыми являются Lovain и Labelpropagation – на небольших данных работают соизмеримо, однако с увеличением объема данных Labelpropagation начинает проигрывать. Если же говорить о разбиениях, то несмотря на то, что каждый делит по-своему, получаются вполне осмысленные группы.

Список литературы

[1] Кластеризация [Электронный ресурс] URL: http://www.machinelearning.ru/wiki/index.php?title=Кластеризация (дата обращения: 10.10.2021).

[2] *Ионкин М.С.* Программная реализация, анализ эффективности и оценка качества алгоритмов кластеризации графовых моделей социальных сетей // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2017. – Т. 17, №. 4. –С. 441-451.

[3] Lovain method for community detection [Электронный ресурс] URL: https://perso.uclouvain.be/vincent.blondel/research/louvain.html (дата обращения: 26.10.2021).