ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ AFFINITY PROPAGATION, DBSCAN К РЕШЕНИЮ ЗАДАЧИ ПОИСКА ПОЛЬЗОВАТЕЛЕЙ-ЭКСПЕРТОВ В СОЦИАЛЬНЫХ СЕТЯХ

Д.А. Виноградова
Томский политехнический университет
dak38@tpu.ru

Введение

Высокий уровень вовлеченности современного человека в социальные сети приводит к колоссальному скоплению различных данных в социальных сетях, анализ которых позволяет решать различные задачи, в том числе социологические и маркетинговые.

С целью повышения эффективности анализа таких данных необходимо определять множество пользователей, рассматриваемых в качестве лидеров общественного мнения в заданной предметной области. Подобная задача описана в [1] как задача KPP POS (Key Players Problem Positive).

Во многих случаях необходимо опираться на группу экспертов без учета значимости каждого из них, для выделения которой применяется кластерный анализ.

Данная работа содержит исследование применения алгоритмов кластеризации данных DBSCAN, Affinity propagation (в дальнейшем: AP) и Fast Affinity Propagation (в дальнейшем: FAP) к задаче идентификации группы наиболее значимых пользователей-экспертов в социальных сетях в заданной предметной области (в дальнейшем: рассматриваемая задача).

Описание алгоритмов кластеризации

Выбор данных алгоритмов объясняется сравнительно низкой ошибкой кластеризации, по сравнению с другими методами, например, с известным методом кластеризации k-means [Jain et al., 1999], способностью находить схожие объекты, даже если эти объекты не удовлетворяют неравенству треугольника. Также методы самостоятельно определяет количество кластеров. Алгоритм FAP рассмотрен, как алгоритм, представляющий более быструю версию алгоритма AP.

1. DBSCAN

DBSCAN – это алгоритм, основанный на плотности распределения точек данных [2].

Входные данные алгоритма: множество объектов X, для которых задана метрическая функция расстояния ρ, а также ε - максимальное расстояние между соседними объектами, и minPts - минимальное количество соседних объектов, необходимых для образования кластера.

Пошаговая инструкция:

1. Выбираем необработанный объект p.
2. Отмечаем объект p как обработанный.
3. Находим соседние объекты в ε-окрестности объекта p.
4. Сравниваем количество соседних объектов с MinPts, определяя, достаточно ли у p соседей, чтобы образовать кластер.
   Если достаточно, то создаём новый кластер и запускаем поиск в ширину из данного объекта по другим не посещённым объектам, находя все объекты кластера.
   Если недостаточно, то отмечаем p шумом.
5. Если присутствуют необработанные объекты, то возвращаемся к шагу 1.

2. AP

AP выделяет среди объектов «образцы» - exemplar - и формирует кластеры вокруг них.

Входные данные алгоритма: метрическая функция схожести s, количественно определяющая сходство между двумя точками.

Пошаговая инструкция

1. Матрицы r (матрица ответственности) и a (матрица доступности) инициализируются ко всем нулям.
2. Выполнять этот шаг заданное T количество раз:
   2.1. Обновить матрицу r следующим образом:

$$r[i,j] = (1 - \lambda)\rho[i,j] + \lambda r[i,j], \quad (1)$$

где ρ [i, j] - распространяемая ответственность, которая вычисляется по выражению:

$$\rho[i,j] = \begin{cases} s[i,j] - max_{k \neq j}\{a[i,k] + s[i,k]\} & (i \neq j) \\ s[i,j] - max_{k \neq j}\{s[i,k]\} & (i = j) \end{cases} \quad (2)$$

   2.2. Обновить матрицу a следующим образом:

$$a[i,j] = (1 - \lambda)\gamma[i,j] + \lambda a[i,j] \quad (3)$$

где $\gamma[i,j]$ - распространяя доступность, которая вычисляется по следующему выражению:

$$\gamma[i,j] = \begin{cases} \min\left(0, r[j,j] + \sum_{k \neq i,j} \max(0, r[k,j])\right) & (i \neq j) \\ \sum_{k \neq i,j} \max(0, r[k,j]) & (i = j) \end{cases} \quad (4)$$

λ - коэффициент затухания, введенный во избежание численных колебаний

3. Вычислить образцы. Образцами считаются точки, удовлетворяющие условие:

$$r(i,i) + a(i,i) > 0 \quad (5)$$

4. Вычислить образец для каждой точки: найти образец, с котором точка максимально похожа.

Для описания алгоритма использовались в основном источники [3] и [4].

3. FAP

Рассмотрена одна из версий FAP, которая представлена в источнике [4].

Входные данные алгоритма такие же, как у AP.

Пошаговая инструкция

1. Для каждой пары точек данных вычислить по следующим определениям верхние/нижние ограничивающие оценки $\underline{a}$, $\overline{r}$ и $\overline{a}$:

$$\underline{a}[i,j] = \begin{cases} min(0, r[j,j]) & (i \neq j) \\ 0 & (i = j) \end{cases} \quad (6)$$

$$\overline{r}[i,j] = \begin{cases} s[i,j] - max_{k \neq j}\left(\underline{a}[i,k] + s[i,k]\right) & (i \neq j) \\ s[i,j] - max_{k \neq j}(s[i,k]) & (i = j) \end{cases} \quad (7)$$

$$\overline{a}[i,j] = \begin{cases} min\left(0, \overline{r}[j,j] + \sum_{k \neq i,j} max(0, \overline{r}[k,j])\right) & (i \neq j) \\ \sum_{k \neq i,j} max(0, \overline{r}[k,j]) & (i = j) \end{cases} \quad (8)$$

2. Соединить все пары точек, для которых истинно следующее условие:

$$\begin{bmatrix} \overline{r}[i,j] \geq 0 \\ \overline{a}[i,j] + s[i,j] \geq max_{k \neq j}(\underline{a}[i,k] + s[i,k]) \end{bmatrix} \quad (9)$$

3. Для всех соединенных пар точек применить 1 и 2 шаги классического AP.
4. Для всех несоединенных пар точек применить следующие выражения:

$$r[i,j] = \rho[i,j] \quad (10)$$

$$a[i,j] = \gamma[i,j] \quad (11)$$

5. Выполнить 3 и 4 шаги классического AP.

Результаты кластеризации

В качестве объектов взяты данные идентификации пользователей-экспертов из социальной сети с хештегом #westworld из источника [5]. Данные имеют 1077 точек.

Алгоритм DBSCAN не является эффективным алгоритмом кластеризации для решения рассматриваемой задачи, так как данный алгоритм исключает самые непохожие на других объекты, которые часто являются лучшими экспертами. Но, если указать, что у объектов не должно быть соседей чтобы образовать кластер, то ни один объект не должен быть признан «шумом», и все объекты должны быть распределены по кластерам.

Анализируя полученные в процессе исследования результаты алгоритма DBSCAN, можно подчеркнуть, что при minPts равной 0, алгоритм не считает ни одну из точек шумом и не отбрасывает их, что дает в рамках решения рассматриваемой задачи нужные результаты. Однако алгоритм становится не устойчивым и время от времени все равно чистит часть выборки, считая точки шумом. Время выполнения DBSCAN очень мало: 0.983 секунды.

В дальнейших исследованиях применимости DBSCAN для кластеризации в рассматриваемой задачи нужно рассмотреть упрощенную версию алгоритма без удаления шума. Также необходимо рассмотреть вариант применения DBSCAN на выборке не содержащий точек, похожие на шум.

На рисунке 1 показаны результаты AP. Время выполнения AP равно 10720.607 секунд. В рамке на рисунке 1 показан фрагмент результата FAP, который отличен от результатов AP. Время выполнения FAP равно 5608.823 секунды.

Сравнивая AP и FAP можно сосчитать, что FAP работает быстрее AP на 5111.784 секунд, что почти в 2 раза быстрее.

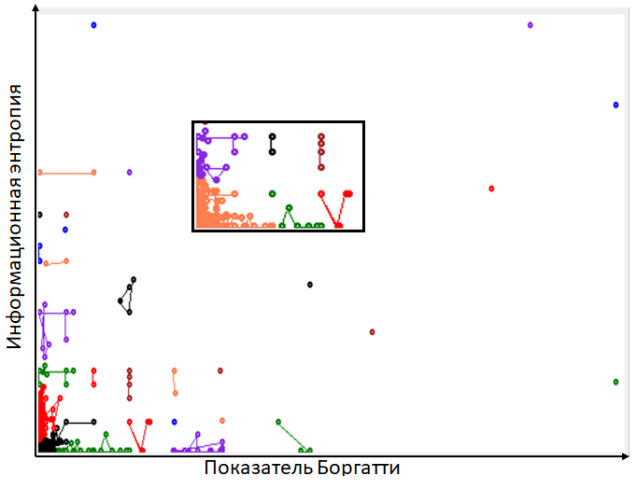

Рис. 1 - Результаты AP и FAP

Результаты FAP практически совпадают с результатами AP. Различие видны лишь на участках выборки плотного скопления точек.

Заключение

Алгоритм DBSCAN может применяться для решения рассматриваемой задачи при условиях, что у алгоритма указано, что точка не должна уметь соседей для того, чтобы попасть в кластер. Однако DBSCAN очень неустойчив в таких условиях и иногда срабатывает некорректно для решения рассматриваемой задачи.

AP однозначно можно применять для кластеризации в данной задачи, но время выполнения этого алгоритма очень велико. В ходе сравнения применения алгоритмов AP и FAP, можно сказать, что FAP работает в 2 раза быстрее, при этом результаты кластеризации практически совпадают с результатами AP. Различие можно увидеть лишь в области плотного скопления точек.

Работа выполнена при финансовой поддержке РФФИ (проект №17-07-00034 А).

Список использованных источников

1. Ortiz-Arroyo D. Discovering Sets of Key Players in Social Networks // Computational Social Networks Analysis. – 2010 – С. 27-47.
2. DBSCAN [Электронный ресурс] Wikipedia, Режим доступа: https://en.wikipedia.org/wiki/DBSCAN, свободный (дата обращения: 27.05.2018)
3. Интересные алгоритмы кластеризации, часть первая: Affinity propagation [Электронный ресурс] habrahabr, Режим доступа: https://habrahabr.ru/post/321216/, свободный (дата обращения: 27.05.2018)
4. Yasuhiro Fujiwara, Go Irie, Tomoe Kitahara Fast Algorithm for Affinity Propagation // Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, С 2238 – 2243.
5. Поиск экспертов в социальной сети Twitter [Электронный ресурс] Режим доступа: http://socgraph.tpu.ru/ProccessTwData, свободный (дата обращения 18.11.2018)