Лекция 17
Спектральная кластеризация

Затехано @TmKarter (Анищенко И.И.)
ФКН ВШЭ

9 мая 2020 г.

Методы кластеризации, которые были рассмотренны ранее на лекциях:

- K-means — метрический метод
- DBSCAN — density-based метод (алгоритм группирует вместе точки, которые тесно расположены)

В рамках этой лекции был разобран один из графовых методов кластеризации — **Спектральная кластеризация**. Основное полезное свойство метода заключается в формировании новых признаков для задачи кластеризации на исходных данных.

1 Построение графа

Так как спектральная кластеризация относится к графовым методам, то для начала по имеющимся данным нам необходимо построить граф. Граф по определению это: $G = (V, E)$, где $V = X = \{x_1, ..., x_\ell\}$ — вершины нашего графа в рамках нашей задачи, а ребра (E) между ними можно будет задать несколькими способами:

- полный граф, все рёбра с весами $w_{ij} = \exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$
- KNN-граф (вершину x_i соединим с k соседями)
- ε-граф (соединяем вершины, между которыми расстояние меньше заданной величины: $\rho(x_i, x_j) < \varepsilon$)

Далее по полученному графу можно вывести его **матрицу смежности** - W, где в ячейке W_{ij} записан 0, если вершины x_i и x_j не соединены, или записан его вес, если ребро между ними есть.
Так же запишем **матрицу степеней** вершин: $D = diag(d_1, ..., d_\ell)$, где $d_i = \sum_{j=1}^{\ell} w_{ij}$ - сумма весов рёбер, соседних с вершиной i.
Через эти две матрицы мы можем получить **лапласиан графа**: $L = D - W$, эта матрица нам понадобится дальше.

2 Лапласиан графа и его свойства

Рассмотрим свойства, которыми обладает лапласиан графа:

- $f \in R^\ell \to f^T L f = \frac{1}{2}\sum_{i,j=1}^{\ell} w_{ij}(f_i - f_j)^2$
- L - симметричная и неотрицательно определенная (неотрицательная определенность следует из свойства выше: $w_{ij} \geqslant 0$ - по определению используемых нами весов и $(f_i - f_j)^2 \geqslant 0$)
- третье свойство выведем в виде теоремы, которую далее докажем

Теорема 2.1.

1. *Собственное значение $\lambda = 0$ для матрицы L имеет кратность, равную числу компонент связности графа*
2. *$A_1, .., A_k$ - компоненты связности; $f_1 = ([x_i \in A_1])_{i=1}^{\ell}, ..., f_k$ - собственные вектора (это индикаторные векторы, показывающие к какой компоненте связности принадлежат вершины x_i), соответствующие $\lambda = 0$*

Доказательство.

Рассмотрим случай с числом компонент связности $k = 1$ (граф связный): разберемся, почему $\lambda = 0$ - собственное значение матрицы L: рассмотрим произведение $f^T L f$ с вектором $f = (1, ..., 1)$. Получим $f^T L f = \frac{1}{2}\sum_{i,j=1}^{\ell} w_{ij}(1-1)^2 = 0 = \lambda f$
Далее предположим, что для $\lambda = 0$ мы имеем какой-нибудь еще соответствующий собственный вектор f', который от предыдущего отличается тем, что имеет среди своих компонент различные значения (т.е. не константный, как предыдущий вектор). Тогда если $f' \neq const \to \exists p, q : f'_p \neq f'_q$ - т.е. в нашем новом векторе есть пара различных значений.
Далее вспомним, что данный граф имеет 1 компоненту связности, а значит между точками p и q будет иметься некий путь: $\exists p \to i_1 \to i_2 \to ... \to q$. Рассмотрим ребро (p, i_1) - оно есть в нашем графе, значит $w_{p,i_1} > 0$ и для того, чтоб слагаемое с этим ребром $w_{i,i_1}(f'_p - f'_{i_1})^2$ давало 0 в итоговую сумму $f'^T L f'$ (т.к. если f' - соб. вектор, то $f'^T L f' = f'^T * 0 = 0$) нам нужно, чтобы $f'_p = f'_{i_1}$. Аналогична ситуация и с ребром (i_1, i_2). И двигаясь дальше к вершине q мы получим, что $f'_p = f'_q$ - тем самым получив противоречие по значениям f_p и f_q, из которого можем вывести следующее: f' не будет собственным вектором для $\lambda = 0$.
Отсюда получаем, что собственному значению λ будет соответствовать только один собственный вектор: $f = const$, что и даем нам кратность для этого соб.зн. $1 = k$ - числу компонент связности. И сам вектор представляет из себя набор индикаторных значений принадлежности к классу, что и рассматривалось во 2 пункте

Случай с $k > 1$: в получившейся матрице L сможем упорядочить вершины так, чтобы получить блочно диагональную матрицу:

$$L' = \begin{pmatrix} L_1 & 0 & \dots & 0 \\ 0 & L_2 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & \dots & 0 & L_k \end{pmatrix}$$

где $L_1, L_2, ..., L_k$ - лапласианы компонент связности. Дальше вспомним, что спектр матрицы L (её мн-во собственных значений) в блочно диагональном виде матрицы будет характеризоваться, как объединение спектров маленьких матриц $L_1, L_2, ..., L_k$. У каждой этой матрицы в отдельности есть нулевое собственное значение с кратностью 1, и ему соотвествует единственный собственный вектор $f = const$.
Тогда при объединении этих спектров у $\lambda = 0$ кратность будет k, а собственными векторами к этому собственному значению будут вектора вида: $f_1 = ([x_i \in A_1])_{i=1}^{\ell}, ..., f_k$ (в векторе i компонента j будет принимать значение 0, если вершина x_j не лежит в компоненте связности i, и будет принимать 1 в противном случае) ■

Далее перейдем к гипотезе, которая не доказывается строго, но подтверждается экспериментально

Гипотеза 2.1. *У собственных векторов f_i, соответствующие маленьким собственным значениям, $f_{ij} \approx f_{ik} \Leftrightarrow x_j$ и x_k близки в графе (т.е. между ними расстояние в графе небольшое).*

Под маленькими собственными значениями мы подразумеваем набор из наименьших собственных значений матрицы L

3 Алгоритм спектральной кластеризации

На основе гипотезы выше можем вывести алгоритм спектральной кластеризации:

1. Строим $L = D - W$; сложность этого шага $O(\ell^2)$
2. Из лапласиана L находим $u_1, ..., u_k$ - нормированную систему из k собственных векторов, соответствующую минимальным собственным значениям; сложность $O(\ell^3)$
3. Составляем матрицу $U = (u_1|u_2|...|u_k)$ размера $\ell \times k$. Суть этой матрицы в следующем: по строчкам расположено описание ℓ объектов, а каждый вектор f_i записан в столбец. Тогда если мы рассмотрим признаки похожих объектов x_i и x_j (близки друг к другу в графе), то в каждая их пара будет приблизительно равна, тогда и $f_{ij} \approx f_{ik}$
4. Кластеризуем U с помощью $K - means$

И после получим нужную нам кластеризацию.

4 Внешние метрики

Теперь что-то стоит сказать про метрики, которые оценивают качество проводимой нами кластеризации. Для оценки качества будем пользоваться нашей моделью $a(x)$ разметкой с истинными ответами ($(y_1, ..., y_\ell)$ - *golden standard*). Фактически тут мы будем проводить supervised проверкой качества для unsupervised моделей, так как это единственный способ измерить качество моделей классификации.

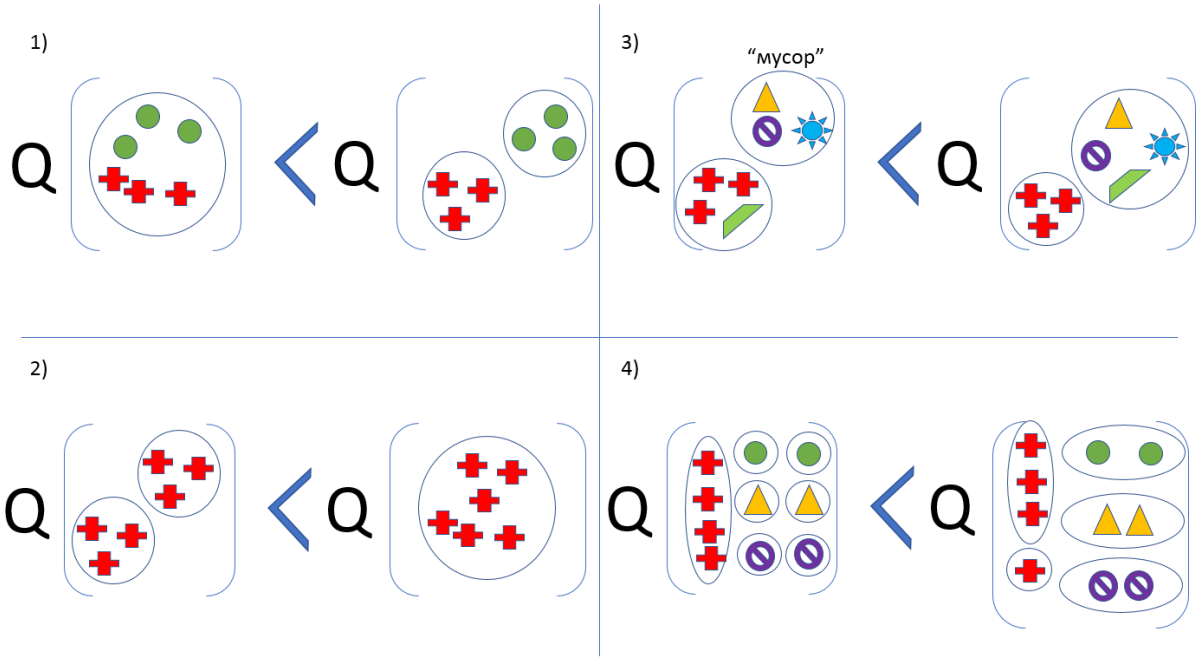

Рис. 1. Требования к правильному сравнению результатов кластеризации во внешних метриках

§4.1 Требования к внешней метрике

Так как имеющаяся у нас разметка не совсем является целевой переменной (к прим. в истинной разметке нам даны классы как метки к статьям: наука, политика, спорт. А наша модель выдает численные метки для разделения на классы) и это отображение истинных меток к нашей модели может быть подобранно не единственным способом, то было придуманно большое множество внешних методов для получения оценки качества. Но у них есть свои проблемы, которые заключаются в правильном сравнении результатов кластеризации. Лучшее разбиение то, где все объекты с одинаковой истинной меткой попали в один кластер или в разные кластеры? Лучше когда объекты с двумя видами меток попали в один кластер или в разные? Для детерминированности поведения в этих различных ситуациях были введены некоторые требования к внешней метрике (схематично они показаны на рис.1). Можно выделить 4 свойства, которыми она должна обладать:

1. Гомогенность. Если мы имеем кластер с несколькими видами меток, то для улучшения разбиения мы хотим потребовать, чтобы эти объекты были разделены по разным кластерам в зависимости от значения меток.

2. Полнота. Если мы имеем несколько кластеров с одним и тем же видом истинной метки, то для улучшения качества разбиения мы хотим “слить“ эти объекты в один кластер.

3. Ray Bay. Пусть мы имеем два кластера и один из них “мусорный“ (в нем много объектов с разными метками). А первый кластер содержит много объектов одного вида метки и несколько “чужих“ (объекты с другой меткой). Тогда наше

разбиение будет лучше, если лишние объекты из первого кластера окажутся в нашем “мусорном“ кластере.

4. Не имеет конкретного названия, но утверждает следующее. Пусть мы имеем большой кластер с объектами одного вида истинной метки и много других кластеров с другими видами меток (среди них есть кластеры с одинаковыми видами меток, но наша модель их выделила как отдельные группы). Если ценой разделения большого первого кластера на два отдельных (“отщипнули“ от большого кластера один объект) мы добьемся слияния нужных нам кластеров с другими метками (то, что нам нужно хорошо показано на картинке), то наша кластеризация станет лучше, чем была первоначально.

§4.2 Метрика BCubed

Если мы решим подыскать себе метрику со всеми выполняющимися 4 требованиями, то нам может подойти метрика **BCubed**. Для её понимания введём следующие обозначения: $L(x)$ - golden standard (истинная метка объекта), $C(x)$ - номер кластера объекта. Так же введём дополнительную меру

$$Correctness(x, x') = \begin{cases} 1;\ C(x) = C(x') \text{ и } L(x) = L(x') \\ 0;\ \text{иначе} \end{cases}$$

И на основе всего введём две составляющих этой метрики:

$$Precision\ BCubed = Avg_x(Avg_{(C(x')=C(x))}\ Correctness(x, x'))$$

$$Recall\ BCubed = Avg_x(Avg_{(L(x')=L(x))}\ Correctness(x, x'))$$

Далее от этих метрик можно посчитать F-меру, которая и будет удовлетворять всем 4 требованиям к внешним метрикам.