

О целочисленном линейном программировании для задачи кластеризации вершин графа

Моршинин Александр

Институт Математики им. С.Л. Соболева СО РАН

Исследование выполнено за счет гранта Российского научного фонда

№ 22-71-10015, <https://rscf.ru/project/22-71-10015/>

Задачи кластеризации

Задачи кластеризации являются важной частью анализа данных. В них необходимо разбить заданное множество объектов на несколько подмножеств (**кластеров**) основываясь на схожести объектов между собой.

В задаче **кластеризации вершин графа (correlation clustering)** требуется разбить вершины графа на кластеры, опираясь на реберную структуру этого графа.

Основные определения

Обыкновенный граф называется **кластерным графом**, если каждая компонента связности этого графа является полным графом. Компоненты связности называются *кластерами*.

Обозначим через $C(V_1, V_2, \dots, V_s)$ кластерный граф, содержащий s компонент.

Расстоянием $\rho(G_1, G_2)$ между помеченными графами $G_1 = (V, E_1)$ и $G_2 = (V, E_2)$ называется величина

$$\rho(G_1, G_2) = |E_1 \setminus E_2| + |E_2 \setminus E_1|.$$

Задачи кластеризации вершин графа

СС. Для графа $G = (V, E)$ найти ближайший к G кластерный граф $C(V_1, V_2, \dots, V_s)$ с не более чем $2 \leq s \leq |V|$ кластерами.

СС _{$\leq k$} . Для графа $G = (V, E)$ найти ближайший к G кластерный граф $C(V_1, V_2, \dots, V_s)$ с не более чем $2 \leq s \leq k < |V|$ кластерами.

Все задачи NP -трудны.

Целочисленное линейное программирование

Чарикар, Гурусвами и Вирт (2005) разработали модель ЦЛП для задачи СС.

$$x_{ij} = \begin{cases} 0, & \text{если } i \text{ и } j \text{ лежат в одном кластере} \\ 1, & \text{иначе} \end{cases}$$

Если $x_{ij} = 0$ и $x_{jr} = 0$, то $x_{ir} = 0$. Тогда $x_{ir} \leq x_{ij} + x_{jr}$ для всех $i, j, r \in V$. Другими словами, кластерный граф не может содержать простую цепь P_2 как порожденный подграф.

Целочисленное линейное программирование

$$\begin{aligned} \sum_{ij \in E} x_{ij} + \sum_{ij \notin E} (1 - x_{ij}) &\rightarrow \min \\ x_{ir} &\leq x_{ij} + x_{jr} \text{ для всех } i, j, r \in V, \\ x_{ij} &\in \{0, 1\} \text{ для всех } i, j \in V. \end{aligned}$$

Вахид и Хасанни (2022) в литературном обзоре задач кластеризации вершин графа добавили следующее ограничение:

$$x_{ij} = x_{ji} \text{ для всех } i, j \in V.$$

Целочисленное линейное программирование

Мы можем переписать неравенство треугольника, чтобы использовать только одну переменную для каждой пары вершин.

$$\begin{aligned} & \sum_{ij \in E} x_{ij} + \sum_{ij \notin E} (1 - x_{ij}) \rightarrow \min \\ & \begin{cases} x_{ir} \leq x_{ij} + x_{jr} \\ x_{ij} \leq x_{ir} + x_{jr} \\ x_{jr} \leq x_{ij} + x_{ir} \end{cases} \text{ для всех } i, j, r \in V, \\ & x_{ij} \in \{0, 1\} \text{ для всех } i, j \in V. \end{aligned}$$

ЦЛП для задач $CC_{\leq k}$

Для задачи $CC_{\leq k}$ допустимое решение не может содержать **пустой граф** O_{k+1} как порожденный подграф:

$$x_{i_1 i_2} + \dots + x_{i_k i_{k+1}} \leq \frac{(k+2)(k-1)}{2} \text{ для всех } i_1, \dots, i_{k+1} \in V.$$

Заметим, что в обе модели содержат $O(n^2)$ вершин и $O(n^{k+1})$ ограничений.

Рассмотрим другие подходы, один из которых позволяет значительно сократить количество переменных и ограничений.

Убираем неравенство треугольника

Если перейти от неравенства треугольника к сумме трех переменных $x_{ir} + x_{ij} + x_{jr}$, то можно избавиться от симметричных переменных, но возникает следующее неравенство: $x_{ir} + x_{ij} + x_{jr} \neq 1$. Его можно линеаризовать введением дополнительных переменных

$$\begin{aligned}x_{ir} + x_{ij} + x_{jr} - 1 &\geq \varepsilon - (1 - y_{ijr})M \text{ для всех } i, j, r \in V, \\x_{ir} + x_{ij} + x_{jr} - 1 &\leq -\varepsilon + y_{ijr}M \text{ для всех } i, j, r \in V.\end{aligned}$$

Такой подход требует $O(n^3)$ дополнительных переменных, при этом количество ограничений снижается константно.

ЦЛП для задачи $CC_{\leq 2}$

Для каждой вершины $i \in V$ введем следующие переменные:

$$x_i = \begin{cases} 0, & \text{если } i \in V_1 \\ 1, & \text{иначе } i \in V_2 \end{cases}$$

Случай 1. $ij \in E$. Тогда $|x_i - x_j| = 0$ при $x_i = x_j$ и $|x_i - x_j| = 1$ при $x_i \neq x_j$.

Случай 2. $ij \notin E$. Тогда $|x_i + x_j - 1| = 0$ при $x_i \neq x_j$ и $|x_i + x_j - 1| = 1$ при $x_i = x_j$.

ЦП для задачи $CC_{\leq 2}$

$$\sum_{ij \in E} |x_i - x_j| + \sum_{ij \notin E} |x_i + x_j - 1| \rightarrow \min$$
$$x_i \in \{0, 1\} \text{ для всех } i \in V.$$

Данная модель не линейна. Сделаем следующий известный трюк. Представим каждый модуль в целевой функции как сумму двух бинарных переменных $u_{ij} + v_{ij}$. Выражения под модулем представим как разность $v_{ij} - u_{ij}$.

ЦЛП для задачи $CC_{\leq 2}$

$$\sum_{i,j \in V} u_{ij} + v_{ij} \rightarrow \min$$

$$x_i - x_j + u_{ij} - v_{ij} = 0 \text{ для всех } i, j \in V, ij \in E,$$

$$x_i + x_j - 1 + u_{ij} - v_{ij} = 0 \text{ для всех } i, j \in V, ij \notin E,$$

$$x_i \in \{0, 1\} \text{ для всех } i \in V,$$

$$u_{ij} \in \{0, 1\} \text{ для всех } i, j \in V,$$

$$v_{ij} \in \{0, 1\} \text{ для всех } i, j \in V.$$

Эта модель содержит $O(n^2)$ переменных и $O(n^2)$ ограничений.

Унитарный код для задачи $CC_{\leq k}$

Для задачи $CC_{\leq k}$, $k \geq 3$ нельзя описать принадлежность вершин кластерам одной переменной. Используем **унитарный код**:

$$x_{is} = \begin{cases} 1, & \text{если вершина } i \text{ принадлежит } s\text{-ому кластеру} \\ 0, & \text{иначе} \end{cases}$$

Для каждой вершины $i \in V$ строится бинарный вектор $[x_1, \dots, x_k]$ с одной компонентой, равной 1.

Унитарный код для задачи $CC_{\leq k}$

Случай 1. $ij \in E$. Тогда $\frac{1}{2} \sum_{s=1}^k |x_{is} - x_{js}| = 0$ если i и j принадлежат одному кластеру, иначе $\frac{1}{2} \sum_{s=1}^k |x_{is} - x_{js}| = 1$.

Случай 2. $ij \notin E$. Тогда $\frac{1}{2} \left(\sum_{s=1}^k |x_{is} + x_{js} - 1| - k + 2 \right) = 0$ если i и j принадлежат разным кластерам иначе,

$$\frac{1}{2} \left(\sum_{s=1}^k |x_{is} + x_{js} - 1| - k + 2 \right) = 1.$$

ЦЛП для задачи $CC_{\leq k}$

$$\sum_{i,j \in V} \sum_{s=1}^k u_{ijs} + v_{ijs} \rightarrow \min$$

$$x_{is} - x_{js} + u_{ijs} - v_{ijs} = 0 \text{ для всех } i, j \in V, ij \in E, s = 1, \dots, k,$$

$$x_{is} + x_{js} - 1 + u_{ijs} - v_{ijs} = 0 \text{ для всех } i, j \in V, ij \notin E, s = 1, \dots, k,$$

$$\sum_{s=1}^k x_{is} = 1 \text{ для всех } i \in V,$$

$$x_i \in \{0, 1\} \text{ для всех } i \in V,$$

$$u_{ij} \in \{0, 1\} \text{ для всех } i, j \in V,$$

$$v_{ij} \in \{0, 1\} \text{ для всех } i, j \in V.$$

ЦЛП для задачи $SS_{\leq k}$

Для каждой вершины необходимо k переменных, для каждой пары вершин требуется ограничение. Таким образом, эта модель содержит $O(k^2 n^2)$ переменных и $O(k^2 n^2)$ ограничений.

Экспериментальное исследование

Тестировалось 3 модели для задачи $CC_{\leq 2}$ - неориентированная треугольная (**TR**), с «неравно» (**IN**) и с модулем (**MOD**). Все модели были реализованы в Python с помощью библиотеки Python-MIP. Использовался NEOS Server с 4 ядрами, решатель IBM ILOG CPLEX.

Случайные графы $G(n, p)$ генерировались с помощью *модели Эрдеша-Реньи* со значением параметра $p \in \{0.33, 0.5, 0.67\}$.

Оптимальные решения находились в зависимости от значений параметров n и p . Для каждого набора параметров было решено 100 задач.

Необходимая память для моделей

	TR	MOD	IN
20	0.56	0.04	0.60
25	1.13	0.07	1.23
30	2.02	0.10	2.18
35	3.29	0.15	3.55
40	4.98	0.20	5.40
45	7.17	0.25	7.79
50	9.94	0.32	10.81
55	13.36	0.39	14.50
60	17.61	0.46	18.97
65	22.61	0.55	24.37
70	28.46	0.64	30.73
75	35.23	0.74	38.05
80	42.97	0.84	46.47
85	51.75	0.96	55.99
90	61.65	1.07	66.80
95	72.72	1.20	78.83

Среднее время работы

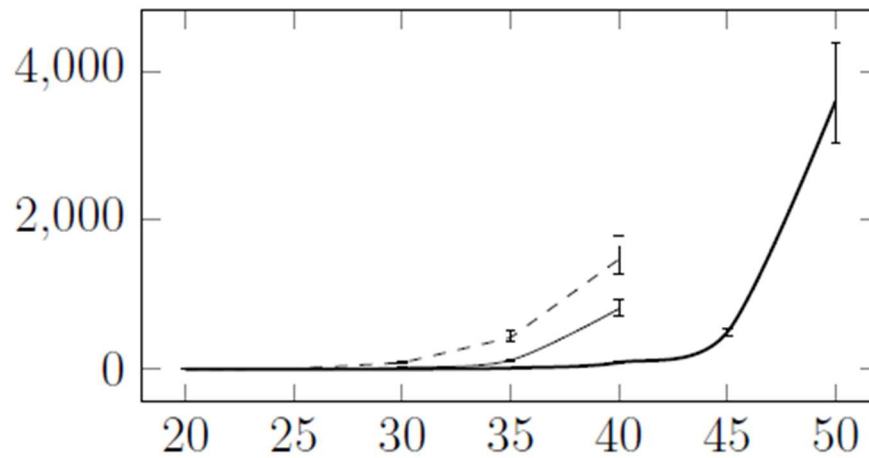
n/p	0.33			0.5			0.67		
	TR	MOD	IN	TR	MOD	IN	TR	MOD	IN
20	0.5	0.1	1.8	0.7	0.2	1.7	0.2	0.2	0.6
25	3.7	1.2	9.7	5.2	2.1	12.5	0.9	0.6	2.6
30	21.1	4.5	89.7	43.8	9.7	179.8	3.4	3.1	13.3
35	123.1	18.2	424.8	240.3	36.3	1344.9	7.3	6.7	37.4
40	812.4	96.2	1476.1	952.1	162.9	3612.7	16.6	24.3	131.7
45	-	499.4	-	-	623.7	-	28.8	26.1	243.3
50	-	3598.5	-	-	2153.2	-	63.9	53.1	407.9
55	-	-	-	-	-	-	85.4	58.8	595.2
60	-	-	-	-	-	-	-	94.9	-
65	-	-	-	-	-	-	-	123.5	-
70	-	-	-	-	-	-	-	203.8	-
75	-	-	-	-	-	-	-	334.7	-
80	-	-	-	-	-	-	-	587.8	-
85	-	-	-	-	-	-	-	993.1	-
90	-	-	-	-	-	-	-	1560.5	-
95	-	-	-	-	-	-	-	2694.6	-

Доверительный интервал среднего времени работы

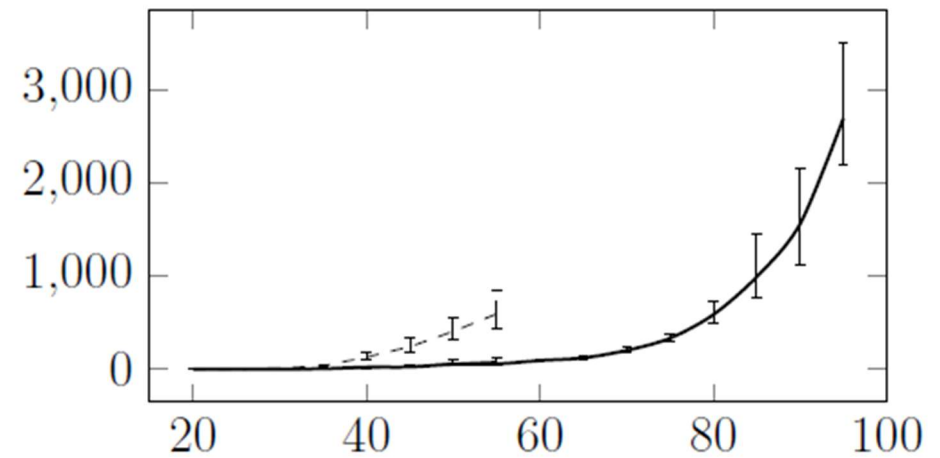
Точечную оценку времени работы нельзя назвать достоверной, поскольку эта величина является случайной. Лучше построить интервальную оценку на некотором уровне значимости.

Поскольку время работы алгоритма не является нормально распределенной случайной величиной, доверительный интервал строился с помощью процедуры бутстрепа. В данном случае использовалась реализация метода бутстрепа с коррекцией смещения и ускорением (BCa) из библиотеки SciPy со значениями параметров $B = 10000$ и $\alpha = 0.05$.

Доверительный интервал среднего времени работы



(a) $p = 0.33$



(b) $p = 0.67$

— TR — MOD --- IN

Спасибо за внимание.