Выделение сообществ в гиперграфах.

Насыров Руслан Рашидович
студент 2 курса,
физтех-школа прикладной математики и информатики,
Московский физико-технический институт(ГУ)
РФ, г. Долгопрудный
E-mail: nasyrov.rr@phystech.edu
Васильева Екатерина Евгеньевна,
Кандидат физ.-мат. наук
Мусатов Даниил Владимирович,
Кандидат физ.-мат. наук, кафедра дискретной математики МФТИ
РФ, Долгопрудный

14 мая 2022 г.

Аннотация

Проблема кластеризации гиперграфов является естественным усложнением задачи кластеризации графов. И если последняя задача уже широко изучена, кластеризация гиперграфов до сих пор остается малоисследованной областью. В данной статье изучается формирование кластеров реальных гиперграфов при ограничениях на размеры его ребер (остаются только ребра с размером $<= 2, <= 3...$). Исследуются 5 гиперграфов. Для нахождения кластеров использованы Лувенский алгоритм и специальный алгоритм для поиска кластеров в гиперграфах - HSBM, предложенный в статье Philip S. Chodrow[11]. Также предлагаются функции для оценки, насколько кластеры «слипаются» при добавлении ребер в гиперграф.

1 Введение.

Графы позволяют выражать взаимодействия в системах. Они являются частным случаем гиперграфов, в которых каждое ребро может соединять множество вершин. Последние в свою очередь все чаще применяются для анализа современных сетей, поскольку сложные взаимодействия часто происходят не между парой объектов, а внутри какой-то группы. Таким образом, гиперграфы дают возможность выразить взаимодействия более точно, чем обычные графы.

При изучении реальных графов было обнаружено, что они обладают рядом различных характеристик, таких как свойство *малого мира* [12], свойство *безмасштабности* [13] и *структурой сообществ*.

Говорят, что граф имеет *структуру сообщества (или кластеризацию)*, если его вершины могут быть легко сгруппированы в множества, вершины которых имеют плотные связи внутри множества и слабо связаны с другими множествами [1].

Структуры сообществ встречаются почти во всех реальных сетях. Например, в социальных сетях это могут быть люди, живущие в одном районе, или дети, ходящие в одну школу. Поиск структуры сообществ таких сетей существенно упрощает их анализ, так как позволяет перейти от рассмотрения точечных взаимодействий между парами вершин к взаимодействиям между сообществами (кластерами). Такой подход применим, так как вершины, состоящие в одном кластере с большой вероятностью обладают одинаковыми наборами характеристик и нет смысла рассматривать несколько похожих вершин по отдельности. Сообщества помогают изучать граф, так как могут иметь свойства, сильно отличающиеся от средних свойств сети. То есть рассматривая только средние характеристики мы упускаем важные детали и признаки, которыми обладают сообщества. Например, анализ лишь средних характеристик упускает такой момент, как существование общительных и малообщительных групп в социальной сети одновременно. [10]

Проблема выделения кластеров в гиперграфах имеет большое значение для множества прикладных задач, таких как параллельные вычисления, сегментации изображений, составления схем [11]. Алгоритмы выделения сообществ применяются для анализа сетей взаимодействия белков. В них сообщества соответствуют белкам с аналогичной функциональностью внутри биологической клетки. А этот анализ уже

применяется для создания лекарств. [14] Еще одним из важных приложений является прогнозирование недостающих ребер и выявление ложных ребер в сети. Так как в процессе сбора информации могут встречаться ошибки, то при небольшом их количестве с помощью выделения сообществ их легко выявить и устранить.

Кластеризация графов уже хорошо изучена и есть множество алгоритмов, ее находящих. Они основаны на разных идеях: *метод минимального разреза* [8] - минимизирует количество ребер между кластерами, *алгоритм Гирвана-Ньюмана* [9] - находит те ребра, которые с большой вероятностью лежат «между» сообществами, *Лувенский алгоритм* [5] - использует иерархическую кластеризацию и метод максимизации модульности [6]. Именно он является одним их самых используемых, поскольку сочетает в себе эффективность и быстроту работы.

Задача кластеризации гиперграфов насчитывает меньшее число алгоритмов. Мы остановимся на алгоритме HSBM (hypergraph stochastic block model) [11], который использует генеративный подход и работает схожим с Лувенским алгоритмом способом.

В статье используются 5 гиперграфов: Contact-primary school, Contact-high-school, NDC-classes, House Committees, Senate Committees [2].

В статье изучается, как формируются кластеры в реальных гиперграфах. А именно, назовем $k-$*ограничением* гиперграфа гиперграф, полученный удалением из исходного его ребер, размером $> k$. Тогда в статье исследуется, как при переходе от k к $k+1$ ограничению меняется кластеризация. Кластеризация вычисляется 2 методами: Лувенским алгоритмом для кликовой проекции гиперграфа-ограничения и алгоритмом HSBM.

Приведено сравнение этих 2 методов кластеризации на 3 гиперграфах. Показано, что HSBM находит более «устойчивые» кластеры, т.е. те, которые при увеличении ограничения (добавлении новых ребер) не дробятся (или дробятся несильно).

Проверяется гипотеза, что кластеры предыдущего ограничения будут объединяться друг с другом, формируя кластеры следующего. Для количественного измерения этого объединения предлагаются 5 функций, основанные на разных идеях и интуитивных соображениях. Для визуального наблюдения за поведением кластеризации представлены графики формирования самых больших кластеров.

2 Ключевые понятия.

Определение 1 *Гиперграф H - это пара H=(V, E) , где V - множество вершин, а E семейство подмножеств V, называемых гиперребрами. [16]*

Определение 2 *Сеть со структурой сообщества - такая сеть, узлы которой могут быть легко сгруппированы в кластеры таким образом, чтобы внутри кластеров лежало много ребер, а между ними - мало. [1]*

Определение 3 *Кликовая проекция гиперграфа $H = (V, E)$ весовой функцией f - это взвешенный граф $G = (V, E')$, где вес каждого ребра $w(v_1, v_2) = \sum_{e \in E} I[v_1 \in e \wedge v_2 \in e] f(e, v_1, v_2)$. Это определение достаточно общее. Далее будет использоваться только функция $f(e) = \frac{e.w}{|e|^{alpha}}$, где e.w - вес ребра ($= 1$ в невзвешенном случае), $|e|-$ размер гиперребра.*

Определение 4 *Кластеризация сети - процесс нахождения структуры сообщества сети. То есть процесс разбиения вершин на кластеры.*

Определение 5 *$k-$ограничение гиперграфа H - это гиперграф $H|_k$, полученный из H удалением всех ребер, содержащих $> k$ вершин.*

3 Используемые алгоритмы кластеризации.

3.1 Алгоритм кластеризации гиперграфа HSBM [11]

Введем обозначения: R_k - все гиперребра графа, имеющие размер k, a_R - вес гиперребра R ($= 1$ для невзвешенного случая), z_R - кластеризация гиперребра R (проекция кластеризации z на гиперребро R), d_i- степень вершины i.

$m_k(H) = m_k = \sum_{R \in R^k} a_R$ сумма весов гиперребер, состоящих из k вершин.

$cut_k(z) = m_k - \sum_{R \in R_k} a_R \delta(z_R)$, где $\delta(z_R) = 1$, если данное гиперребро лежит в 1 кластере целиком (функция кронекера).

$vol(label) = \sum_{i=1}^{n} d_i \delta(z_i, label)$– суммарная степень всех вершин в данном кластере. Где label - метка какого-то кластера, z_i– метка вершины i в кластеризации.

$vol(H) = \sum_{i=1}^{n} d_i$ - «объем» гиперграфа H. Где d_i - степень вершины i.

Реализован алгоритм кластеризации гиперграфа, предложенный в статье [11]. Основная функция - функция модульности:

$$Q(z, \Omega, d) = -\sum_{i=1}^{k} \beta_i [cut_i(z) + \gamma_i \sum_{l=1}^{L} vol(l)^i]$$

Где $\beta_k = log(w_{k1}) - log(w_{k0}), \gamma_k = \beta_k^{-1}(w_{k1} - w_{k0})$, $cut_k(z) = m_k - \sum_{R \in R_k} a_R(z_R)$.

Из strict modularity: $\beta_k = 1$ и $\gamma_k = \frac{m_k}{(vol(H))^k}$.

Кластеризация очень сильно зависит от параметров β, γ.

β отвечает за то, какого размера гиперребра самые важные (изначально взяты $\beta = 1$ для всех размеров).

γ отвечает за размеры получающихся кластеров. При уменьшении γ кластеризация будет дробиться.

3.2 Алгоритм кластеризации гиперграфа через проекции.

Алгоритм:

1. Фиксируем проектирующую функцию $f(e, v_1, v_2) : E \times V \times V \to \mathcal{R}_+$
2. Проектируем гиперграф H на взвешенный граф G по следующему правилу: в каждом гиперребре e каждая пара вершин $v_1, v_2 \in e$ вносит свой вклад в вес ребра проекции в соответствии с функцией $f : f(e, v_1, v_2)$.
3. Вес ребра (v_1, v_2) в G равен $w(v_1, v_2) = \sum_{e \in E} I[v_1 \in e \wedge v_2 \in e] f(e, v_1, v_2)$.
4. Для графа G вычисляем кластеризацию Лувенским алгоритмом.
5. Переносим кластеризацию G на H.
6. Оцениваем качество кластеризации с помощью функции Q (приведена выше).

В качестве проектора-функции f будем брать $f(e, v_1, v_2) = \frac{e.w}{|e|^\alpha}$ при различных alpha, где $e.w$– вес гиперребра (или $= 1$ в невзвешенном случае).

При α близких к 0 получается обычная кликовая проекция, близких к 1 - такую, которая сохраняет степени вершин (если вершина v была в ребре e, то после проектирования вклад гиперребра e в ее степень будет $= 1$).

3.3 Метрики объединения кластеров.

Задача состоит в следующем: даны 2 кластеризации z, z_{new}. Нужно определить количественную меру, которая показывает, насколько z является подкластеризацией z_{new} (то есть насколько кластеры из z_{new} являются объединением кластеров z).

Введем обозначения: $\mathcal{C} = \{C_1, ..., C_l\}$ - совокупность всех кластеров кластеризации z, $\mathcal{D} = \{D_1, ..., D_r\}$ - совокупность всех кластеров кластеризации z_{new}.

Пусть $C \in \mathcal{C}$, тогда $\mathcal{D}_C = \{D \in \mathcal{D} : D \cap C \neq \varnothing\}$ - совокупность кластеров, на которые в новой кластеризации раздробились вершины кластера C.

Обозначим k_C– количество кластеров, на которые разбился кластер C в новой кластеризации.

Рассмотрим в кластеризации z какой-нибудь кластер C, вершины которого в новой кластеризации z_{new} попали в кластеры $\{D_1, ..., D_{k_C}\} = \mathcal{D}_C$.

Обозначим $C_i = D_i \cap C \neq \varnothing, i = 1, ..., k_c$– те вершины кластера C, которые попали в D_i.

Нам нужно ввести функцию $f(C, \mathcal{D}_C)$, которая будет «штрафовать» кластер C, за то что он «разбился». Эта функция зависит как от кластера C, так и от его разбиения в новой кластеризации.

Также нужно агрегировать эти штрафы в итоговую функцию потерь для изменения кластеризации $L(\mathcal{C}, \mathcal{D})$.

Для простоты будем считать, что агрегация проводится просто суммированием:

$$L(\mathcal{C}, \mathcal{D}) = \sum_{C \in \mathcal{C}} f(C, \mathcal{D}_C)$$

Сформулируем требования к функции-штрафу f:

1. Если $k_C = 1$, то $f = 0$ (так как кластер не разбился).

2. Чем меньше $|C|$, тем меньше штраф, чем больше $|C|$, тем больше штраф при тех же пропорциях разделения (так как нам в основном интересно только что происходит с большими кластерами).

3. Чем ближе $|C_i|$ к $\frac{|C|}{k_C}(k_C > 1)$, тем больше штраф (так как происходит сильное дробление).

4. Чем больше $|D_i|$ по сравнению с $|C|$, тем меньше штраф (нам хочется, чтобы следующая кластеризация была укрупнением предыдущей).

5. Чем меньше $|C_i|$, тем штраф меньше (нас интересуют только большие кластеры).

Возможные функции:

1. Количество кластеров z, которые разбиты несколькими кластерами z_{new}.

$$f_1(z, z_{new}) = \sum_{C \in \mathcal{C}} I[\exists v_1, v_2 \in C : z_{new}[v_1] \neq z_{new}[v_2]].$$

2. Для каждого старого кластера, который разделился на k, в штраф добавляется k-1 (k_C- количество кластеров, на которые разбился кластер C).

$$f_2(z, z_{new}) = \sum_{C \in \mathcal{C}} (k_C - 1)$$

3. Суммарное количество пар вершин, которые раньше были в 1 кластере, а теперь в разных.

$$f_3(z, z_{new}) = \sum_{C \in \mathcal{C}} \sum_{v_1, v_2 \in C} I[z_{new}[v_1] \neq z_{new}[v_2]]$$

4. Штраф, учитывающий количество кластеров, на которые разделился старый кластер (чем больше $|C| - |C_i|$, тем лучше):

$$f_{ord}(C) = \sum_{i=1}^{k_C} \frac{|C| - |C_i|}{k_C} = |C| - \sum_{i=1}^{k_C} \frac{|C_i|}{k_C} = |C|(1 - \frac{1}{k_C})$$

   В данном случае рассмотрим такую агрегацию:

$$L(\mathcal{C}, \mathcal{D}_C) = \sum_{C \in \mathcal{C}} \frac{f_{ord}(C)}{|C|^{1-\alpha}} = \sum_{C \in \mathcal{C}} |C|^{\alpha}(1 - \frac{1}{k_C}), 0 \leq \alpha \leq 1.$$

   Здесь параметр α регулирует, насколько больший вклад дают большие кластеры по сравнению с маленькими. Если $\alpha = 0$, то вклады одинаковые, если $\alpha = 1$, то получаем обычную агрегацию суммой.

5. Штраф, который учитывает все указанные критерии. Введем функцию частичного штрафа $h(C, C_i)$ (C разбился на кластеры $C_1, ..., C_{k_C}$).

   Тогда при $k_C = 1 : h = 0$. Если $k_C \neq 1$:

$$h(C, C_i) = \frac{|C_i|^{1-\alpha}}{|\ln(|C|) - \ln(|C_i|k_C)| + \beta} \times \frac{|C|}{|D_i|}$$

   Далее

$$f_{smart}(C) = \sum_{i=1}^{k_C} h(C, C_i)$$

4 Эксперименты на гиперграфах.

4.1 граф Contact-primary school

Характеристики[11]:

1. Вершин - 242
2. Ребер - 12,704
3. Средняя степень - 127
4. Средний размер ребра: 2.4
5. Размеры ребер: [2, 3, 4, 5]
6. Реальное число кластеров: 11

4.1.1 Добавление ребер в порядке увеличения размера.

Посмотрим, как меняется кластеризация по методу HSBM и по методу проекций. На графике изображены значения функций f_1, f_2, f_3 между парами соседних проекций и количество найденных кластеров для данной проекции.

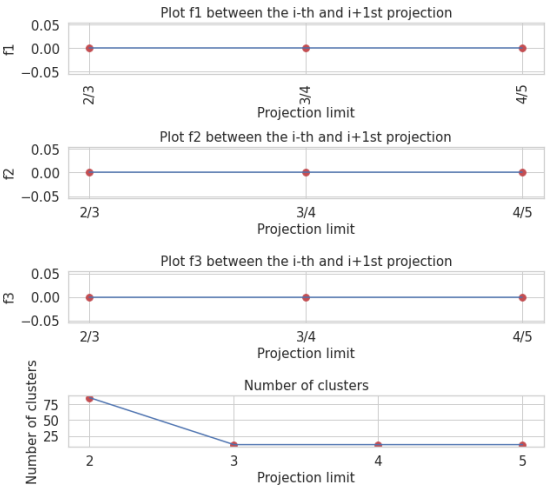
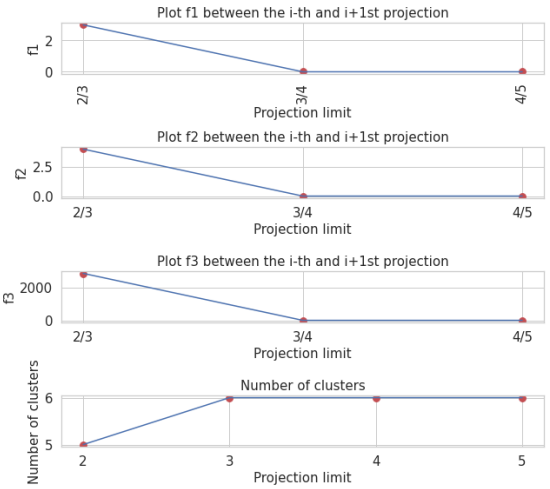
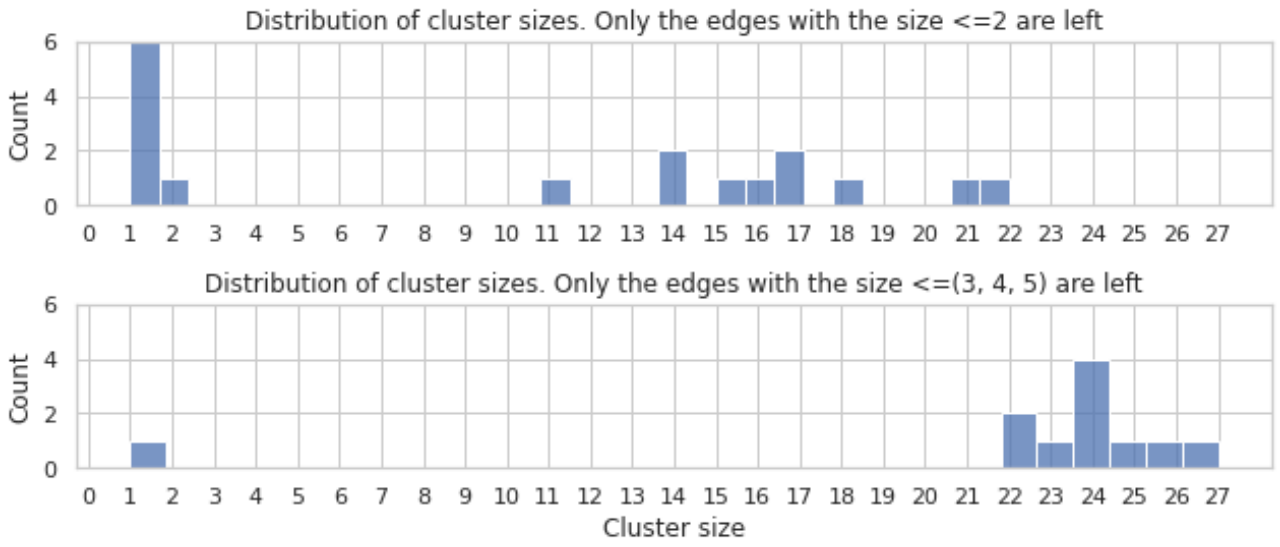


Рис. 1: Кластеризация методом HSBM



Кластеризация методом проекций ($\alpha = 1$)

А вот распределение размеров кластеров при кластеризации HSBM.



Теперь посмотрим, как формируются кластеры 2-й проекции из кластеров 1-й проекции:
Будем смотреть только на топ-k (≤ 12) кластеров с наибольшими размерами и как они изменяются.
График формируется так:

1. Столбцы обозначают кластеры предыдущей кластеризации, строки - следующей.
2. В последней строке - размеры топ-k самых больших кластеров предыдущей кластеризации.
3. В правом столбце - размеры самых больших кластеров в следующей кластеризации.
4. На пересечении строк и столбцов - размер пересечения соответствующей пары из старого и нового кластера.

Аналогично устроен график различия между найденной и истинной кластеризацией: внизу - истинные кластеры, справа - найденные.

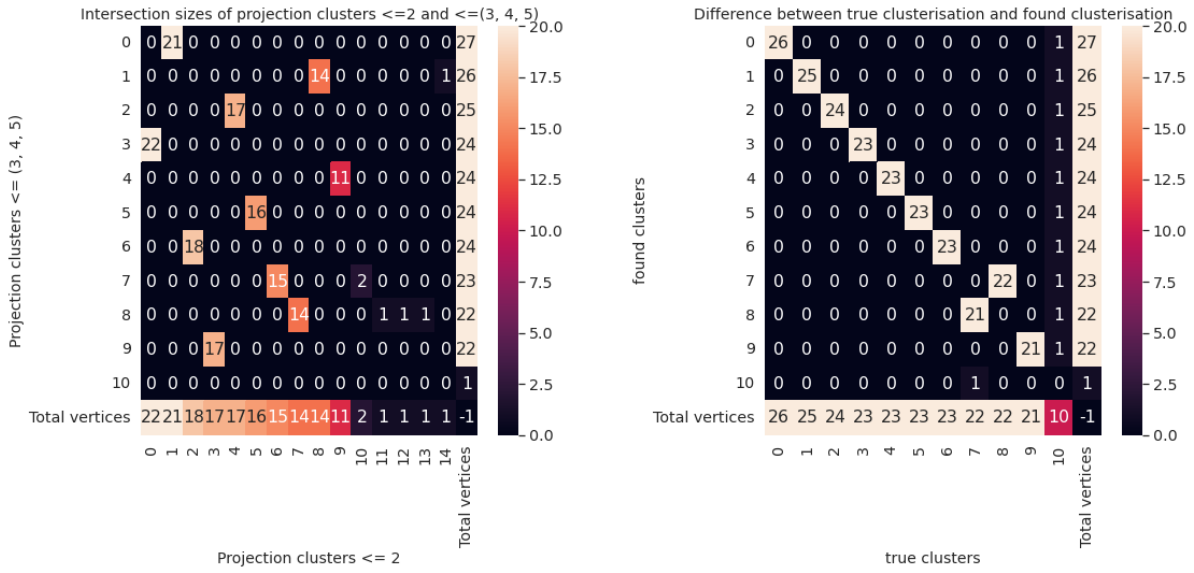
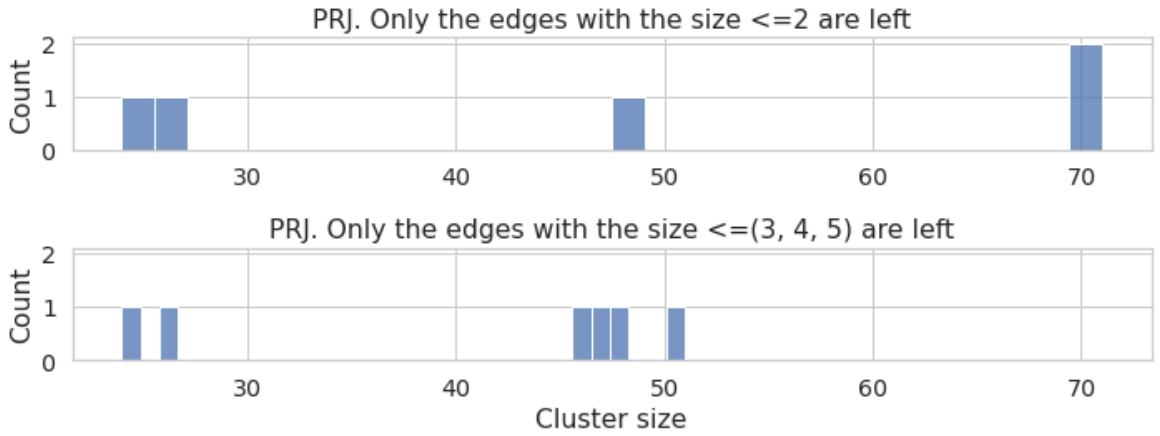


Рис. 2: Пересечения кластеров: 2-3 проекций истинных и найденных ($\alpha = 1$)

Видим: на проекции 2 есть несколько больших кластеров (10 штук), остальные - размера 1. На следующей проекции большие кластеры переходят каждый в свой отдельный, а меленькие - присоединяются к большим.

Также видно (правый график) что кластеры восстановлены почти идеально (только кластер размера 10 не распознался и был распределен по всем остальным).
А вот аналогичный результат, если кластеризацию проводить Лувенским алгоритмом через взвешенную проекцию.



Как видим, кластеров меньше, чем ожидалось (6 против 11). И распределение их размеров не соотносится с истинным.

Теперь посмотрим, как формируются кластеры 2-й проекции из кластеров 1-й проекции и на взаимосвязь найденных кластеров с истинными.

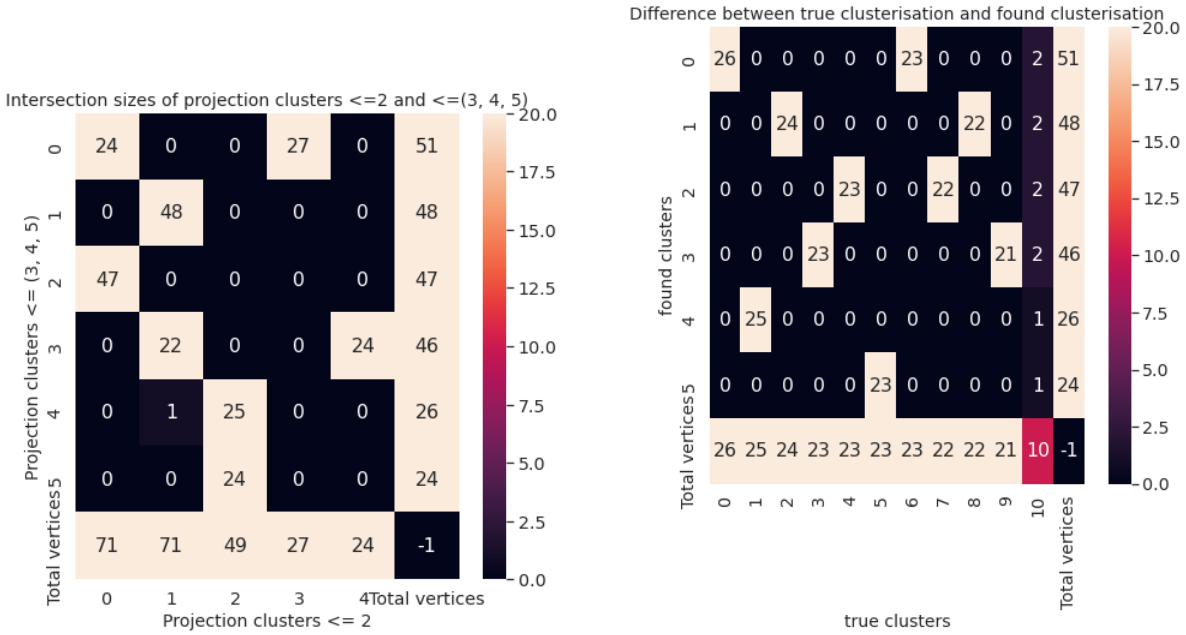


Рис. 3: Пересечения кластеров: 2-3 проекций истинных и найденных ($\alpha = 1$)

Вывод: при переходе от 2- к 3-проекции три кластера старой кластеризации разбилось на 2 большие части. Это отразилось на значении функции f_3 (она более 2000).

Итоговое количество количество классов получилось 6, что намного меньше желаемого: 11.

Мы видим, что получившиеся кластеры - это либо целевые кластеры. либо объединение 2 целевых кластеров.

В целом это говорит о том, что параметр разрешения (в алгоритме поиска кластеризации) выставлен неподходящий.

4.2 Contact-high-school

Характеристики[11]:

1. Вершин - 327
2. Ребер - 7,818
3. Средняя степень - 55.6
4. Средний размер ребра: 2.3
5. Размеры ребер: [2, 3, 4, 5]
6. Реальное число кластеров: 9

Вот график, показывающий, насколько хорошо происходит объединение кластеров.

И распределение количества кластеров при различных проекциях, для кластеризации HSBM.

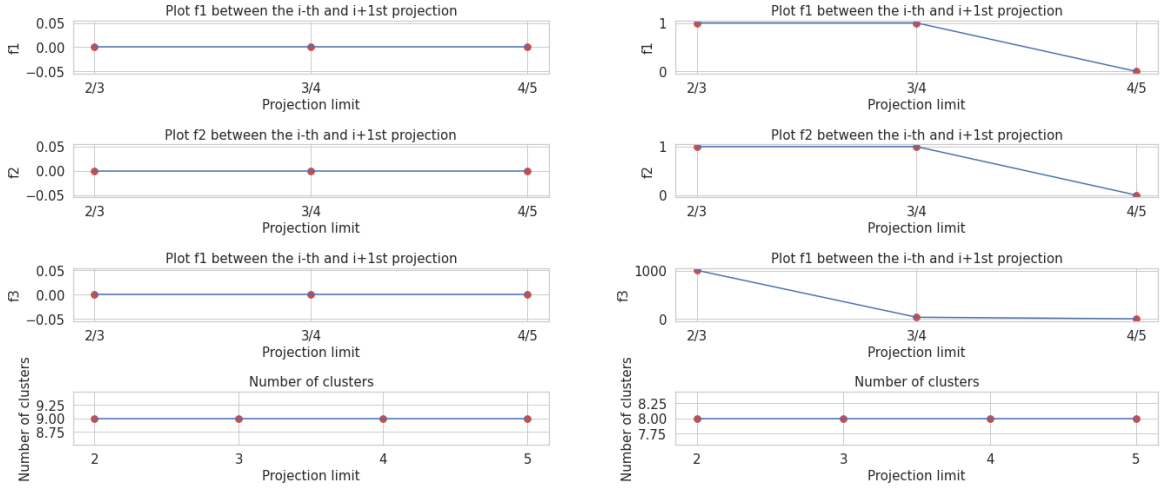
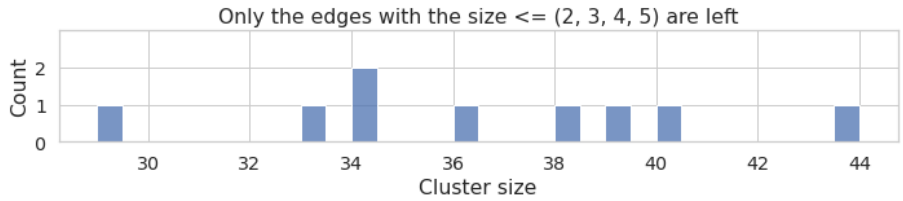


Рис. 4: Кластеризация методом HSBM Кластеризация методом проекций ($\alpha = 1$)



Посмотрим на таблицу формирований кластеров (от 1-й проекции ко 2-й) и взаимосвязь найденных и истинных кластеров:

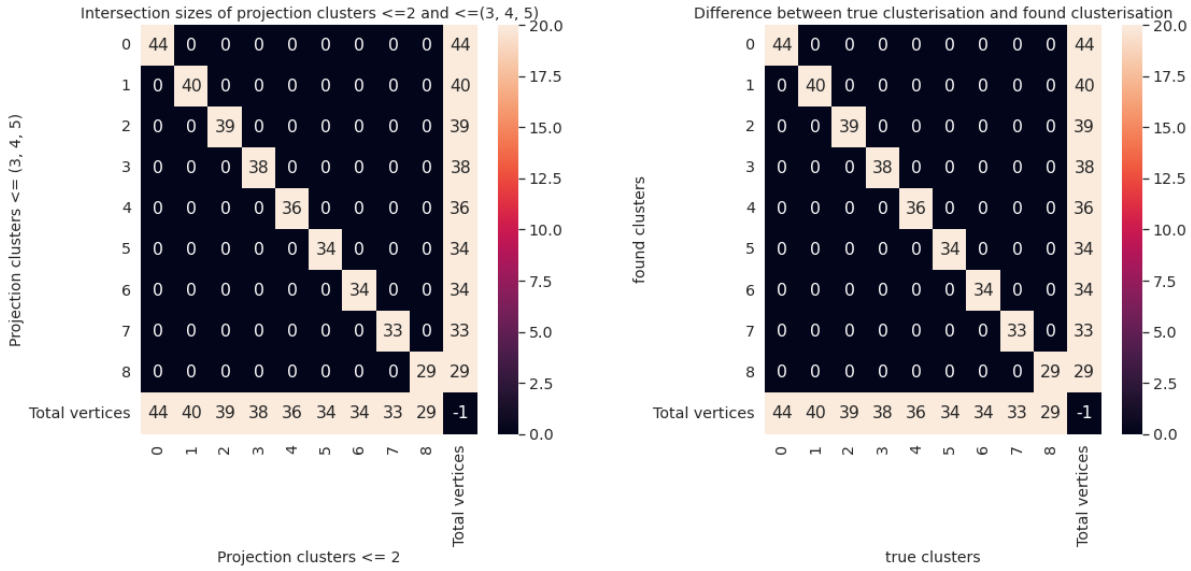


Рис. 5: Пересечения кластеров: 2-3 проекций истинных и найденных ($\alpha = 1$)

Как видим, уже при 2-проекции кластеризация стабилизировалась. И новые кластеры совпадают со старыми. Также найденные кластеры полностью совпадают с истинными.

Распределение размеров кластеров, если кластеризовать Лувенским алгоритмом, используя взвешенные проекции:

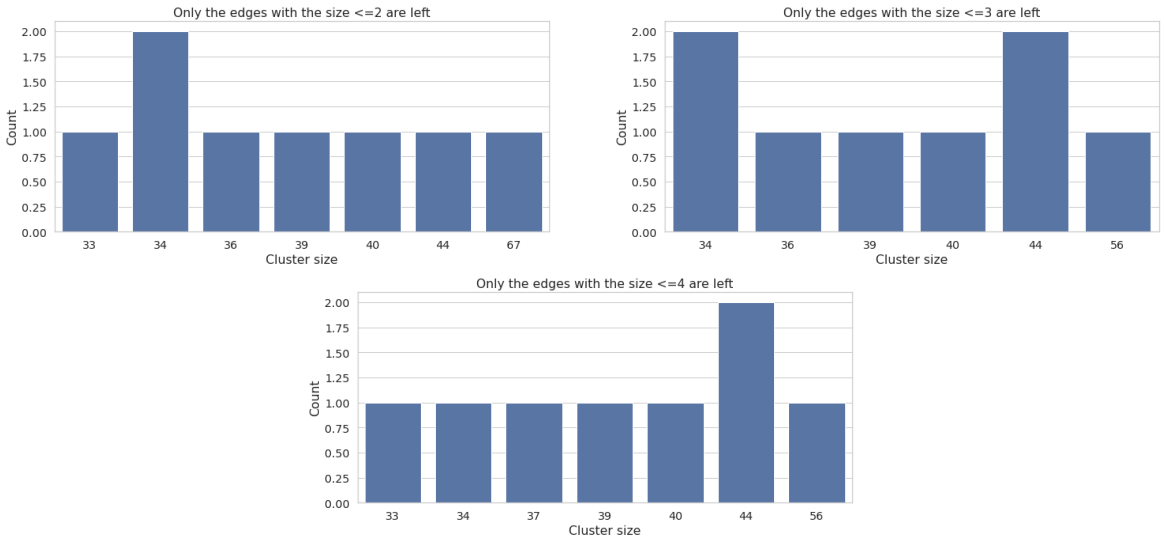


Рис. 6: Распределение размеров кластеров при различных проекциях.

Теперь посмотрим, как формируются кластеры 2-й проекции из кластеров 1-й проекции:

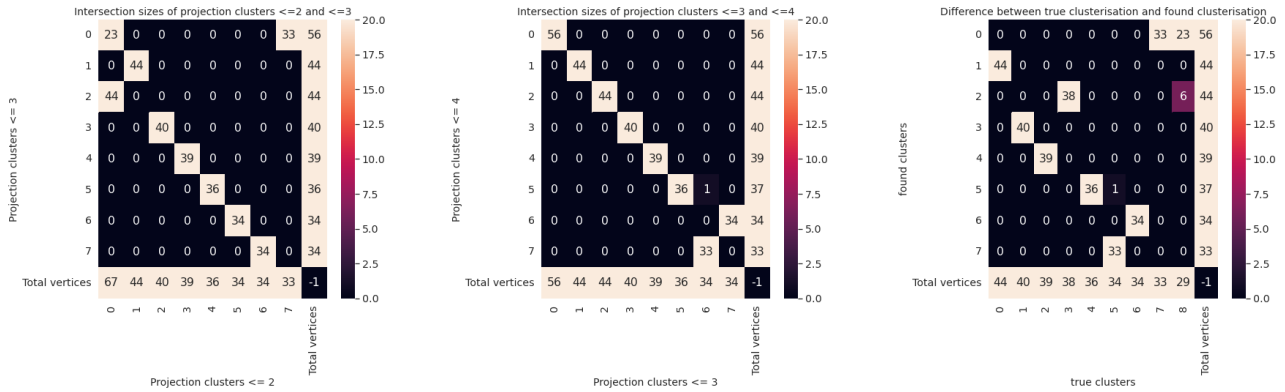


Рис. 7: формирование кластеров: 2-3 проекции (слева) 3-4 проекции (центр), истинные и найденные (справа).

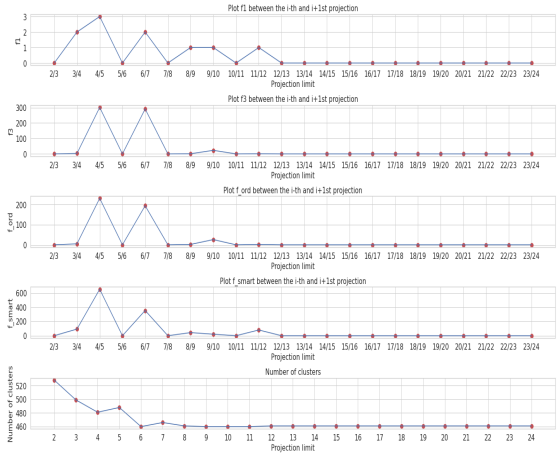
Как видим, на 2-3 проекции дробится 1 кластер на 2 части.
На 3-4 проекции только 1 вершинка переходит к другому кластеру.
И в целом кластеры хорошо восстановлены, но только два целевых объединены в один.

4.3 NDC-classes

Характеристики гигантской компоненты:

1. Вершин - 628
2. Ребер - 816
3. Средняя степень - 9
4. Средний размер ребра: 6.9
5. Размеры ребер: [2 - 24]

Как видим в первом случае, даже на поздних стадиях наблюдаются скачки. Скорее всего это связано с тем, что вообще говоря кластеризации почти не происходит.



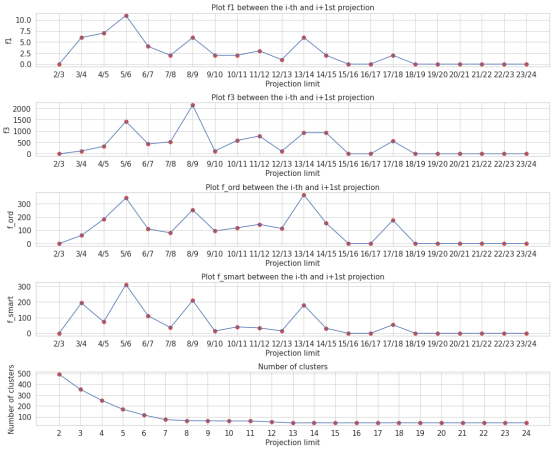


Рис. 8: Кластеризация методом HSBM Кластеризация методом проекций (формула $\alpha = 1$)

Теперь посмотрим на распределение размеров кластеров при кластеризации HSBM:

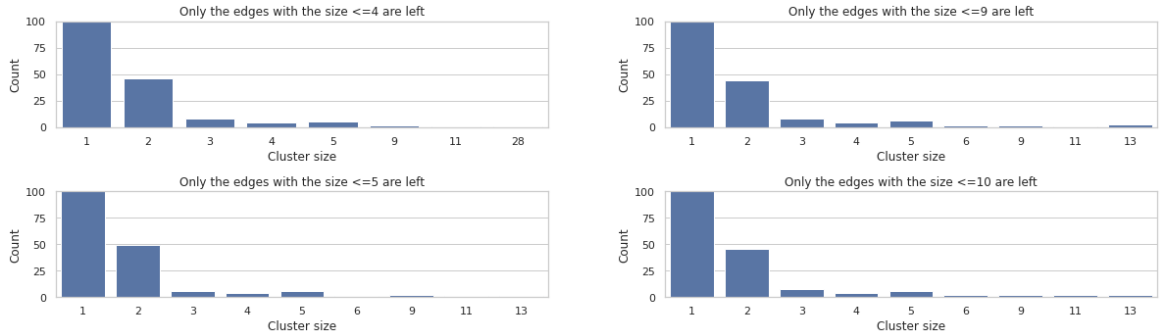


Рис. 9: Распределение размеров кластеров.

Как видим, увеличение функций f_1, f_2, f_3 связано с тем, что большой кластер распадается на более мелкие.

Теперь посмотрим на распределение кластеров при кластеризации методом проекций. Отобразим только пары графиков, в когорых достигается максимум функций f.

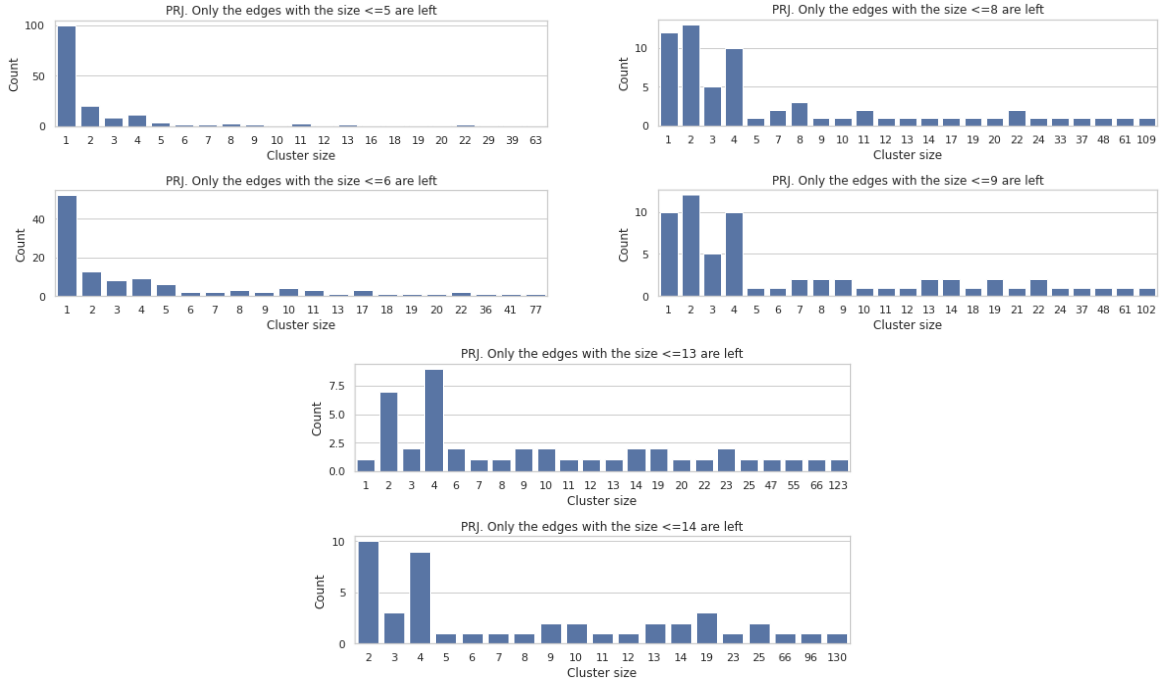


Рис. 10: Распределение размеров кластеров.

Посмотрим, как формируются кластеры в случае кластеризации проекциями.

Рассмотрим только те этапы формирования, при которых наблюдаются большие скачки функций потерь (5-6 и 8-9).

Можно заметить, что сумма в строке (столбце) не всегда совпадает с размером кластера, так как вершины могли уйти (прийти) в этот кластер из более мелких, информация о которых не отражается на графике.

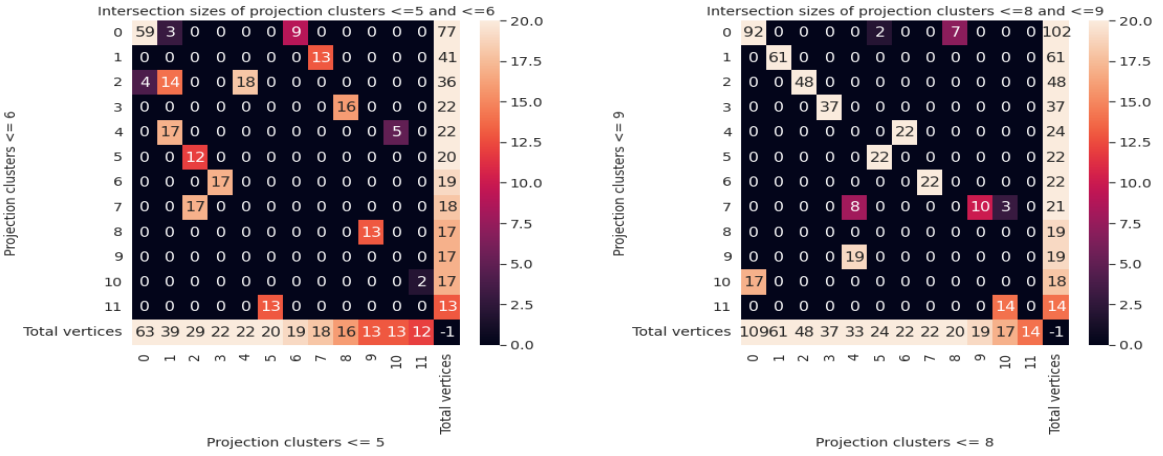


Рис. 11: 5 - 6 проекция 8-9 проекция

Как видим, на 5-6 проекции присутствует среднее дробление: кластер 63 дробится на 4 и 59, что нас устраивает, но кластера размера 39 и 29 дробятся почти на равные части, кластеры размера 20,19,13, 12 теряют больше половины своих вершин (эти вершины попадают в маленькие кластеры).

При 8-9 проекции дробление чуть меньше: четко выделяются 4 самых больших кластера и кластеры размером 24, 22, 22, которые почти не дробятся.

Но, например, кластер размером 33, 20, 19, 17, 14 (с небольшими размерами) раздробились сильно: потеряли более половины своих вершин.

Видим еще одну особенность: большие кластеры в основном или не дробятся, или обмениваются вершинами только с другими большими кластерами.

4.4 House Committees

Характеристики[11]:

1. Вершин - 1290
2. Ребер - 340
3. Средняя степень - 9.2
4. Средний размер ребра: 35.2
5. Размеры ребер: [2-80]
6. Число кластеров: 2

Как видим, примерно до 18/19 проекции функции потерь небольшие, а потом начинают расти и остаются высокими до конца.

Также видим, что функция f_{ord} даже спустя половину проекций продолжает расти, а я f_{smat} стабилизируется. Это говорит о том, что в конце происходят небольшие изменения.

Формирование кластеров:

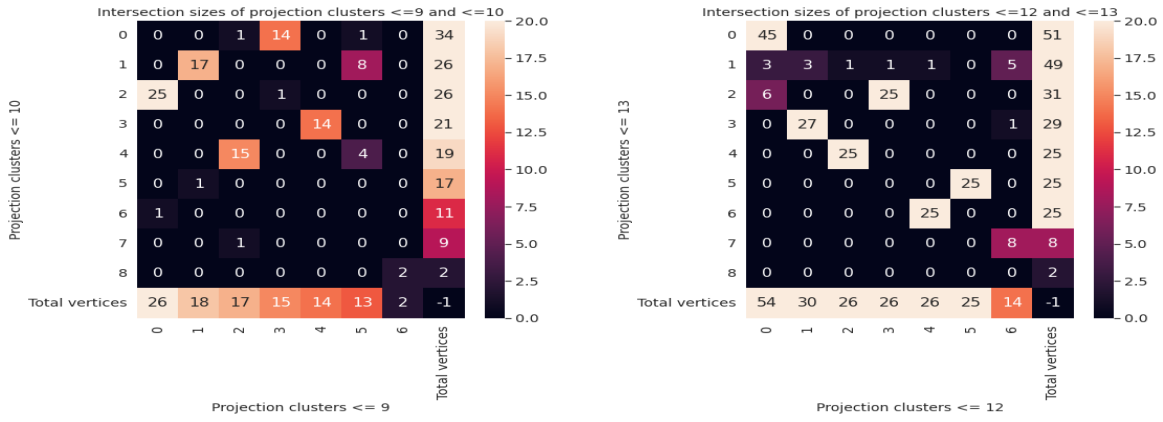


Рис. 12: 9-10 проекция 12-13 проекция

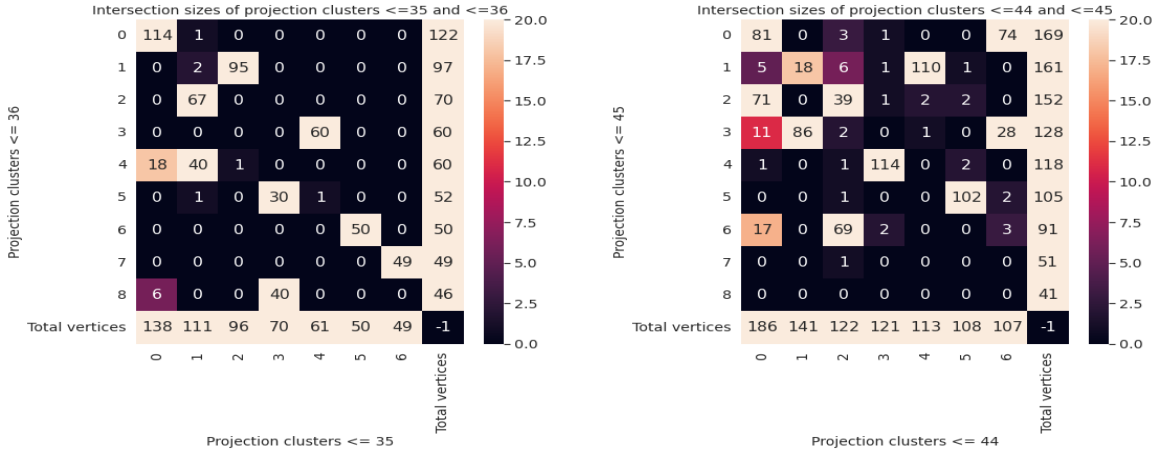


Рис. 13: 35-36 проекция 44-45 проекция

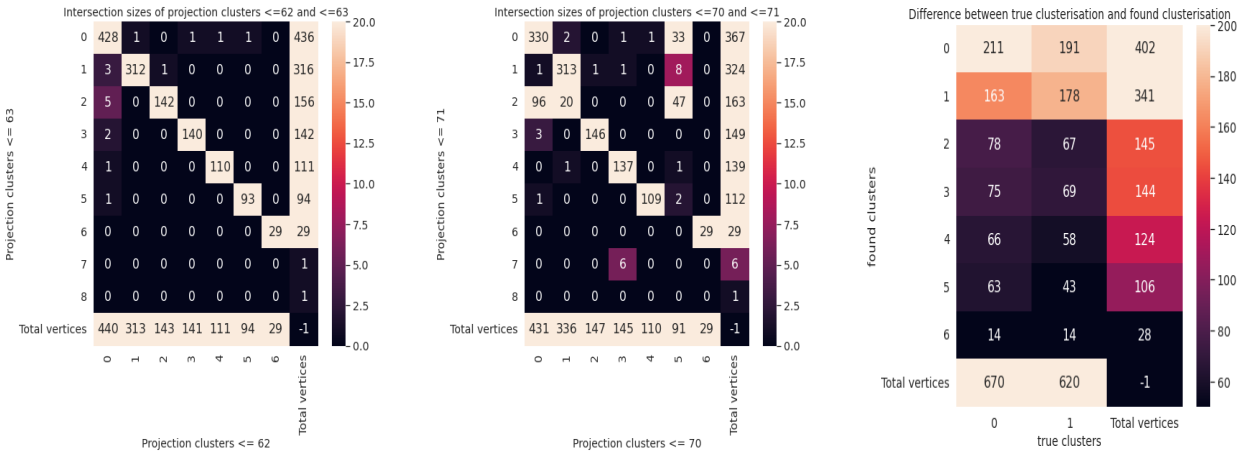


Рис. 14: 62-63 проекция(слева), 70-71 проекция(центр) истинные и найденные (справа).

Как видно, с самого начала выделяются 6-10 крупных кластеров (все остальные по 1 или 2 вершин).

Также в 44-45 проекции видно сильное дробление (кластеры размером 186, 141, 122, 107 раздробились на 2 большие половины).

Далее (на 62-63) мы видим стабилизацию: ничего не дробится, кластеры сохраняются.

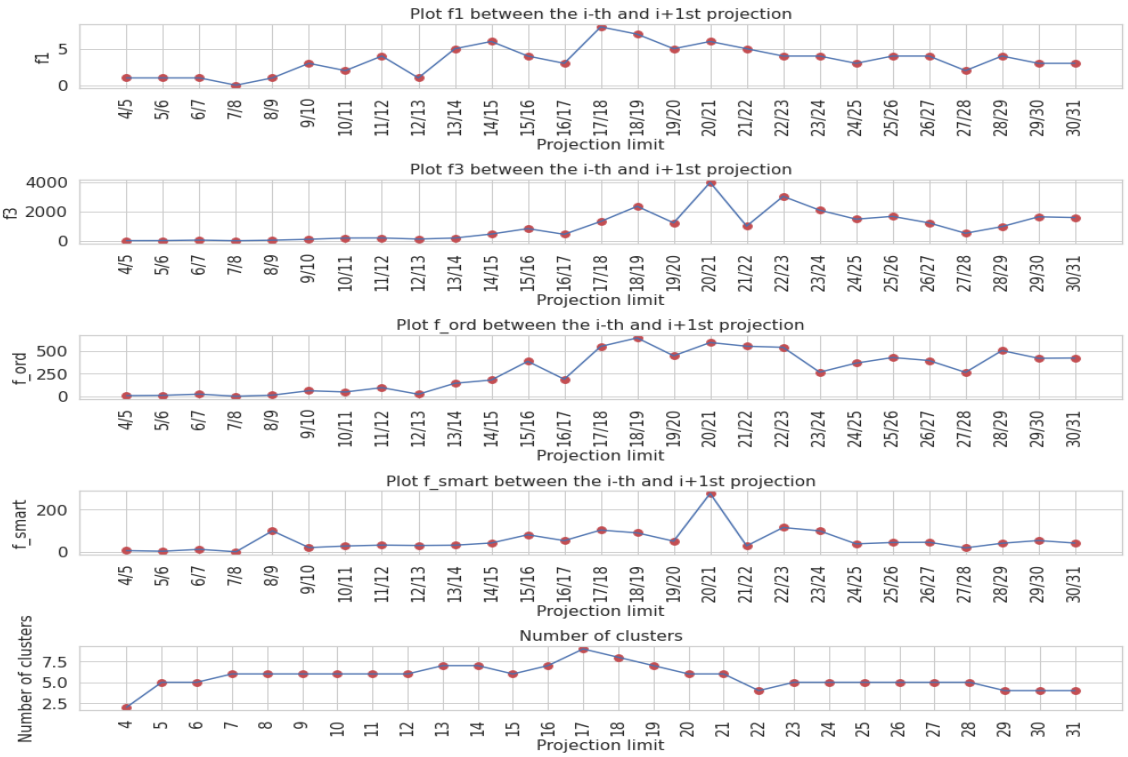
А в конце: на 70-71 мы видим значительное дробление самого большого кластера. Видимо это происходит потому, что целевое количество кластеров - 2, а алгоритм находит 7.

На правом нижнем графике мы можем видеть причину дробления: почему-то все найденные нами кластеры делятся пополам истинными кластерами. То есть найденная кластеризация совершенно не совпадает с истинной.

4.5 Senate Committees

Характеристики[11]:

1. Вершин - 282
2. Ребер - 315
3. Средняя степень - 19
4. Средний размер ребра: 17.5
5. Размеры ребер: [2 - 31]
6. Число кластеров: 2



Как видим, у функции f_{smart} есть один скачок: на 20/21 проекции. Так что в целом можно сказать, что на более поздних итерациях кластеры не дробятся.

Формирование кластеров:

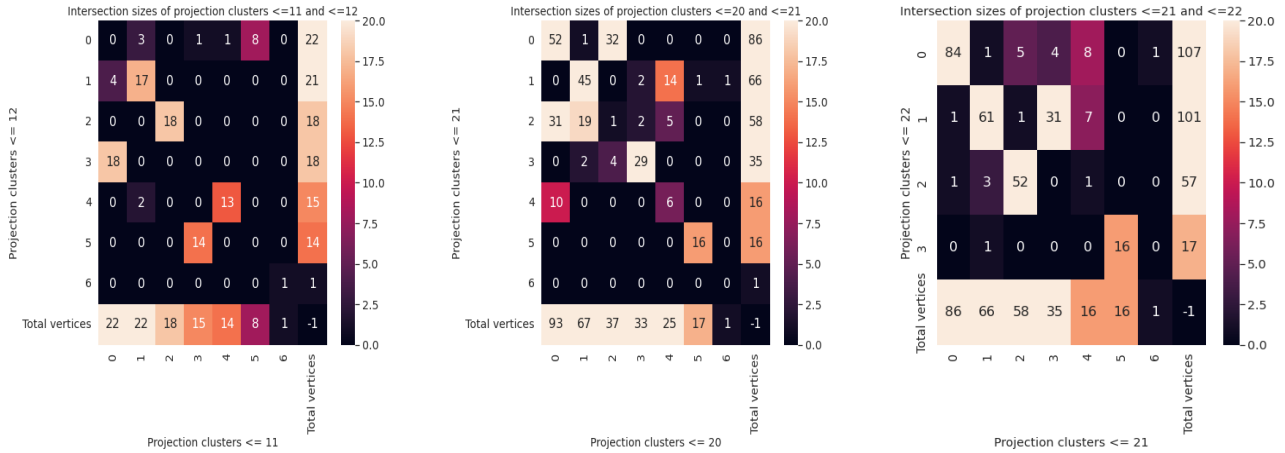


Рис. 16: 11-12, 20-21 и 21-22 проекции.

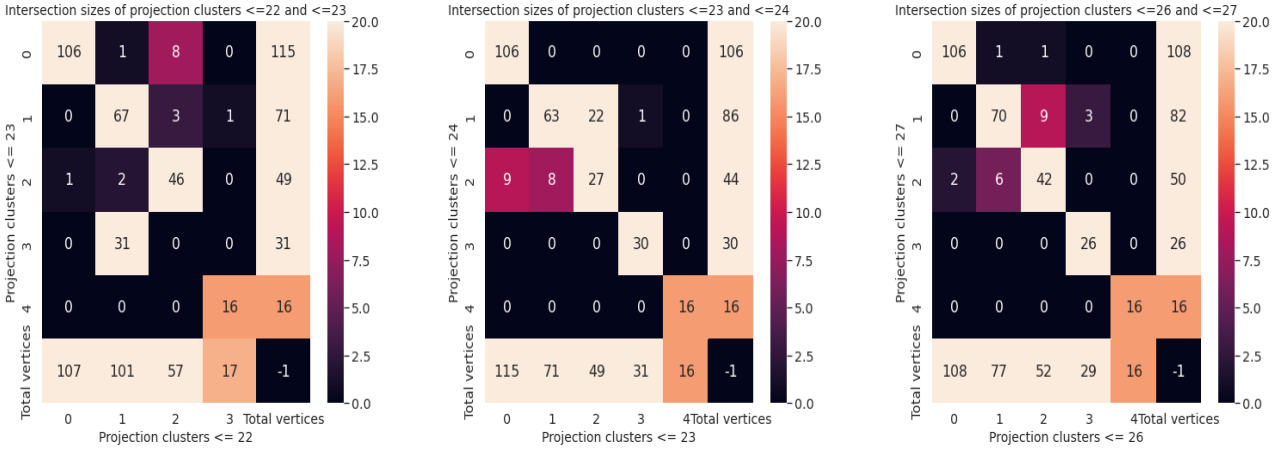
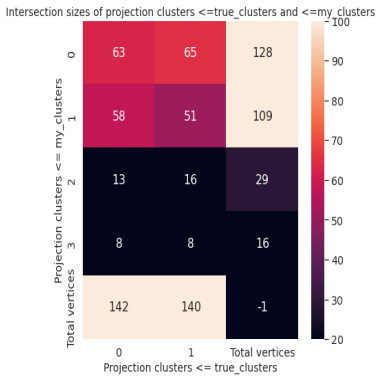


Рис. 17: 22-23, 23-24 и 24-25 проекции.



Как видим, сильное дробление кластера происходит даже на поздней стадии (22-23 проекция): кластер размера 101 дробится на 31 и 67. Но далее существуют как самостоятельные (к ним ничего не прибавляется). Так что такое дробления связано с тем, что целевое количество кластеров - 2, а алгоритм находит 4.

Нижний график - это взаимосвязь истинных кластеров с найденными. Как видим, найденные нами кластеры делятся истинными почти наполовину. То есть найденная кластеризация - совершенно не соответствует действительности.

5 Дальнейшие направления исследования.

1. Смотреть на формирование кластеров не между соседними проекциями, а через 1 проекцию (и через 1 вычислять ошибку).

2. Warmstart для поэтапной кластеризации: начинать кластеризовать с уже имеющейся предыдущей кластеризации.

3. Проводить кластеризацию проекциями пользуясь разными функциями-проекторами (изменять параметр α и смотреть, как будет меняться кластеризация).

4. Исследовать вопросы устойчивости кластеризации. (например, если алгоритм кластеризации закончился, а я перемещу 1, 2 ... или d вершинок и потом запущу алгоритм снова. Будет ли он работать (начнет ли перемещать вершины) или нет?

5. Настроить написанный алгоритм кластеризации гиперграфа, пользуясь статьей [11]. Подобрать параметры β, γ и сравнить получившуюся кластеризацию с представленной в статье.

6. Изучить двойственный к данному гиперграф. Понять, какая связь между кластеризацией двойственного и текущего.

6 Вывод.

Гипотеза о том, что кластеры «слипаются» от ограничения к ограничению оказалась частично подтверждена (первыми 3 графами). На последних 2 мы видим, что даже на поздних проекциях присутствует очень сильное дробление больших кластеров.

Также алгоритм HSBM дает более «устойчивую» кластеризацию, чем алгоритм кластеризации проекциями. Это говорит о том, что при проекции теряется существенная информация о структуре гиперграфа и что разработка новых алгоритмов кластеризации гиперграфа имеет смысл.

Список литературы

[1] M. Girvan; M. E. J. Newman (2002). «Community structure in social and biological networks». Proc. Natl. Acad. Sci. USA. 99 (12): 7821–7826.

[2] www.cs.cornell.edu

[3] Random Walks on Hypergraphs with Edge-Dependent Vertex Weights

[4] Ahn, Y.-Y.; Bagrow, J.P.; Lehmann, S. (2010). "Link communities reveal multi-scale complexity in networks". Nature. 466 (7307): 761–764.

[5] Blondel, Vincent D; Guillaume, Jean-Loup; Lambiotte, Renaud; Lefebvre, Etienne (9 October 2008). "Fast unfolding of communities in large networks". Journal of Statistical Mechanics: Theory and Experiment. 2008 (10): P10008.

[6] M. E. J. Newman (2004). "Fast algorithm for detecting community structure in networks". Phys. Rev. E. 69 (6): 066133. arXiv:cond-mat/0309508.

[7] Newman, M. E. J. (2006). "Modularity and community structure in networks". Proceedings of the National Academy of Sciences of the United States of America. 103 (23): 8577–8696.

[8] M. E. J. Newman (2004). "Detecting community structure in networks". Eur. Phys. J. B. 38 (2): 321–330. Bibcode:2004EPJB...38..321N. doi:10.1140/epjb/e2004-00124-y.

[9] M. Girvan; M. E. J. Newman (2002). "Community structure in social and biological networks". Proc. Natl. Acad. Sci. USA. 99 (12): 7821–7826

[10] M.E.J.Neman (2006). "Finding community structure in networks using the eigenvectors of matrices".

[11] Hypergraph clustering: from blockmodels to modularity Philip S. Chodrow, Nate Veldt, Austin R. Benson

[12] S. H. Strogatz, D. J. Watts (1998). "Collective dynamics of 'small-world' networks". Nature.

[13] A. Barabasi, E. Bonabeau (2003). "Scale-Free Networks". Scientific American.

[14] Wang, Jianxin et al. “Recent advances in clustering methods for protein interaction networks.” BMC genomics vol. 11 Suppl 3,Suppl 3 S10. 1 Dec. 2010,

[15] github.com/2001092236/Community-detection

[16] В. А. Емеличев, О. И. Мельников, В. И. Сарванов, Р. И. Тышкевич. Глава XI: Гиперграфы // Лекции по теории графов. — М.: Наука, 1990. — С. 298—315. — 384 с. — ISBN 5-02-013992-0.

[17] matplotlib.org

[18] networkx.org

[19] github.com/mattbierbaum/arxiv-public-datasets

[20] www.machinelearningmastery.ru/generating-twitter-ego-networks-detecting-ego-communities

[21] fin-az.ru

[22] towardsdatascience.com/community-detection-algorithms