Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук
Магистерская программа математических методов оптимизации и стохастики

Курсовая работа

Выделение пересекающихся сообществ во взвешенных графах

Выполнил:
студент группы м15МОС
Славнов Константин Анатольевич

Научный руководитель:
к.ф.-м.н.
Панов Максим Евгеньевич

Москва, 2016

Содержание

1 Введение

В данной работе будет рассмотрена задача выделения сообществ — группы вершин в графе, плотно связанных между собой, но не с остальным графом. На текущий момент известно множество подходов и методов для выделения непересекающихся сообществ [1]. Гораздо меньше внимания уделено случаю пересекающихся групп. В данной работе будет предложен новый метод решения задачи в еще более конкретной постановке: на взвешенных графах с пересекающимися группами вершин. Метод основан на алгоритме BigClam [2], который разработан для случая невзвешенных графов. Можно сказать, что новый метод является обобщением модели BigClam для случая взвешенных графов.

Работа начинается с постановки задачи и подробного описания метода BigClam. Особое внимание уделено методу инициализации. Будет показано, как небольшими усилиями можно улучшить оригинальный метод инициализации. Новый подход ускоряет алгоритм, немного уточняет итоговый результат, позволяет сойтись к лучшему значению функции.

Далее речь пойдет о подходах обобщения метода на случай взвешенного графа. Рассматривается самое простое, наивное обобщение BigClam, описываются его недостатки. Предлагается перейти к более сложной модели, которая является центральным результатом данной работы. Заканчивается работа экспериментами на модельных данных и сравнением с другими методами решения задачи.

В заключении сформулированы основные выводы, указаны направления дальнейшего исследования.

2 Постановка задачи

Общий метод базируется на следующем очевидном наблюдении: чем больше сообществ разделяют две вершины, тем больше вероятность, что они будут соединены ребром. Этот факт находит подтверждение на реальных данных [2]. Модели, решающие задачу поиска структуры сообществ, должны учитывать этот факт.

В данной работе задача выделения пересекающихся групп вершин будет рассматриваться как некоторая обобщенная проблема матричного разложения. Ее общность заключается в выборе функционала. Критерий качества будет строится исходя из предположений о природе данных. Такой подход позволяет использовать современные наработки по решению оптимизационных проблем такого типа в задаче поиска структуры сообществ. Опишем его.

Представим, что каждая вершина v графа G взаимодействует с сообществом A с некоторой силой F_{vA}. Нулевая сила означает отсутствие взаимодействия. Такую модель можно представить как двусвязный граф. Разместим вершины исходного графа G в первой компоненте и вершины-сообщества во второй (рис. 1). Отметим, что подобная концепция позволяет отразить идею не только пересекающихся сообществ, но и вложенных.

Определим силу взаимодействия X_{uv} между вершинами u и v, которая будет определять вероятность появления ребра между вершинами.

$$X_{uv} = F_u \cdot F_v^T,$$

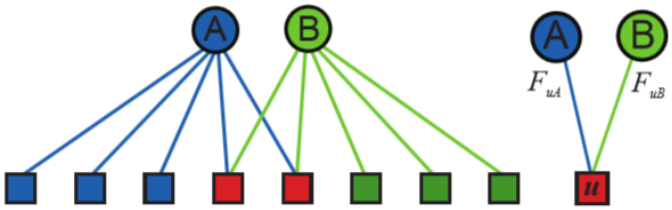

Рис. 1. Двусвязный граф модели BigClam. Сверху вершины-сообщества, снизу вершины исходного графа. Вершины и сообщества взаимодействуют с неотрицательной силой F_{vA}. Ребра, которые соответствуют нулевым весам опущены для большей наглядности.

Обозначение	Описание
$G = (V, E)$	граф
N	количество вершин в графе
$A \in \mathbb{R}_+^{N \times N}$	матрица смежности
$F \in \mathbb{R}_+^{N \times K}$	матрица силы принадлежности к сообществам
K	количество сообществ
C	множество сообществ
$\mathrm{P}(u, v)$	вероятность появления ребра (u, v)
$\mathrm{P}((u, v) \mid c)$	вероятность появления ребра (u, v) при условии, что u и v принадлежат сообществу c
$l(F)$	логарифм функции правдоподобия
$\mathcal{N}(u)$	1-окрестность вершины u (соседи вершины)
w_{uv}	вес ребра (u, v) для взвешенного графа

Таблица 1. Основные обозначения работы.

где F_u — вектор-строка, составленная из F_{uA} — сил взаимодействия вершин с сообществами графа.

Таким образом, получено желаемое свойство: чем больше общих сообществ разделяют вершины, тем сильнее они связаны. Определим вероятность появления ребра (u, v) как $p(u, v) = 1 - \exp(-X_{uv})$. Т.е. чем сильнее связаны вершины, тем вероятнее появление ребра между ними. Таким образом получена вероятностная модель. Предполагается, что наблюдаемый нами граф сгенерирован измено из нее. Позже будет описана вероятностная интерпретация, которая позволит лучше разобраться в описанной модели данных.

Итак, кратко опишем оригинальный метод, его основные предположения.

§2.1 Cluster Affiliation Model for Big Networks (BigClam)

Основные обозначения. Введем основные обозначения, которые будут использоваться на протяжении всей работы в Таблице 1.

Предположения.

1. Каждая вершина $v \in V$ относится к сообществу $c \in C$ с некоторой силой

$$F_{vc} \geq 0.$$

2. Вероятность появления ребра (u, v), при условии, что вершины u, v находятся в одном сообществе c определяется по формуле

$$\mathrm{P}((u, v) \mid c) = 1 - \exp(-F_{uc} \cdot F_{vc}).$$

3. Каждое сообщество c генерирует ребра независимо от других, а значит, что вероятность появления ребра можно посчитать по формуле для независимых случайных величин. Получим

$$\mathrm{P}(u, v) = 1 - \exp(-\sum_{c \in C} F_{uc} F_{vc}) = 1 - \exp(-F_u F_v^T),$$

$$F = \{F_u\} = \{F_{uc}\} \in \mathbb{R}^{N \times K}.$$

Вероятностная интерпретация. Предложенная модель имеет простую вероятностную интерпретацию. Предположим, что существуют скрытые случайные переменные X_{uv}, которые определяют силу взаимодействия вершин, а ребро появляется, если $X_{uv} > 0$. Каждое сообщество графа дает свой независимый вклад $X_{uv}^{(c)}$ в X_{uv}. Предположим, что $X_{uv}^{(c)} \sim \mathrm{Pois}(F_{uc} \cdot F_{vc})$, где $F_{vc} \geq 0$ сила взаимодействия вершины v и сообщества c. Значит, что

$$X_{uv} \sim \mathrm{Pois}(\sum_c F_{uc} \cdot F_{vc}) = \mathrm{Pois}(F_u \cdot F_v^T).$$

Вероятность появления ребра равна

$$p(u, v) = \mathrm{P}(X_{uv} > 0) = 1 - \exp(-F_u F_v^T),$$

что соответствует формулам, полученным выше.

Метод и схема оптимизации. Для восстановления матрицы F предлагается использовать метод максимизации правдоподобия. Из приведенных выше формул не сложно вывести, что правдоподобие $l(F)$ определяется как

$$l(F) = \log(\mathrm{P}(A \mid F)) = \sum_{(u,v) \in E} \log(1 - \exp(-F_u F_v^T)) - \sum_{(u,v) \notin E} F_u F_v^T.$$

Для оптимизации возьмем алгоритм блочного координатного спуска с методом проекции градиента на каждом шаге. Фиксируется значение F_v, оптимизация ведется по F_u, $u \neq v$. Задача становится выпуклой.

$$\forall u: \quad \arg\max_{F_u \geq 0} l(F_u),$$

$$l(F_u) = \sum_{v \in \mathcal{N}(u)} \log(1 - \exp(-F_u F_v^T)) - \sum_{v \notin \mathcal{N}(u)} F_u F_v^T,$$

где $\mathcal{N}(u)$ — соседи вершины u.

$$\nabla l(F_u) = \sum_{v \in \mathcal{N}(u)} F_u \frac{\exp(-F_u F_v^T)}{1 - \exp(-F_u F_v^T)} - \sum_{v \notin \mathcal{N}(u)} F_v^T.$$

Основная сложность формулы (линейная по размеру графа) сконцентрирована во втором слагаемом. Заметим, что

$$\sum_{v \notin \mathcal{N}(u)} F_v^T = \sum_v F_v - F_u - \sum_{v \in \mathcal{N}(u)} F_v.$$

Значение $\sum_v F_v$ легко поддерживать в памяти, обновляя на каждой итерации за константное время. Получаем сложность одной итерации $O(\mathcal{N}(u))$. В этом заключается значимое отличие рассматриваемого метода. Такая сложность позволяет обсчитывать графы с количеством вершин до 10^5 за приемлемое время.

Для подбора градиентного шага используется backtracking line search [3].

Связь с матричными разложениями. Подобная постановка задачи позволяет рассмотреть задачу выделения пересекающихся сообществ как задачу неотрицательного матричного разложения с общим функционалом. То есть, необходимо найти такую низкоранговую матрицу $F \in \mathbb{R}_+^{N \times K}$, что она наилучшим образом приближает значение A в смысле некоторого функционала:

$$F = \arg\min_{F \geq 0} D(A, f(FF^T)).$$

В качестве меры ошибки между матрицей и ее аппроксимацией выступает функция $D(\cdot, f(\cdot))$. В нашем случае $D = -l(F)$ — значение правдоподобия, а $f(x) = 1 - \exp(-x)$ — функция, которая преобразует силы взаимодействия вершин в вероятности появления ребра (link function). Эта часть функционала делает его более пригодным к анализу бинарных матриц, чем стандартная l_2-норма.

Поэтому в качестве оптимизационной схемы берется стандартный метод для решения задач матричного разложения.

Восстановление структуры сообществ. После того, как метод оптимизации сошелся к некоторому оптимальному значению матрицы, осталось перейти к задаче выделения групп вершин по F. Для того, чтобы восстановить исходную структуру сообществ C, сравним значение матрицы F с некоторым порогом δ. Если $F_{vc} > \delta$, то $v \in c$. δ выберем следующим образом.

Обозначим за ε вероятность появления ребра в графе (если бы все ребра появлялись равномерно):

$$\varepsilon = \frac{2|V|}{|E| \cdot (|E| - 1)}.$$

Возьмем δ так, чтобы две вершины принадлежали одному сообществу, если модельная вероятность появления ребра между ними выше чем ε:

$$\varepsilon \leq 1 - \exp(-\delta^2).$$

А значит

$$\delta = \sqrt{-\log(1-\varepsilon)}.$$

ε-сообщество. Если две вершины не разделяют хотя бы одного общего сообщества, между ними не может быть ребра. Очевидно, что в настоящих сетях такого свойства нет. Поэтому введем так называемое ε-сообщество. Предполагаем, что все вершины относятся к единому ε-сообществу с малой силой δ, определенной выше. То есть дополнительно предполагается, что ребро могло возникнуть случайно с вероятностью, которая равна доле существующих ребрер в графе:

$$\mathrm{P}((u,v) \mid \varepsilon) = \frac{2|V|}{|E| \cdot (|E|-1)}.$$

3 Инициализация

В предыдущем разделе был полностью описан метод BigClam зи исключением способа инициализации. Так как задача не является выпуклой, большое внимание необходимо уделить этому аспекту. В ходе анализа было замечено, что метод подбора начального приближения матрицы F можно усовершенствовать. Все результаты и наблюдения будут подробно описаны в этой части работы. Начнем с описания оригинального метода инициализации из статьи [4].

§3.1 Оригинальный подход

Введем метрику на подмножестве множества вершин $S \subset V$.

$$\varphi(S) = \frac{\mathrm{cut}(S)}{\min(\mathrm{vol}(S), \mathrm{vol}(\bar{S}))},$$

где $\mathrm{cut}(S)$ — разрез подмножества S, $\mathrm{vol}(S)$ — его объем:

$$\mathrm{cut}(S) = \mathrm{cut}(S, \bar{S}) = \sum_{\substack{(v,u)\in E \\ v\in S\, u\in\bar{S}}} a_{vu},$$

$$\mathrm{vol}(S) = \sum_{\substack{(v,u)\in E, \\ v\in S}} a_{vu}.$$

Величина $\varphi(S)$ называется проводимостью (Conductance) и очень похожа на взвешенный разрез. Утверждается, что эго-графы (вершина с ее 1-окрестностью), которые достигают локального значения функционала $\varphi(S)$ являются хорошими сообществами и могут использоваться в качестве инициализации для других методов. Локальность понимается в смысле локальности на графе. То есть значение проводимости эго-графа любой соседней вершины должно быть больше, чем в данной.

В качестве инициализации предлагается выбрать необходимое количество эго-графов, которые достигают локального минимума. Если таких графов больше, чем требуется, выберем те, у которых минимальное значение проводимости. То есть для

$$S_1 \equiv \mathcal{N}(v_1), \ldots, S_K \equiv \mathcal{N}(v_K) \,:\, \varphi(S_1) \leq \cdots \leq \varphi(S_K),$$

$v_i \in V$ — вершины локального минимума: $\forall u \in \mathcal{N}(v_i) \ : \ \varphi(\mathcal{N}(u)) > \varphi(\mathcal{N}(v_i))$,

$$F_{ij} = \begin{cases} 1, \text{если } v_i \in S_j; \\ 0, \text{иначе.} \end{cases}$$

Если S_i меньше необходимого числа, заполним остальные столбцы матрицы случайным образом.

Недостатки. Опишем минусы такого подхода. Матрица F получается детерминированной, а значит нельзя перезапускать метод для поиска лучшего результата. F состоит всего из двух значений 0 и 1, при чем подавляющее большинство нули. Ноль является плохой точкой для старта, т.к. $F = 0$ — точка локального минимума. То есть модель потенциально может сойтись к плохому значению функционала. Помимо этого, часто так получается, что 2 или более значений среди $S_1, \ldots, S_K$ соответствуют одному сообществу. Продемонстрируем последний недостаток на следующем модельном примере.

Рассмотрим матрицу $F \in \mathbb{R}_+^{3\times 140}$ как на рис. 2. Сгенерируем из нее модельный граф согласно модели BigClam. Найдем 3 вершины, эго-графы которых будут образовывать начальное приближение матрицы F. На рис. 3 изображен семплированный граф. На левой части красным цветом отмечены 3 найденные вершины.

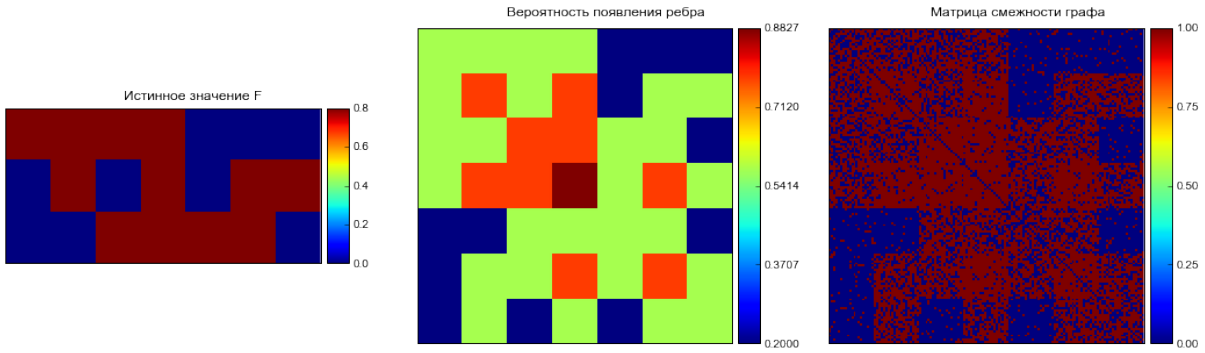

Рис. 2. Модельный пример графа из трех пересекающихся сообществ. Всего в графе 140 вершин. **Слева** матрица F размера 3 на 140; **по середине** матрица $1 - \exp(FF^T)$ — определяет вероятность появления ребра согласно модели BigClam; **справа** случайная семплированная матрица смежности.

Видно, что 2 вершины являются представителями одного и того же сообщества. И это не единичный случай, такая ситуация часто встречалась в том числе на реальных графах.

Формально, можно сказать, что часто находятся такие $i, j \in 1, \ldots, K$, что ${F_{\bullet i}}^T F_{\bullet j} \gg 0$, где $F_{\bullet i} = \{F_{ji}\}_{j=1}^N$, а значит эго-графы вершин значительно пересекаются. Вершины лежат рядом и велика вероятность того, что принадлежат одному и тому же сообществу.

§3.2 Новый подход

Решением такой проблемы может послужить дополнительная регуляризация за близкое расположение к уже взятым в качестве инициализации вершинам. То есть

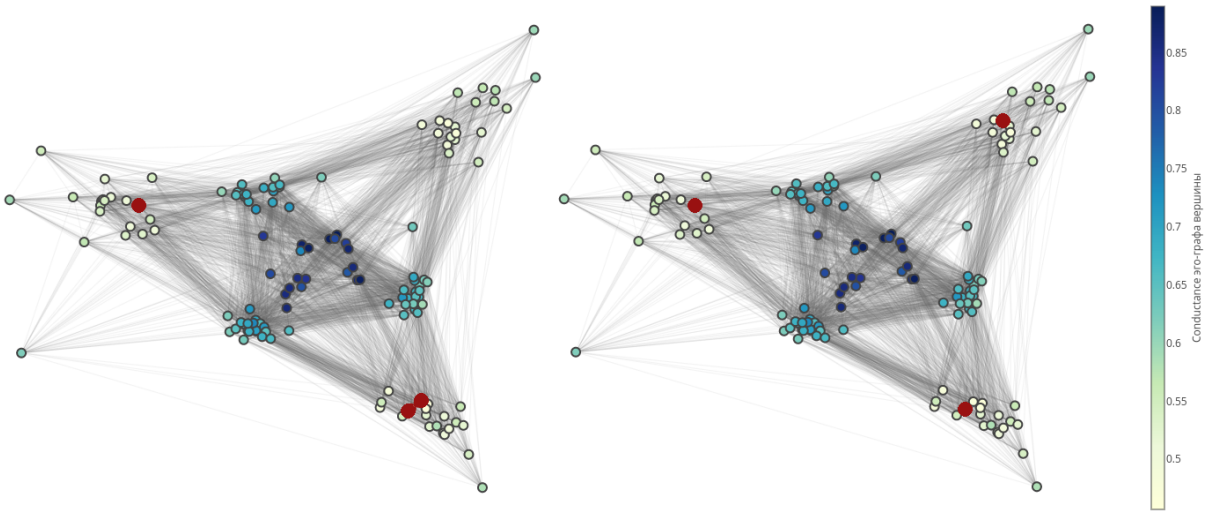

Рис. 3. Граф, сгенерированный из матрицы F на рис. 2. Красным цветом обозначены вершины, эго-графы которых образуют начальное приближение матрицы F. **Слева** оригинальный метод. 3 вершины с минимальной проводимостью, которые достигают локального минимума в окрестности вершины. Две вершины попали в один кластер, что портит начальное приближение. **Справа** новый метод. 3 вершины с минимальной регуляризованной проводимостью, которые достигают локального минимума в окрестности вершины. Все 3 вершины лежат в своих кластерах, что улучшает качество инициализации.

при инициализации следующего вектор-столбца $F_{\bullet j}$ матрицы F к значению проводимости S_i эго-графа i вершины добавим штраф, равный

$$R = \gamma \cdot {F_{\text{selected}}}^T F_{\text{ego}_i},$$

где $F_{\text{selected},l} = 1$ в вершинах, которые уже входят в инициализацию ($\exists k : F_{lk} = 1$), а $F_{\text{ego}_i,l} = 1$, если l лежит в эго-графе i вершины ($l \in \mathcal{N}(i)$), $\gamma = 1/\sum_l F_{\text{selected},l}$ — нормировка. То есть R — доля уже выбранных вершин, которые попали в рассматриваемый эго-граф.

Результат работы метода на текущем примере отражен на правой части рис. 3. Желаемый результат достигнут. Для большей общности можно было бы ввести коэффициент регуляризации, но было решено этого не делать и учитывать 2 критерия (проводимость и коррелированность) с одинаковым весом.

Таким образом получилось избавится от одной описанной выше проблемы. Проблема нулей и детерминированности решается простым добавлением равномерного шума в диапазоне $[0; 0,1]$. Константа $0,1$ подобрана экспериментально.

Дополнительно возникла идея, что вершины соседние к эго-графу имеют большую вероятность принадлежать к тому же сообществу, чем остальные. Значит вокруг найденного начального приближения можно “распространить” его на соседние вершины. То есть дать половину веса (0.5) всем вершинам, соседним к найденному эго-графу.

Обозначение	Описание
rand	Инициализация равномерным шумом от 0.75 до 1.25
cond	Инициализация в локальных максимумах проводимости (стандартный метод)
cond_new	Новый метод со штрафом за пересечение с уже выбранными вершинами
cond_randz	Дополнительно заменяем нули из метода *cond* на значения от 0 до 0.1
cond_new_randz	Дополнительно заменяем нули из метода *cond_new* на значения от 0 до 0.1
cond_randz_spr	Применяем метод *cond*. Соседние с найденными эго-графами вершины получают половину его веса. Затем заменяем нули матрицы F на значения от 0 до 0.1
cond_new_randz_spr	Применяем метод *cond_new*. Соседние с найденными сообществами вершины получают половину его веса. Затем заменяем нули матрицы F на значения от 0 до 0.1

Таблица 2. Расшифровка обозначений для методов инициализации.

§3.3 Эксперименты

По итогам исследования появилось целое семейство методов инициализации. В ходе экспериментов было изучено поведение правдоподобия на модельных и реальных данных. Т.к. инициализация не основной предмет работы, эксперименты проводились буквально на нескольких примерах. Более детальное изучение предложенных методов предстоит изучить в дальнейшей работе.

Модельный пример был описан ранее, реальные данные представляют собой набор из 4 небольших эго-графов знакомых автора работы из социальной сети.

В Таблице 2 приведено описание исследуемых методов инициализации. На рис. 4 и 5 приведено поведение правдоподобия в зависимости от номера итерации для модельных и реальных данных соотвественно.

§3.4 Выводы

По итогам тестов можно сказать, что предложенные методы инициализации демонстрируют лучший результат. Начальные приближения оказались лучше по правдоподобию, как следствие, метод сходится быстрее и к немного лучшим значениям функционала. Если на модельных примерах преимущество было не столь значительным, на реальных данных оно очевидно. Похоже, что старая инициализация приводит к плохим локальным максимумам.

Отметим, что все примеры имели достаточно небольшой размер и маленький диаметр графа, поэтому методы с приставкой *_spr* проявили себя незначительно, т.к. почти все вершины оказывались соседними к выбранным для инициализации эго-

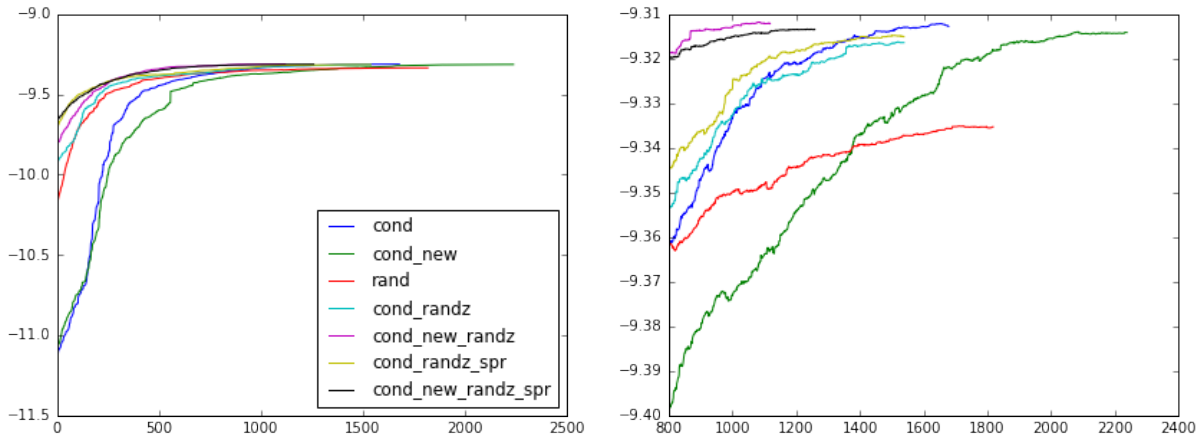

Рис. 4. Поведение минус логарифма от минус логарифма правдоподобия (ось y) в зависимости от номера итерации (ось x) для различных начальных приближений из таблицы выше. **Слева** общий план. **Справа** увеличенный участок с 800 итерации. Лучше всех себя показывают новые методы *cond_new_randz* и *cond_new_randz_spr*.

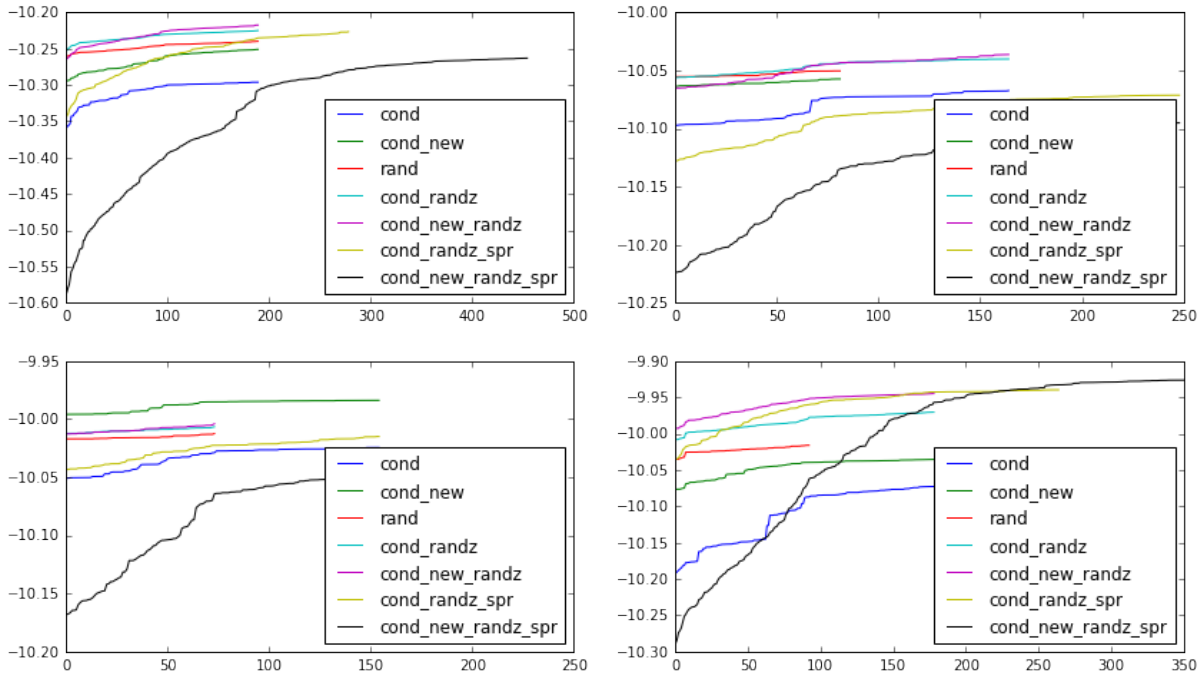

Рис. 5. Поведение минус логарифма от минус логарифма правдоподобия (ось y) в зависимости от номера итерации (ось x) для различных начальных приближений для четырех реальных графов (4 графика). Показано завершение оптимизации и 0 по горизонтальной оси соответствует 250 итерации. Во всех четырех случаях лидируют новые методы, только в одном метод без предложенной регуляризации.

графам. В дальнейшей работе эксперименты будут повторно проведены с большими графами.

4 Новые модели для взвешенных графов

Перейдем к основной теме данной работы: выделение пересекающихся сообществ в взвешенном графе. Такие методы полезны тем, что позволят учесть дополнительную информацию, которая часто дополняет матрицу смежности, но не может напрямую использоваться в методах, работающих с бинарной матрицей смежности, как BigClam.

Начнем с самого простого и интуитивного обобщения метода BigClam, который будем называть наивный взвешенный BigClam. Вес ребра (u, v) обозначим за w_{uv}. Для обработки взвешенных ребер изменим функционал качества следующим образом:

$$l(F) = \sum_{(u,v)\in E} \log\left(1 - \exp\left(-\frac{F_u F_v^T}{w_{uv}}\right)\right) - \sum_{(u,v)\notin E} F_u F_v^T.$$

Изменилось только первое слагаемое — силу взаимодействия вершин $F_u F_v^T$ нормализовали на вес ребра. Тем самым получается, что чем больше вес w_{uv}, тем выше должно быть значение сил F_u и F_v, которые его объясняют, а значит вероятность того, что вершины лежат в одном сообществе увеличивается.

То есть исходное предположение, что вероятность появления ребра (u, v) , при условии, что вершины u, v находятся в одном сообществе c определяется как

$$\mathrm{P}((u, v) \mid c) = 1 - \exp\left(-F_{uc} F_{vc}\right),$$

заменяется на предположение, что вероятность появления ребра (u, v) *с весом* w_{uv}, при условии, что вершины u, v находятся в одном сообществе c есть

$$\mathrm{P}((u, v) \mid c, w_{uv}) = 1 - \exp\left(-\frac{F_{uc} F_{vc}}{w_{uv}}\right).$$

Такой подход имеет множество недостатков. Функционал теряет симметричность оригинальной модели: наличие или отсутствие ребра вносят разный вклад в функционал. Нельзя предложить похожей красивой вероятностной интерпретации для сил взаимодействия вершин типа

$$X_{uv}^{(c)} \sim \mathrm{Pois}\left(\frac{F_{uc} \cdot F_{vc}}{w_{uv}}\right),$$

так как в случае отсутствия ребра параметр распределения будет бесконечен.

Если рассматривать 2 случая: наличия или отсутствия ребра, получается, что в модели необходимо изначально задать какие ребра присутствуют, а какие нет, иначе из такой модели нельзя будет сгенерировать рассматриваемый граф. Дополнительно в модель изначально заложены ожидаемые веса на ребрах. Все эти наблюдения крайне обременительны.

Несмотря на перечисленные недостатки, эксперименты подтверждают работоспособность модели, и она имеет право на существование. Описанные сложности в интерпретации такой модели мотивируют искать другие способы работы с взвешенными данными.

§4.1 Гамма Модель

Попробуем построить новую модель по типу BigClam. В ходе его анализа было замечено, что все определяется распределением на скрытых переменных X_{uv}. В случае BigClam они распределены по Пуассону. Если выпало что угодно кроме нуля — в графе есть ребро.

Такая модель легко обобщается на случай целочисленных весов. Если пренебречь последним утверждением, получим пуассоновкую модель на ребрах графа. Для случая непрерывных весов такая модель не подойдет. Значит необходимо подобрать непрерывный аналог распределения Пуассона.

Самое главное свойство, которые использовались при выводе оригинального метода — аддитивность распределения. То есть сумма независимых случайных величин из этого распределения не должна выводить за его пределы.

В качестве непрерывного аналога распределения Пуассона рассмотрим гамма распределение. Их сумма (с одинаковым коэффициентом масштаба) не выводит из заданного класса.

Обозначим за $\Gamma(k, \theta)$ гамма распределение. Аналогично BigClam, выпишем базовые предположения, которые используются в гамма модели.

Предположения.

1. Вероятность появления ребра с весом w_{uv}^c, при условии, что вершины принадлежат сообществу c определяется как

$$\mathrm{P}\left(w_{uv}^c \mid c\right) \sim \Gamma\left(k = F_u F_v^T + 1, \theta = 1\right).$$

2. Каждое сообщество c генерирует ребра независимо друг от друга, а значит, что вес ребра в графе

$$w_{uv} = \sum_c w_{uv}^c \sim \Gamma\left(\sum_c F_{uc} F_{vc} + 1, 1\right) = \Gamma\left(F_u F_v^T + 1, 1\right)$$

Поясним почему беремся именно $F_u F_v^T + 1$, а не $F_u F_v^T$. При $F_u F_v^T = 0$ вероятность появления ребра между вершинами должна быть минимальной. $\Gamma(0, 1)$ является экспоненциальным распределением. Если в графе не существует сообществ, именно это распределение будет объяснять возникающие ребра между вершинами графа, которое является самым естественным для социальных графов.

Для большей наглядности дальнейших рассуждений будем опускать параметр θ и обозначим за $K_{uv} = F_u F_v^T + 1$. Выведем формулу правдоподобия данной модели.

Модель.

$$\begin{aligned} l(F) &= \log\left(\mathrm{P}\left(A \mid F\right)\right) = \sum_{w_{uv}} \log \mathrm{P}(w_{uv}) \\ &= \sum_{w_{uv}} \left[-\log\Gamma\left(K_{uv}\right) - K_{uv}\log\theta + \left(K_{uv}-1\right)\cdot\log w_{uv} - \frac{w_{uv}}{\theta}\right] = \left[\theta = 1\right] \\ &= \sum_{w_{uv}} \left[-\log\Gamma\left(K_{uv}\right) + \left(K_{uv}-1\right)\cdot\log w_{uv} - w_{uv}\right] \\ &= \sum_{w_{uv}} \left[-\log\Gamma\left(F_u F_v^T + 1\right) + F_u F_v^T \cdot \log w_{uv} - w_{uv}\right] \to \max_{F\geq 0}. \end{aligned}$$

Схема оптимизации. Схема оптимизации используется та же самая, что и в BigCLAM.

Для того, чтобы посчитать градиент, понадобится дигамма функция:

$$\Psi(x) = \frac{\mathrm{d}}{\mathrm{d}x}\log\left(\Gamma(x)\right).$$

Тогда градиент можем записать как

$$\frac{\mathrm{d}l(F)}{\mathrm{d}F_u} = -\sum_v F_v \Psi\left(F_u F_v^T + 1\right) - F_v \log w_{uv}.$$

Ко всем весам прибавляется небольшое ε, чтобы избежать нулевых значений под логарифмом.

По сравнению с оригинальным методом, из-за того, что сумма взвешенная, провести такой же прием с упрощением сложности вычисления градиента не получится. Для каждого шага, для каждого F_u придется пересчитывать сумму целиком. Получается линейная сложность.

Вычисление значения правдоподобия $l(F_u)$, также линейно, поэтому для подбора шара нецелесообразно использовать backtraking. Используется обычный убывающий шаг.

Эксперименты. В ходе экспериментов было рассмотрено 2 варианта моделирования матрицы смежности взвешенного графа. 1 модель была взята прямо из предположений, описанных выше: задается матрица F, веса генерируются из гамма распределения. На таких данных оптимизационная схема надежно работает и сходится из любого, даже случайного приближения.

Однако, такая модель данных не соответствует реальным графам, т.к. в настоящие социальные графы разреженные. Вторая модель данных учитывала этот факт. Сначала генерировалась структура графа (есть ребро или нет), затем, только для проявившихся ребер генерируется его вес. Ниже приведен алгоритм генерации:

1. Задается матрица F и параметр $\gamma \geq 0$.
2. $\forall\, u \in V, v \in V$ с вероятностью $1 - \exp(-\gamma F_u {F_v}^T)$ в графе создается ребро $(u, v) \in E$.

3. $\forall (u,v) \in E$ — созданных ребер генерируется вес $w_{uv} \sim \Gamma\left(\sum_c F_{uc}F_{vc} + 1, 1\right)$.

Появился дополнительный параметр модели γ. Чем меньше его значение, тем более разреженной является результирующая матрица смежности A.

Анализ. Оказалось, что предложенная гамма модель не может объяснить большое количество нулей в подобного рода данных. Оптимизация не приводит ни к какому адекватному результату даже из хороших начальных приближений (близких к истинному F). Необходимо дополнительно учитывать возникающие в данных нули.

По этим причинам было решено отказаться от дальнейшего рассмотрения этой модели и перейти к следующей. Единственное, что стоит отметить, что в данных с малым количеством нулей или полным их отсутствием такой подход может оправдать себя. Например, для решения задачи кластеризации.

§4.2 Разреженная гамма модель

Предположения. Все предположения о природе данных возьмем из описанной в предыдущем параграфе процедуры генерации данных. Собственно, именно эта генеративная модель и привела к созданию данного метода.

Модель. Обозначим вес ребра за w_{uv}, а бинаризованные элементы матрицы смежности за a_{uv}. То есть $a_{uv} = \mathbb{I}[w_{uv} \neq 0]$. Заметим, что вес ребра w_{uv} отличен от 0 только если $a_{uv} \neq 0$, а значит, что

$$\mathrm{P}(w_{uv} = 0 \mid a_{uv} = 0) = 1.$$

Теперь, с учетом замечания, выведем формулу логарифма правдоподобия, воспользовавшись формулой полной вероятности.

$$\begin{aligned}
l(F) &= \sum_{\forall (u,v)} \log \mathrm{P}(w_{uv}) = \sum_{u,v} \log \mathrm{P}(w_{uv} \mid a_{uv}) + \log \mathrm{P}(a_{uv}) \\
&= \sum_{(u,v)\in E} \log \mathrm{P}(w_{uv} \mid a_{uv} = 1) + \log \mathrm{P}(a_{uv} = 1) + \\
&\qquad + \sum_{(u,v)\notin E} \log \mathrm{P}(w_{uv} = 0 \mid a_{uv} = 0) + \log \mathrm{P}(a_{uv} = 0) \\
&= \sum_{(u,v)\in E} \log \mathrm{P}(w_{uv} \mid a_{uv} = 1) + \log \mathrm{P}(a_{uv} = 1) + \sum_{(u,v)\notin E} \log \mathrm{P}(a_{uv} = 0) \\
&= \sum_{(u,v)\in E} \log \mathrm{P}_{\Gamma}(w_{uv}) + \sum_{(u,v)\in E} \log\left(1 - \exp\left(-\gamma F_u {F_v}^T\right)\right) - \gamma \sum_{(u,v)\notin E} F_u {F_v}^T.
\end{aligned}$$

Первое слагаемое — это правдоподобие предыдущей гамма модели на ребрах с ненулевыми весами, а последние 2 слагаемых это оригинальная BigClam модель для матрицы $\sqrt{\gamma}F$!

Значит, получившаяся модель является их комбинацией. Она сохраняет все преимущества BigClam-модели, в том числе скорость вычисления производной, но при этом учитывает взвешенные ребра и имеет дополнительный параметр γ, который связывает матрицы для гамма и оригинальной модели.

Схема оптимизации. Поскольку модель является комбинацией двух других, для вычисления градиента необходимо просто сложить градиенты из исходных методов. Отметим только, что в гамма модели сумму необходимо взять не по всем вершинам, а только по соседним к u.

Используется схема оптимизации с убывающим шагом, так как backtracking приводил модель к плохим значениям и оказался неэффективен.

Эксперименты. На аналогичных модельных экспериментах, на которых простая гамма модель не сумела восстановить матрицу F, был запущен новый метод. Новый метод без труда восстанавливает истинное значение матрицы F, даже если параметр γ задано не точно.

Разреженная гамма модель является главным результатом текущей работы. Метод без существенных затрат обобщает оригинальный BigClam на случай взвешенного графа и имеет под собой понятные и простые вероятностные предположения.

5 Эксперименты

§5.1 Функционалы качества

Оценивать качество получаемых результатов будем по трем функционалам: модулярности, нормализованной общей информации (Normalized Mutal Information (MNI) и среднему значению проводимости (Conductance) сообществ. Первые две меры изначально рассматриваются в случае непересекающихся сообществ, поэтому необходимо взять некоторые обобщения предложенных функционалов.

Опишем их подробнее.

Модулярность. Для начала рассмотрим случай непересекающихся сообществ. Функционал был предложен Ньюманом и Гирваном в ходе разработки алгоритмов кластеризации вершин графа [5].

Модулярность — это скалярная величина из отрезка $[-1, 1]$, которая определяет качество разбиения на “модули”:

$$Q = \frac{1}{2m} \sum_{i,j} \left(a_{ij} - \frac{d_i d_j}{2m} \right) \delta(c_i, c_j),$$

где A — Матрица смежности графа, a_{ij} — (i, j) элемент матрицы, d_i — степень i вершины графа, c_i — метка вершины (номер сообщества, к которому относится вершина), m — общее количество ребер в графе. $\delta(c_i, c_j)$ — дельта-функция: равна единице, если $c_i = c_j$, иначе нулю.

Для взвешенных графов, под a_{ij} понимается вес ребра соединяющий вершины i и j, а $m = \frac{1}{2} \sum a_{ij}$.

Модулярность по сути сравнивает предложенное разбиение со случайным. Ее значение равно разности между долей ребер внутри сообщества и ожидаемой долей связей, если бы ребра размещены случайно.

Главным недостатком функционала называют проблему разрешающей способности (функционал плохо работает с маленькими сообществами) [6].

Для случая пересекающихся сообществ воспользуемся одним из обобщений, предложенных в [7]. Возьмем вместо дельта-функции степень принадлежности вершины к сообществу и дополнительно просуммируем по всем сообществам.

$$Q = \frac{1}{2m} \sum_{c \in c} \sum_{i,j} \left(a_{ij} - \frac{d_i d_j}{2m} \right) \beta_{ic} \beta_{jc} \,,$$

$$\text{где} \quad \forall i : \quad \sum_{c} \beta_{ic} = 1.$$

В случае с рассматриваемыми моделями значение β_{ic} легко получить путем нормализации матрицы F по столбцам.

Conductance. Conductance или проводимость была описана в разделе про инициализацию. Для того, чтобы использовать предложенную метрику в роли функции качества разбиения, нужно перейти от оценки одного сообщества ко всему разбиению. Будем брать среднюю и максимальную проводимость по всем сообществам. Для того, чтобы использовать значение как метрику качества, вычтем из единицы ее значение.

Normalized Mutal Information. Опишем метрику в случае непересекающихся сообществ. Утверждается, что если одно разбиение похоже на другое, то необходимо малое количество информации, чтобы восстановить одно из другого. Значит, значение такой величины можно интерпретировать как меру сходства данных разбиений.

Рассмотрим два разбиения $\{x_i\}$ и $\{y_i\}$, где i это номер вершины, а x_i и y_i метки сообществ из разбиений. Предполагается, что метки x и y являются значениями двух случайных величин X и Y, которые имеют совместное распределение

$$\mathrm{P}(x, y) = \mathrm{P}(X = x, Y = y) = \frac{n_{xy}}{n},$$

где n — общее количество вершин, n_{xy} - количество вершин, которым в разбиениях $\{x_i\}$ и $\{y_i\}$ сопоставлены метки x и y. Аналогично

$$\mathrm{P}(X = x) = \frac{n_x}{n}, \quad \mathrm{P}(Y = y) = \frac{n_y}{n}.$$

Тогда общая информация определяется как

$$\mathrm{I}(X, Y) = \mathrm{H}(X) - \mathrm{H}(X \mid Y),$$

где $\mathrm{H}(X)$ — энтропия распределения X, а $\mathrm{H}(X \mid Y)$ — условная энтропия:

$$\mathrm{H}(X) = -\sum_{x} \mathrm{P}(x) \log \mathrm{P}(x),$$
$$\mathrm{H}(X \mid Y) = -\sum_{x,y} \mathrm{P}(x, y) \log \mathrm{P}(x \mid y).$$

Параметр	Значение	Описание
N	1000; 1500	Количество вершин
μ_t	0,1; 0,3	Величина смешивания (нечеткость сообществ)
$k_{\max}$	50; –	Максимальная степень вершины
k	30; –	Средняя степень вершины
μ_ω	0,1; 0,3	Сила смешивания весов на ребрах
γ	от 0 до 0,7	Доля вершин в пересекающихся частях сообществ
ξ	2	Параметр распределения на весах
τ_1	2	Параметр распределения на степенях вершин
τ_2	2	Параметр распределения на размерах сообществ
o_m	2	Количество сообществ, в которые входит одна вершина при их наложении

Таблица 3. Значения параметров в модельных данных. Прочерк означает, что на параметр не было никаких ограничений.

В качестве меры сходства используют нормализованную общую информацию (normalized mutual information):

$$\mathrm{I}_{norm} = \frac{2\mathrm{I}(X,Y)}{\mathrm{H}(X)+\mathrm{H}(Y)}.$$

Величина I_{norm} равна единице, когда разбиения совпадают, и нулю, если они независимы.

В случае с пересекающимися сообществами NMI определяется аналогично. В работе используется обобщение предложенное в [8].

§5.2 Данные

Большинство стандартных наборов данных, на которых тестируют алгоритмы, не подходят для тестирования предложенных методов. Либо нет истинной пересекающейся структуры сообществ, либо граф не взвешенный. Поэтому основные тесты будут проведены на модельном наборе данных.

Модель данных предложена в работе [9]. В работе используется код, предоставленный авторами статьи. Модель имеет много параметров. Было выбрано два набора, представленные в Таблице 3. Первые значения были выбраны такие же, как в работе [10]. Вторые выбраны произвольно.

Значение параметра γ варьируется от 0 до 0.7. В 0 сообщества не пересекаются, при $\gamma = 0.7$ сообщества перекрываются на 70%. Для анализа построим графики зависимости описанных функционалов качества от значения γ. Чем выше окажется график, тем лучше работает соответствующий метод.

§5.3 Результаты экспериментов

Так как для данных известен правильный ответ, наибольший интерес представляет метрика NMI, по которой сравниваются истинное разбиение с результатом работы методов. На рис. 6 и 7 представлены результаты на модельных данных с 1000 и 1500 вершинами соответственно. Каждая точка является усреднением 20 запусков. Рассматриваются следующие методы.

1. *SparseGamma* — разреженная гамма модель.
2. *BigClamWeighted* — наивный взвешенный BigClam.
3. *BigClam-orig-zeros* — оригинальный BigClam.
4. *COPRA* — label propagation для пересекающегося случая [11].
5. *NMF* — неотрицательное матричное разложение с квадратичной нормой.

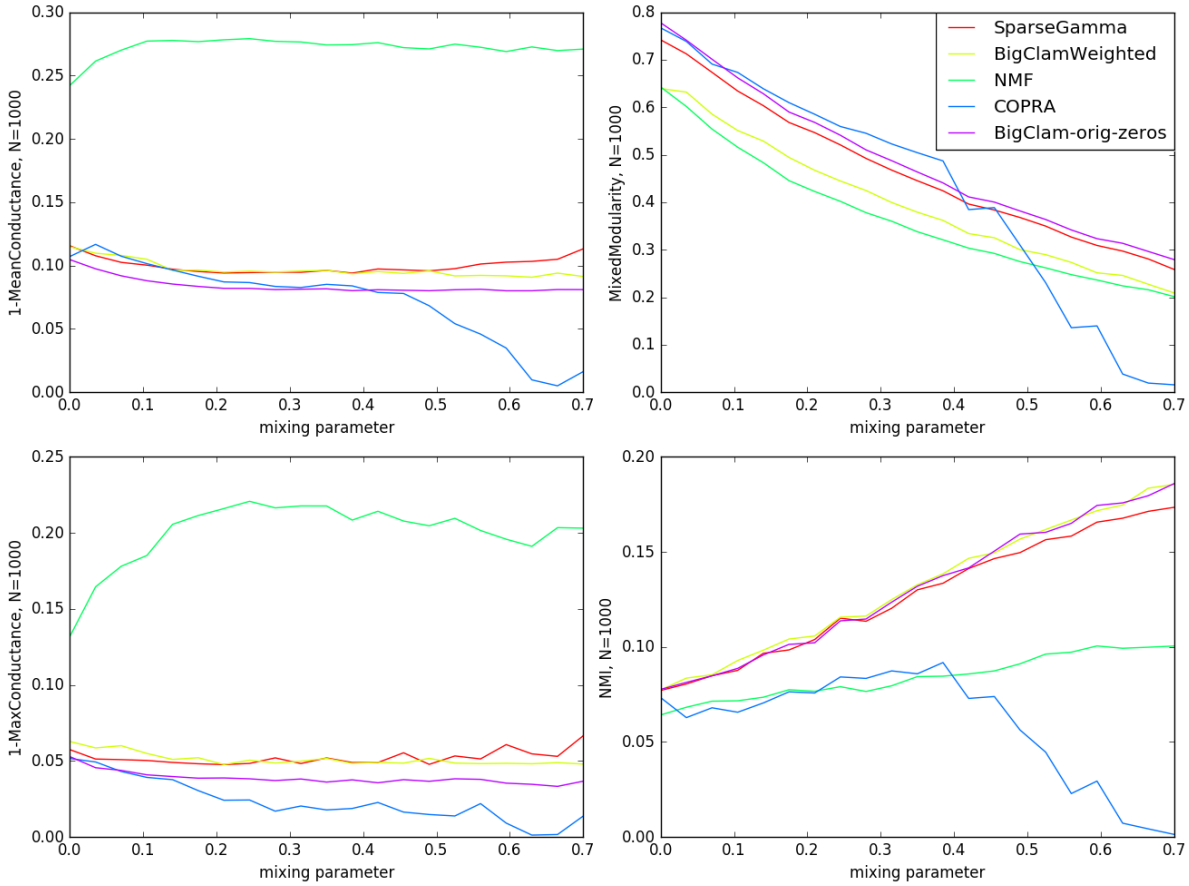

Рис. 6. Результаты работы методов на модельных данных. На графиках изображена зависимость качества разбиения от параметра γ. Граф с 1000 вершинами, остальные параметры указаны в Таблице 3. В левой части сверху единица минус максимальное значение проводимости, снизу единица минус среднее значение проводимости. В правой части сверху модулярность, снизу NMI.

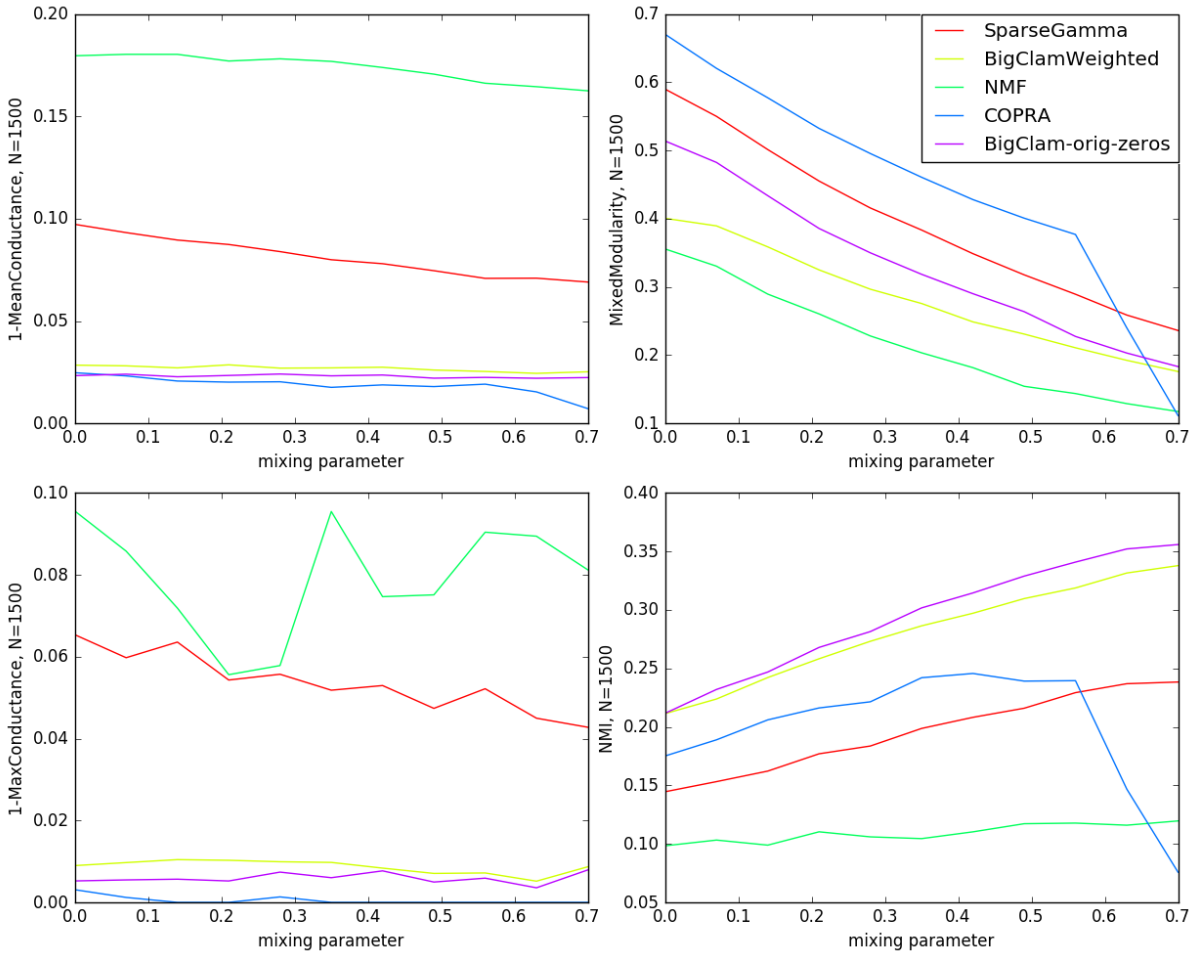

Рис. 7. Результаты работы методов на модельных данных. На графиках изображена зависимость качества разбиения от параметра γ. Граф с 1500 вершинами, остальные параметры указаны в Таблице 3. В левой части сверху единица минус максимальное значение проводимости, снизу единица минус среднее значение проводимости. В правой части сверху модулярность, снизу NMI.

Можно заметить интересную особенность, что такой простой метод как *NMF* в данной задаче имеет подавляющее преимущество по функционалам связанными с проводимостью. При чем, как можно судить по NMI, результирующая структура сообществ получилась хуже чем у любого BigClam метода. Метод *COPRA* с ростом доли вершин в пересекающихся частях сообществ начинает часто в качестве ответа выдавать одно большое сообщество. Это типичная проблема методов типа label propagation.

Интересно, что с ростом параметра γ алгоритмы начинают лучше справляться с поставленной задачей. В случае BigClam методов это можно объяснить. Замечено, что BigClam часто отдает предпочтения структуре сообществ с значительными пересечениями групп вершин между собой. Значит, что чем сильнее пересечение на самом деле, тем проще методу восстановить структуру.

Что касается двух предложенных метов *SparseGamma* и *BigClamWeighted*, то они ведут себя очень похоже с оригинальным методом *BigCLam*. Можно предпо-

ложить, что в данном модельном примере информация, которую с собой несет вес ребра, не достаточно значительна, чтобы при ее удалении ситуация ухудшалась.

6 Результаты работы

Выводы. В работе предложен новый метод для выделения пересекающихся сообществ во взвешенных графах. Алгоритм является обобщением алгоритма BigClam на взвешенный случай. Метод протестирован на двух модельных наборах данных. Полученные результаты говорят о том, что метод работает на уровне современных методов, решающих подобную задачу, но не лучше их.

В ходе работы предложены новые улучшенные способы инициализации на основе значения проводимости, которые позволяют ускорить существующие методы выделения пересекающихся сообществ и достичь лучших значений функционала.

Дальнейшее исследование. Из экспериментов видно, что предложенный метод не использует всю ту информацию, которая дополнительно содержится в весах графа. Подход нуждается в дальнейшей доработке. Также будут проведены дополнительные эксперименты по предложенным методам инициализации.

7 Список литературы

[1] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.
http://www.arxiv.org/abs/0906.0612.

[2] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013.
http://i.stanford.edu/~crucis/pubs/paper-nmfagm.pdf.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] David Gleich and C Seshadhri. Neighborhoods are good communities. *arXiv preprint arXiv:1112.0031*, 2011.
http://arxiv.org/abs/1112.0031.

[5] Mark EJ. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.
http://www.santafe.edu/media/workingpapers/01-12-077.pdf.

[6] Erwan Le Martelot and Chris Hankin. Fast multi-scale detection of relevant communities in large-scale networks. *The Computer Journal*, page bxt002, 2013.
http://comjnl.oxfordjournals.org/content/56/9/1136.full.pdf?keytype=ref&ijkey=Eqs2wpLhn8ZDmvA.

[7] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*, 45(4):43, 2013.
https://arxiv.org/abs/1110.5813.

[8] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.
https://arxiv.org/abs/0802.1218.

[9] Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, 2009.
https://arxiv.org/abs/0904.3940.

[10] Zongqing Lu, Xiao Sun, Yonggang Wen, Guohong Cao, and Thomas La Porta. Algorithms and applications for community detection in weighted networks. *Parallel and Distributed Systems, IEEE Transactions on*, 26(11):2916–2926, 2015.
http://mcn.cse.psu.edu/paper/zongqing/tpds-zongqing15.pdf.

[11] Steve Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10):103018, 2010.
https://arxiv.org/abs/0910.5516.