Выделение пересекающихся сообществ в графах на основе методов факторизации матриц

Константин Славнов
НИУ ВШЭ
kaslavnov@edu.hse.ru

Максим Панов
НИУ ВШЭ, ИППИ РАН
panov.maxim@gmail.com

Аннотация

В данной работе рассмотрена задача выделения сообществ — групп вершин в графе, плотно связанных между собой, но не с остальной частью графа. Известно множество подходов для выделения непересекающихся сообществ [4]. Гораздо меньше внимания уделено случаю пересекающихся групп. В работе предложен новый метод решения задачи на взвешенных графах с пересекающимися группами вершин. Метод является обобщением алгоритма BigClam [16]. В конце работы приведены эксперименты. Сравнение с другими методами решения задачи показали, что предложенные в статье алгоритмы работают на уровне современных аналогов, но не лучше их.

Особое внимание уделено методам инициализации BigClam, предложено несколько улучшений, которые ускоряет алгоритм и позволяет сойтись к лучшему значению функции.

Ключевые слова: Выделение пересекающихся сообществ, социальные графы, матричное разложение.

1. Введение

Сообщества представляют собой основной структурный блок графов реального мира и позволяют понять их структуру и проанализировать свойства. Целый пласт работ в компьютерных науках, статистике, физике и прикладной математике посвящен выделению структуры сообществ в сложных сетях [4]. Сообщество (группа или кластер) интуитивно может быть определено как множество вершин с большим количеством взаимодействий между собой, чем с остальными вершинами графа [5]. Такая группа вершин может быть воспринята как структурная единица социальных сетей, либо функциональная в биохимических графах [8], либо как научная дисциплина в сетях цитирования [2].

Подобная задача в случае непересекающихся сообществ имеет множество подходов и достаточно подробно изучена [4]. Тем не менее, подобные задачи на графах продолжают активно изучаться. В частности, остаются вопросы, как эффективно работать в случае пересекающихся сообществ. Для решения этой задачи уже разработано несколько методов, например, [1], [13], [7]. Однако, проблема остается не до конца изученной. Взвешенные графы представляют дополнительный интерес в этой работе, как более общий случай. Дополнительную сложность составляет отсутствие большого количества данных с правильным известным разбиением на сообщества, из-за чего приходится использовать модельные данные.

Итак, перейдем к описанию общего подхода к поставленной проблеме и формальной постановке задачи.

2. Постановка задачи

Основное предположение, на котором базируется исследование состоит в том, что чем больше сообществ разделяют две вершины, тем больше вероятность, что они будут соединены ребром. Этот факт находит подтверждение на реальных данных [16]. 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 строиться исходя из предположений о природе данных. Такой подход позволяет использовать современные наработки по решению оптимизационных проблем такого типа в задаче поиска структуры сообществ.

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

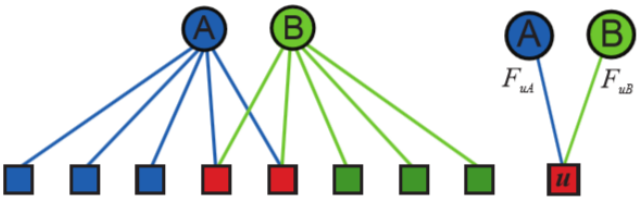

Рис. 1. Двусвязный граф модели BigClam. Сверху вершины-сообщества, снизу вершины исходного графа. Вершины и сообщества взаимодействуют с неотрицательной силой F_{vA}. Ребра, которые соответствуют нулевым весам, опущены для большей наглядности. [16].

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

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

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

Итак, получено желаемое свойство: чем больше общих сообществ разделяют вершины, тем сильнее они связаны. Определим вероятность появления ребра (u,v) как $\mathrm{P}(u,v) = 1 - \exp(-X_{uv})$. Т.е. чем сильнее связаны вершины, тем вероятнее появление ребра между ними. Таким образом, получена вероятностная модель. Предполагается, что наблюдаемый нами граф сгенерирован именно 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)

Введем основные обозначения, которые будут использоваться на протяжении всей работы в Таблице 1. Рассмотрим следующие предположения, которые лежат в основе всей модели.

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

Таблица 1: Основные обозначения работы.

Обозначение	Описание
$G = (V,E)$	граф
N	количество вершин в графе
K	количество сообществ
$A \in \mathbb{R}_+^{N \times N}$	матрица смежности
$F \in \mathbb{R}_+^{N \times K}$	матрица силы принадлежности к сообществам
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

независимых случайных величин. Получим

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

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

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.

Для восстановления матрицы F предлагается использовать метод максимизации правдоподобия. Из приведенных выше формул несложно вывести, что правдоподобие $l(F)$ определяется как

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

Для оптимизации воспользуемся алгоритмом 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 и может быть сформулирована как

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

где $l(F_u) = \sum_{v \in \mathcal{N}(u)} \log(1 - \exp(-F_u F_v^T)) - \sum_{v \notin \mathcal{N}(u)} F_u F_v^T$, а $\mathcal{N}(u)$ — множество соседей вершины u.

Несложно убедиться, что градиент такого функционала может быть вычислен как

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

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].

Опишем связь данной задачи 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 $F \in \mathbb{R}_+^{N \times K}$, которая 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 [11].

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

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

Описанный метод обладает одним незначительным недостатком. 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. Предположим, 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 ребер 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 за исключением способа инициализации. Так как задача не является выпуклой, значительное 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 [6].

3.1. Оригинальный подход

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

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

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 $\phi(S)$ называется проводимостью (conductance) и очень похожа на взвешенный разрез. Утверждается, что эго-графы (вершина с ее 1-окрестностью), которые достигают локального минимума функционала $\phi(S)$, являются хорошими сообществами и могут использоваться в качестве

инициализации для других методов [6]. 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

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

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

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

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

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, из которых подавляющее большинство — нули. Ноль является плохой точкой для старта, так как $F = 0$ — точка локального минимума, и модель потенциально может сойтись к плохому значению функционала. Помимо этого, часто получается так, 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. Видно, что 2 вершины являются

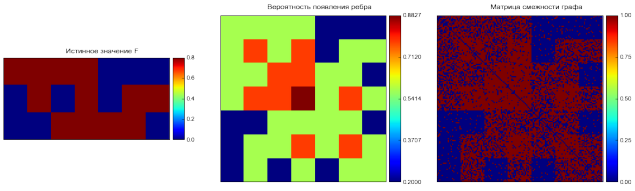

Рис. 2. Модельный пример графа с 140 вершинами и тремя пересекающимися сообществами. Слева матрица F; в центре матрица вероятностей появления ребра $1 - \exp(FF^T)$; справа случайная семплированная матрица смежности.

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

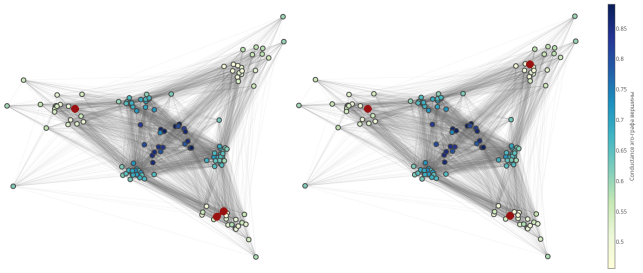

Рис. 3. Граф, сгенерированный из матрицы F на рис. 2. Красным цветом обозначены вершины, эго-графы которых образуют начальное приближение матрицы F. Слева оригинальный метод. Две вершины попали в один кластер, что ухудшает начальное приближение. Справа новый метод. Все 3 вершины лежат в своих кластерах, что улучшает качество инициализации.

значит эго-графы вершин значительно пересекаются. Вершины лежат рядом, и с большой вероятностью принадлежат к одному и тому же сообществу.

3.2. Новый подход

Решением описанной выше проблемы может послужить дополнительная регуляризация за близкое расположение к уже взятым в качестве инициализации вершинам. То есть, при инициализации следующего вектор-столбца $F_{\bullet j}$, добавим штраф к значению проводимости S_i эго-графа i вершины. Штраф положим равным $R = \gamma \cdot F_s{}^T F_{e_i}$, где $F_{s,l} = 1$ в вершинах, которые уже входят в инициализацию ($\exists k : F_{lk} = 1$), а $F_{e_i,l} = 1$, если l лежит в эго-графе i вершины ($l \in \mathcal{N}(i)$), $\gamma = 1/\sum_l F_{s,l}$ — 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 ввести коэффициент регуляризации, но было решено этого не делать и учитывать 2 критерия (проводимость и коррелированность) с одинаковым весом.

Таким образом, получилось избавиться 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 найденное начальное приближение можно “распространить” на соседние вершины. То есть дать половину веса (0.5) всем вершинам, соседним к найденному эго-графу.

Таблица 2: Расшифровка обозначений для методов инициализации.

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

3.3. Эксперименты

По итогам исследования появилось целое семейство методов инициализации. В ходе экспериментов было изучено поведение правдоподобия на модельных и реальных данных. Эксперименты проводились на модели данных, предложенной в [9]. Таблица 3 содержит используемые для генерации параметры. О модели и ее параметрах речь пойдет позже, в секции посвященной основным экспериментам данной работы.

В качестве реальных графов взяты 4 эго-графа (граф знакомств друзей конкретного человека) из социальной сети “Вконтакте”.

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

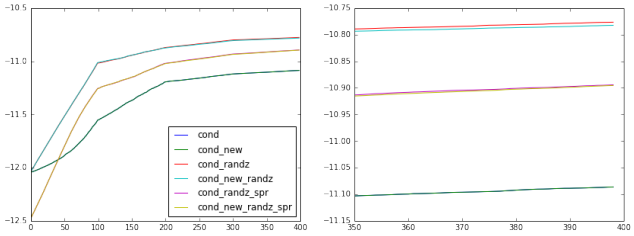

Рис. 4. Поведение двойного логарифма правдоподобия (ось y) в зависимости от номера итерации (ось x) для различных начальных приближений для модельных данных [9] с 1000 вершинами. Было произведено усреднение по 50 графам. Справа показано завершение оптимизации. Видно, что добавление случайного шума значительно улучшает итоговый результат, в то время как метод со штрафом несущественно отличается от оригинального.

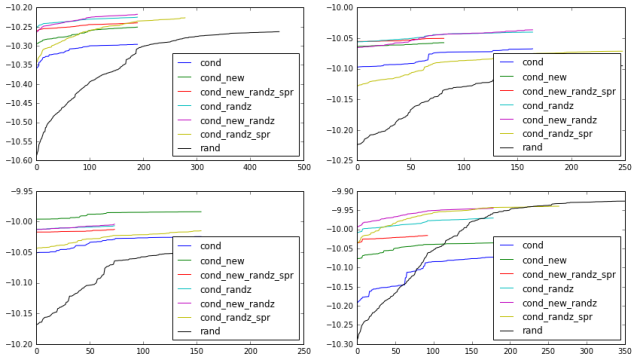

Рис. 5. Поведение двойного логарифма правдоподобия (ось y) в зависимости от номера итерации (ось x) для различных начальных приближений для четырех реальных графов (4 графика). Показано завершение оптимизации и 0 по горизонтальной оси соответствует 250 итерации. В трех случаях лидируют новые методы, в одном лидирует случайная инициализация, но проигрывает по временным затратам.

3.4. Выводы

По итогам экспериментов 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. На модельных данных основное улучшение достигнуто путем добавления равномерного шума, и идея со штрафом не оправдала себя.

На реальных данных преимущества нового подхода с регуляризацией очевидны. Возможно, старая инициализация приводит к плохим локальным максимумам. Идея же с распространением (методы с приставкой *_spr*) не оправдалась и как минимум нуждается в дополнительной проверке.

4. Модели для взвешенных графов

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

По сравнению с оригинальной моделью, изменилась вероятность появления ребра (u,v) *с весом* w_{uv}. Теперь $\mathrm{P}((u,v)$, при условии что вершины u,v находятся в одном сообществе c, определяется как

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

Такой подход имеет множество недостатков. Функционал теряет симметричность оригинальной модели: наличие или отсутствие ребра вносят разный вклад в функционал. Нельзя предложить аналогичной 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, то 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. Такие проблемы 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

Построим новую модель по типу BigClam. В ходе 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, то значит в графе есть ребро.

Такая модель легко обобщается на случай целочисленных весов. Если в качестве веса брать значение X_{uv}, то получится пуассоновская модель на ребрах графа. Для случая непрерывных весов необходимо подобрать непрерывный аналог распределения Пуассона. Рассмотрим гамма распределение $\Gamma(k,\theta)$. Аналогично BigClam, выпишем базовые предположения, которые используются в гамма модели.

1. Вероятность появления ребра с весом w_{uv}^c, если вершины принадлежат сообществу c

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

2. Каждое сообщество c генерирует ребра независимо друг от друга. Тогда 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 берется 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.

В итоге получаем следующую модель.

$$\sum_{w_{uv}} \left[-\log\Gamma\left(F_u F_v^T + 1\right) + F_u F_v^T \cdot \log w_{uv} - w_{uv}\right] \to \max_{F\geq 0}.$$

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

Для того, чтобы посчитать градиент, введем дигамму функцию: $\Psi(x) = \dfrac{\mathrm{d}}{\mathrm{d}x}\log\left(\Gamma(x)\right)$. 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.

В отличие от оригинального метода, провести такой же прием с упрощением сложности вычисления градиента не получится из-за того, что сумма взвешенная. Для каждого шага, для каждого F_u придется пересчитывать сумму целиком, а значит это займет линейное время.

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

Эта модель легла в основу разреженной гамма модели, о которой речь пойдет ниже, а от дальнейшего рассмотрения этой модели было решено отказаться по следующим соображениям. 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, так как 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 $w_{uv} \sim \Gamma(\sum_c F_{uc} F_{vc} + 1, 1)$.

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

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). Стоит отметить, что в данных с малым количеством нулей или полным их отсутствием такой подход может оправдать себя. Например, для решения задачи кластеризации. А в данной ситуации необходимо дополнительно учитывать возникающие в данных нули. Поэтому была разработана следующая модель.

4.2. Разреженная гамма модель

Итак, 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. Возьмем их из описанной в предыдущем параграфе процедуры генерации данных и опишем новую 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:

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

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.

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 оригинальная схема оптимизации.

5. Эксперименты

5.1. Функционалы качества

Оценивать качество получаемых результатов будем по трем функционалам: модулярности (*MixedModularity*), нормализованной общей информации (*Normalized Mutal Information (MNI)* и среднему значению проводимости (*1-MeanConductance*) сообществ. Первые две меры изначально рассматриваются в случае непересекающихся сообществ, поэтому необходимо взять некоторые обобщения предложенных функционалов. Для модулярности используется обобщение, предложенное в [15], обобщение нормализованной общей информации можно найти в [10].

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.

Таблица 3: Значения параметров в модельных данных.

Параметр	Значение	Описание
N	1000; 5000	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 0.5	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

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. Параметры были выбраны тем же способов, что и в работе [12]. А именно выбрано 4 набора данных: два по 1000 вершин и два по 5000. В каждой из них были данные с выраженными сообществами ($\mu_t = \mu_\omega = 0.1$) и менее выраженными ($\mu_t = \mu_\omega = 0.3$).

Значение параметра γ варьируется от 0 до 0.5. В 0 сообщества не пересекаются, при $\gamma = 0.5$ половина вершин лежит в перекрывающихся частях сообществ. Для анализа построим графики зависимости описанных функционалов качества от значения γ. Чем выше окажется график, тем лучше работает соответствующий метод по данному критерию.

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 представлены результаты на четырех группах модельных данных. Рассматриваются следующие методы.

1. *SparseGamma* — разреженная гамма модель.
2. *BigClamWeighted* — наивный взвешенный BigClam.
3. *BigClam* — оригинальный BigClam.
4. *COPRA* — label propagation для случая пересекающихся сообществ [7].
5. *NMF* — неотрицательное матричное разложение с квадратичной нормой.
6. *groundtruth* — истинное разбиение на сообщества.
7. *walktrap* — метод выделения непересекающихся сообществ, основанный на случайных блужданиях [14].

Планировалось добавить к сравнению метод CFinder [13], но скорость его работы на порядок ниже перечисленных методов, что не позволило дождаться результатов его работы в приемлемые сроки.

Проанализировав графики, можно заметить, что предложенные методы не лучше остальных в решении данной задачи. Лидируют такие алгоритмы как оригинальный *BigClam*, *COPRA*, *walktrap*. Последний метод занимается поиском непересекающихся сообществ и проигрывает первым двум методам как только значение γ отходит от нуля. Метод *SparseGamma* работает лучше, чем наивное обобщение BigClam, но хуже, чем оригинальный метод. Отметим, что по смешанной модулярности на больших графах BigClam проигрывает двум другим методам.

Интересно, что такой простой метод как *NMF* и наивное обобщение BigClam в данной задаче имеет подавляющее преимущество по функционалам связанными с проводимостью. Причины такого поведения не были до конца прояснены. Причем, как можно судить по NMI, результирующая структура сообществ получилась хуже, чем у других методов. Скорее всего, последняя метрика плохо подходит для этих данных, либо этого класса задач, так как значение истинного разбиения по этой метрике находится в самом низу графика, как и лучшие по NMI методы.

Можно предположить, что в данном модельном примере информация, которую с собой несет вес ребра, не достаточно значительна, чтобы при ее удалении ситуация ухудшалась, а предложенные модели объяснения весов на ребрах не соответствуют действительной природе данных.

6. Результаты работы

В работе предложен новый подход для выделения пересекающихся сообществ во взвешенных гра-

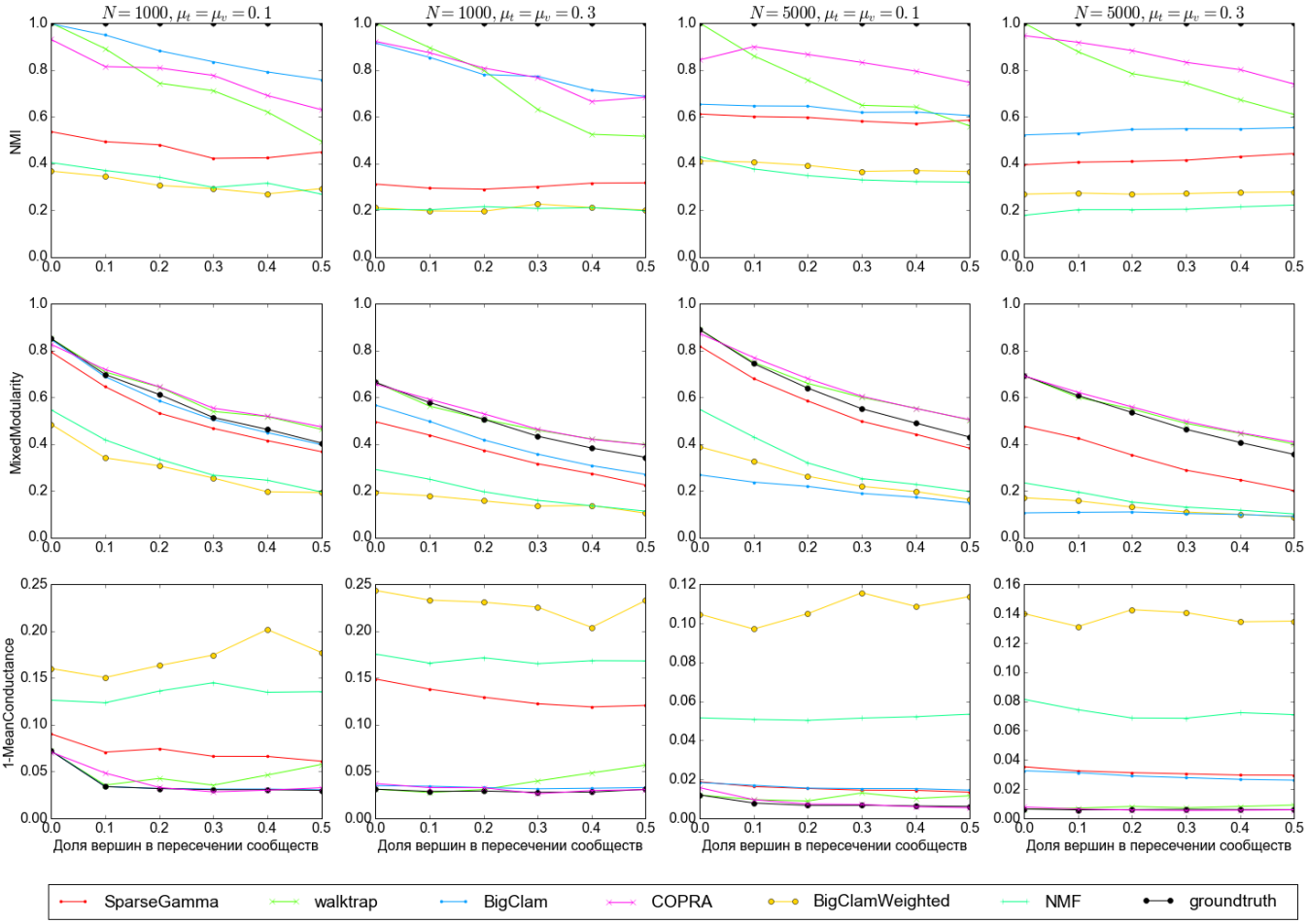

Рис. 6. Результаты работы методов на модельных данных. На графиках изображена зависимость качества разбиения от параметра γ — доли вершин в пересекающихся участках сообществ. По столбцам указаны 4 варианта параметров, остальные параметры зафиксированы и указаны в Таблице 3. По строкам различные метрики: NMI, MixedModularity, 1-MeanConductance (Единица минус среднее значение проводимости по всем сообществам).

фах. Алгоритм является обобщением BigClam на взвешенный случай. Метод протестирован на четырех модельных наборах данных. Полученные результаты говорят о том, что алгоритм работает хуже современных методов, решающих подобную задачу.

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. Из экспериментов видно, что предложенный метод не использует всю ту информацию, которая дополнительно содержится в весах графа. В дальнейшем планируется доработка предложенного подхода.

Список литературы

[1] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing, *Mixed membership stochastic blockmodels*, Journal of Machine Learning Research **9** (2008), no. Sep, 1981–2014, `http://www.arxiv.org/abs/0709.2938`.

[2] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan, *Group formation in large social networks: membership, growth, and evolution*, Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2006, `https://www.cs.cornell.edu/home/kleinber/kdd06-comm.pdf`, pp. 44–54.

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

[4] Santo Fortunato, *Community detection in graphs*, Physics Reports **486** (2010), no. 3, 75–174, `http://www.arxiv.org/abs/0906.0612`.

[5] Michelle Girvan and Mark EJ Newman, *Community structure in social and biological networks*, Proceedings of the national academy of sciences **99** (2002), no. 12, 7821–7826, http://arxiv.org/pdf/cond-mat/0112110.pdf.

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

[7] Steve Gregory, *Finding overlapping communities in networks by label propagation*, New Journal of Physics **12** (2010), no. 10, 103018, https://arxiv.org/abs/0910.5516.

[8] Nevan J Krogan, Gerard Cagney, Haiyuan Yu, Gouqing Zhong, Xinghua Guo, Alexandr Ignatchenko, Joyce Li, Shuye Pu, Nira Datta, Aaron P Tikuisis, et al., *Global landscape of protein complexes in the yeast saccharomyces cerevisiae*, Nature **440** (2006), no. 7084, 637–643, http://www.nature.com/nature/journal/v440/n7084/full/nature04670.html.

[9] Andrea Lancichinetti and Santo Fortunato, *Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities*, Physical Review E **80** (2009), no. 1, 016118, https://arxiv.org/abs/0904.3940.

[10] Andrea Lancichinetti, Santo Fortunato, and János Kertész, *Detecting the overlapping and hierarchical community structure in complex networks*, New Journal of Physics **11** (2009), no. 3, 033015, https://arxiv.org/abs/0802.1218.

[11] Chih-Jen Lin, *Projected gradient methods for nonnegative matrix factorization*, Neural computation **19** (2007), no. 10, 2756–2779, http://ntur.lib.ntu.edu.tw/bitstream/246246/200609271228551177 16/1/pgradnmf.pdf.

[12] 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** (2015), no. 11, 2916–2926, http://mcn.cse.psu.edu/paper/zongqing/tpds-zongqing15.pdf.

[13] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek, *Uncovering the overlapping community structure of complex networks in nature and society*, Nature **435** (2005), no. 7043, 814–818, http://cfinder.org/wiki/papers/communitylettm.pdf.

[14] Pascal Pons and Matthieu Latapy, *Computing communities in large networks using random walks*, Computer and Information Sciences-ISCIS 2005, Springer, 2005, http://arxiv.org/abs/physics/0512106v1, pp. 284–293.

[15] 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** (2013), no. 4, 43, https://arxiv.org/abs/1110.5813.

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