Использование модели социальной сети с сообществами пользователей для распределенной генерации случайных социальных графов

К. К. Чихрадзе, А. В. Коршунов, Н. О. Бузун, Н. Н. Кузюрин
chykhradze@ispras.ru; korshunov@ispras.ru; nazar@ispras.ru; nnkuz@ispras.ru
Институт системного программирования РАН, Москва

Для тестирования алгоритмов определения сообществ пользователей в социальных графах принято использовать графы с известной структурой сообществ в качестве тестовых данных. В статье предложен распределенный метод для генерации случайных социальных графов с реалистичной структурой пользовательских групп. В предложенной модели поддерживается несколько недавно открытых свойств структуры социальных сообществ: плотные пересечения сообществ, суперлинейный рост количества ребер внутри сообщества в зависимости от его размера, а также степенное распределение количества сообществ, к которым принадлежит пользователь. Метод реализован с использованием фреймворка для распределенных вычислений Apache Spark и показал линейную масштабируемость на кластере Amazon EC2. Сгенерированные графы обладают свойствами реальных социальных сетей и могут применяться для оценки точности работы алгоритмов поиска сообществ пользователей в социальных графах с более чем 10^9 пользователей.

Ключевые слова: *социальная сеть; сообщество; случайный граф; Apache Spark*

On a model of social network with user communities for distributed generation of random social graphs

K. K. Chykhradze, A. V. Korshunov, N. O. Buzun, N. N. Kuzyurin
The Institute for System Programming of the Russian Academy of Sciences, Moscow

In the field of social community detection, it is commonly accepted to utilize graphs with reference community structure for accuracy evaluation. The method for generating large random social graphs with realistic structure of user groups is introduced in the paper. The proposed model satisfies some of the recently discovered properties of social community structure: dense community overlaps, superlinear growth of number of edges inside a community with its size, and power law distribution of user-community memberships. Further, the method is by-design distributable and showed near-linear scalability in Amazon EC2 cloud using Apache Spark implementation. The generated graphs possess the properties of real social networks and could be utilized for quality evaluation of algorithms for community detection in social graphs of more than 10^9 users.

Keywords: *social network; community; random graph; Apache Spark*

Введение

Структура сообществ — естественное свойство разного рода сетей, включая социальные сети, которые во многих случаях наследуют характерную для человеческого общества структуру социальных групп. Пользователи социальных сетей объединяются в сообщества как явно (путем вступления во внутрисетевые группы), так и неявно (путем установления связей, основанных на общей роли, деятельности, круге общения, интересах, функциях

или каких-либо других свойствах). Например, часто можно встретить сообщества, в которых участников объединяют общие интересы, политические и религиозные предпочтения, географическая близость и т.д.

К настоящему времени исследователями социальных сетей созданы наборы данных для тестирования методов определения структуры сообществ. Как правило, такие *шаблонные сети* (англ. *benchmark networks*) состоят из набора вершин и ребер социального графа (пользователи и связи между ними), а также списка сообществ, в которых состоит каждый пользователь. Однако сбор реальных данных из сервисов социальных сетей и создание шаблонных сетей часто занимает длительное время, а свойства полученных наборов данных не соответствуют желаемым.

Поэтому важно определить фундаментальные свойства структуры сообществ пользователей, основываясь на реальных шаблонных сетях, и разработать инструмент для генерации синтетических шаблонных сетей со схожими свойствами и различными характеристиками: количество вершин и ребер графа, количество и размеры сообществ, количество сообществ у пользователя и т.д. Таким образом, можно выполнять более надежное и комплексное тестирование методов определения структуры сообществ, поскольку различные значения параметров шаблонной сети могут оказывать существенное влияние на результаты.

Вместе с тем, многие известные генераторы шаблонных сетей не учитывают ряда важных структурных свойств сообществ. Недавние достижения в изучении модульной структуры социальных сетей [1, 2, 3] позволили выявить несколько ранее неизвестных фундаментальных свойств структуры сообществ: повышенная вероятность ребра между парой вершин в области пересечения сообществ, суперлинейный рост количества ребер внутри сообщества в зависимости от его размера, степенной закон распределений количества сообществ у пользователя и размеров сообществ и т. д. Это указывает на необходимость пересмотра требований к методам определения структуры сообществ, а также к методам оценки точности этих методов.

Кроме того, известные генераторы шаблонных сетей имеют существенные ограничения в плане производительности при генерации графов с $> 10^6$ вершин, что затрудняет оценку применимости методов определения структуры сообществ к социальным графам большой размерности.

Основные результаты работы могут быть представлены следующим образом:

— мы разработали *CKB*[1] – метод для распределенной генерации случайных графов с реалистичными свойствами социальных графов и структуры сообществ пользователей. В основе метода лежит *графовая модель принадлежности пользователей к сообществам AGM* [1], где принадлежность пользователей к сообществам моделируется как двудольный граф, а связи между пользователями обусловлены принадлежностью к общим сообществам. Работа метода базируется на разработанных распределенных алгоритмах для генерации двудольного графа «пользователь-сообщество» на основе модели конфигураций [4, 5] и модели Чунг-Лу [6, 7], а также для генерации связей между пользователями внутри сообществ на основе модели Эрдёша–Реньи [8];
— мы сделали предложенный метод особенно удобным для генерации шаблонных сетей для тестирования алгоритмов поиска сообществ, предоставляя набор параметров для настройки наиболее важных структурных свойств генерируемого графа: количество

[1]От первых букв фамилий авторов метода латиницей: **C**hykhradze-**K**orshunov-**B**uzun.

пользователей, средняя степень вершины (среднее число социальных связей пользователя), вероятность ребра внутри сообщества, показатель степени для степенных распределений размеров сообществ и принадлежностей пользователей к сообществам, минимальный и максимальный размер сообщества, минимальное и максимальное число сообществ у пользователя и др.;

— мы реализовали предложенный метод на основе фреймворка для распределенных вычислений Apache Spark и путем экспериментальных исследований на кластере Amazon EC2 подтвердили возможность генерации случайных социальных графов из сотен миллионов вершин с реалистичной структурой сообществ;

— мы сделали нашу разработку доступной для научно-исследовательского сообщества путем предоставления веб-сервиса с возможностью настройки параметров метода и загрузки сгенерированного графа[2].

Обзор известных методов

Современные исследователи в области идентификации социальных сообществ широко применяют шаблонные сети с эталонной структурой сообществ для оценки качества результатов алгоритмов. Графы и соответствующие им множества сообществ могут быть как синтезированы, так и получены из данных реальных социальных сервисов.

Реальные шаблонные сети

Наиболее известным хранилищем данных социальных сетей со структурой сообществ является Stanford Large Network Dataset Collection[3]. Для создания этих наборов данных авторы исследования Янг и Лесковец [2] использовали различные социальные сети (LiveJournal, Friendster, Orkut, а также 225 различных социальных сетей на платформе Ning), где пользователи создают явные группы для общения и обмена контентом. Эти группы созданы на основе специфических тем, интересов, хобби и географического положения. Например, в LiveJournal группы категоризованы по следующим типам: культура, развлечения, игры, спорт, студенческая жизнь, технологии и др.

Авторы полагают, что каждая такая явная группа является сообществом. Для представления всех сетей согласовано сеть полагается невзвешенным ненаправленным статичным графом. Так как члены групп могут быть отделены от остальной части сети, компонента связности группы полагается отдельным сообществом. Тем не менее сообщества могут быть вложенными и пересекаться.

Синтетические шаблонные сети

Модель Гирвана–Ньюмена (GN) [9]. В графе, созданном в соответствии с этой моделью, вершины разбиты на l эквиразмерных групп по g вершин в каждой. Вершины из одинаковых групп соединяются ребром с вероятностью p_{in}, а вершины из разных групп — с вероятностью p_{out}. Внутри каждой группы ребра генерируются в соответствии с моделью Эрдёша–Реньи для генерации случайного графа с вероятностью связи p_{in}. Ожидаемые внутренняя и внешняя средние степени графа равны соответственно $z_{\text{in}} = p_{\text{in}}(g-1)$ и $z_{\text{out}} = p_{\text{out}}g(l-1)$. Средняя степень вершины во всем графе $\langle k \rangle = z_{\text{in}} + z_{\text{out}}$.

Также существуют и другие варианты данной модели: с сообществами разного размера [10], с пересекающимися сообществами [11], иерархическая [12] и взвешенная версии [16].

[2] http://ckb.at.ispras.ru/

[3] http://snap.stanford.edu/data/index.html

Таким образом, модель Гирвана–Ньюмена создает взаимно связанные между собой случайные графы модели Эрдёша–Реньи. Поэтому все вершины имеют примерно одну и ту же степень. Кроме того, все сообщества по умолчанию создаются эквиразмерными. Эти два свойства не соответствуют структуре реальных социальных сетей. Распределения степеней обычно подчинено степенному закону: вершин с маленькой степенью на порядки больше, чем вершин с большой степенью. Схожее свойство имеет и распределение размеров сообществ.

Блочная двухуровневая модель Эрдёша–Реньи (BTER) [14] позволяет генерировать социальные графы со степенными распределениями для распределения степеней вершин и размеров сообществ. Генерация проходит в 3 этапа. Прежде всего, выполняется *предварительная обработка*: каждая вершина степени 2 или выше распределяется в сообщество. Далее, в *Фазе 1* моделируется локальная структура внутри каждого сообщества как граф Эрдёша–Реньи. Вероятность ребра для сообщества G_k определяется как $\rho_k = \rho\left[1 - \eta\left(\log(d_k+1)/\log(d_{\max}+1)\right)^2\right]$, где $d_k = \min\{d_i | i \in G_k\}$, $d_{\max}$ – максимальная степень вершины во всем графе, а ρ и η – параметры. После этого во время *Фазы 2* создаются связи между сообществами. Применяется модель Чунг-Лу [6] для исходящей степени вершины e_i, которая определяется следующим образом:

$$e_i = \begin{cases} 1, & \text{если } d_i = 1; \\ d_i - \rho_{k_i}(|G_{k_i}| - 1), & \text{иначе,} \end{cases}$$

где $|G_k|$ — размер k-го сообщества.

Минусы этой модели в том, что она не позволяет создавать сети с пересекающейся структурой сообществ, а также вероятность ребра не зависит от размера сообщества.

Модель **случайных графов пересечений (RIT)** [15, 16] может быть представлена следующим образом: V – множество вершин ($|V| = n$), A – множество множеств из m элементов. Для $p \in [0, 1]$ строится двудольный граф $B(n, m, p)$ с двумя долями вершин V и A включением каждого из возможных nm ребер между элементами из V и элементами из A независимо с вероятностью p. После создается случайный граф пересечений $G(n, m, p)$ с множеством вершин V путем связывания двух различных вершин $i, j \in V$ если и только если существует элемент $a \in A$ такой, что и i, и j смежны с a в $B(n, m, p)$.

Если рассматривать вершины в V как пользователей, а элементы множества A как сообщества, то получается модель социальной сети, в которой пара пользователей может быть связана ребром, если они одновременно состоят в хотя бы одном общем сообществе.

Недостатком модели RIT является необходимость задания количества сообществ m в качестве входного параметра, а также несоответствующее степенному закону распределение степеней вершин. Кроме того, отсутствует возможность управления вероятностью ребра в отдельных сообществах.

Генератор шаблонных сетей Ланчичинетти–Фортунато–Радиччи (LFR) [17] основан на более реалистичной модели социального графа с сообществами. В этой модели распределения степеней вершин и размеров сообществ генерируются в соответствии со степенным законом с различными экспонентами τ_1 и τ_2 соответственно.

Сам граф строится следующим образом:

1. Генерируется последовательность размеров сообществ, подчиняющаяся степенному закону с экспонентой τ_2.
2. Генерируется последовательность степеней вершин, подчиняющаяся степенному закону с экспонентой τ_1. Для каждой вершины i со степенью k_i определяется внутренняя

степень $k_i^{(in)} = (1 - \mu_t)k_i$, где $0 \leqslant \mu_t \leqslant 1$ — *топологический параметр смешивания*. Внутренняя степень вершины i соответствует числу ее соседей, которые состоят как минимум в 1 общем сообществе с i.

3. Вершины в каждом сообществе соединяются с использованием модели конфигураций [5].
4. Для каждой вершины вычисляется внешняя степень $k_i^{(\text{out})} = k_i - k_i^{(\text{in})}$, после чего вершины случайным образом соединяются ребрами с другими вершинами из различных сообществ. При этом сохраняется внутренняя степень $k_i^{(\text{in})}$ для вершин, входящих в несколько сообществ.

Вместо модели конфигураций на этапе генерации ребер внутри сообществ Орман и Лабатут [18] предложили использовать модель предпочтительного присоединения Барабаши-Альберта (BA) и одну из вариаций этой модели – эволюционную модель предпочтительного присоединения (EV). Обе модификации позволяют генерировать сети с меньшей средней длиной пути и с большей корелляцией степеней вершин по сравнению с оригинальным методом LFR.

Несмотря на тот факт, что шаблонные сети LFR являются де-факто золотым стандартом для оценки результатов алгоритмов поиска сообществ, у них есть несколько существенных недостатков:

— вершины делятся на пересекающиеся и непересекающиеся, а все вершины из пересечений входят в одно и то же количество сообществ;
— все вершины имеют один и тот же топологический параметр смешивания;
— длительное время генерации.

Генератор шаблонных сетей agmgen из Стэнфордского проекта по анализу сетей[4] основан на предложенной разработчиками *графовой модели принадлежности пользователей к сообществам* (англ. *Community-Affiliation Graph Model*, или *AGM*) [1].

AGM – вероятностная генеративная модель для графов, которая наиболее достоверно воспроизводит организацию сетей в виде пересекающихся сообществ. Она была разработана путем анализа данных о реальных сообществах пользователей в социальных сетях. Модель представляет принадлежность вершин к сообществам как двудольную сеть, в которой ребра от пользователя идут к сообществам, которым этот пользователь принадлежит.

Другая часть модели основана на том, что люди принадлежат многим сообществам (друзья, члены семьи, коллеги), но связи между ними часто возникают как результат одной доминирующей причины. Это моделируется с помощью того, что у каждого сообщества есть вероятность, с которой вершина будет соединена ребром с другой вершиной из этого сообщества. Это означает, что каждое сообщество, которому принадлежит некая пара вершин, имеет независимый шанс создать ребро между этими двумя вершинами. Следовательно, чем большему количеству сообществ принадлежит пара пользователей, тем больше вероятность того, что между ними будет образовано ребро.

Генератору agmgen в качестве входных данных требуется двудольный граф, задающий принадлежность пользователей сообществам. Далее выполняется генерация ребер независимо в каждом сообществе по модели Эрдёша–Реньи.

Вместе с тем, при генерации шаблонных сетей удобнее указать параметры генерации графа «пользователь–сообщество», нежели использовать данные из реальных сетей или

[4] `http://snap.stanford.edu/snap/description.html`

генерировать такой граф отдельно. Кроме того, agmgen (как и LFR) не позволяет генерировать графы из сотен миллионов вершин, что затрудняет оценку применимости методов определения структуры сообществ к социальным графам большой размерности.

Тем не менее, AGM является одной из наиболее реалистичных на данный момент моделей социальной сети с сообществами пользователей. Поэтому предложенный в данной работе метод генерации шаблонных сетей также основывается на этой модели, но при этом лишен описанных недостатков agmgen.

Постановка задачи

Рассмотрим граф $G = (V, E)$, где $|V| = N_1$ и $|E| = m$. Сообщество C_i определяется как индуцированный подграф, размер сообщества $|C_i| = n_{c_i}$. Количество сообществ N_2, все сообщества вместе составляют *покрытие* графа. Количество вхождений j-й вершины в разные сообщества – m_j.

Внутренняя (d^{int}_{j,C_i}) и внешняя (d^{ext}_{j,C_i}) степени вершины $j \in C_i$ определяются как количество ребер, соединяющих j с другими вершинами в C_i или с остальной частью графа соответственно. Тогда общая степень вершины j равна $d_j = d^{\text{int}}_{j,C_i} + d^{\text{ext}}_{j,C_i}$. Количество ребер внутри сообщества C_i – $d_{c_i} = \frac{1}{2}\sum_{\forall j \in C_i} d^{\text{int}}_{j,C_i}$.

Список входных параметров для генерации приведен в табл. 1.

Задача состоит в том, чтобы сгенерировать граф G со следующими свойствами[5]:

1. распределение степеней вершин графа подчиняется степенному закону с экспонентой β [20]:

$$p(d) \sim d^{-\beta} \tag{1}$$

2. присутствует одна большая компонента связности:

$$\exists\, G^* \subset G : |V^*| \sim N_1, \forall\, i, j \in V^* \;\exists\, w_1, w_2, ..., w_k \in E^* \;\exists\, t_l \in V: \\ w_0 = (i, t_0), w_1 = (t_0, t_1), ..., w_D = (t_{D-1}, j) \tag{2}$$

3. малый эффективный диаметр [21]:

$$\forall \varepsilon > 0, N_1 \to \infty\colon \mathbb{P}\left((1-\varepsilon)\frac{\ln N_1}{\ln\ln N_1} \leqslant D \leqslant (1+\varepsilon)\frac{\ln N_1}{\ln\ln N_1}\right) \to 1, \tag{3}$$

где

$$D = \max_{i,j \in V^*}(d_{i,j})$$

$$d_{i,j} = \min_{w_s \in E^*}(|\{w_1, w_2, ..., w_k | w_0 = (i, t_0), w_1 = (t_0, t_1), ..., w_k = (t_k, j)\}|)$$

4. вершины могут иметь нулевую степень и/или не входить ни в одно сообщество:

$$\forall i \in \mathbb{N} \Rightarrow d_i \geqslant 0,\ m_i \geqslant 0 \tag{4}$$

5. сообщества могут пересекаться [22]:

$$\exists\, i, j \in \mathbb{N} : C_i \cap C_j \neq \varnothing \tag{5}$$

[5]В работе рассматриваются наиболее общепринятые свойства, подтвержденные рядом исследований. Однако некоторые из этих свойств часто подвергаются критике или дорабатываются (улучшаются). Так, например, Алессандра Сала и др. утверждают, что распределение степеней в социальных сетях подчиняется логнормальному закону [19].

6. каждое сообщество C_i связно с большой вероятностью:

$$\forall k \in \mathbb{N} \Rightarrow \mathbb{P}(C_k = (V_k, E_k) : \forall\ i, j \in C_k\ \exists\ w_t \in E_k : \\ w_0 = (i, t_0), w_1 = (t_0, t_1), ..., w_l = (t_l, j)) \geqslant 1 - \frac{1}{n_{C_i}} \tag{6}$$

7. плотность ребер внутри сообществ больше, чем средняя плотность ребер во всем графе G [23]:

$$\forall i \in \mathbb{N} \Rightarrow \frac{d_{C_i}}{n_{C_i}(n_{C_i} - 1)} > \frac{m}{N_1(N_1 - 1)} \tag{7}$$

8. количество ребер внутри сообщества больше, чем количество ребер, соединяющих вершины сообщества с остальной частью графа:

$$\forall i \in \mathbb{N} \Rightarrow d_{C_i} > \sum_{\forall j \in C_i} d^{\text{ext}}_{j,C_i} \tag{8}$$

9. количество ребер в сообществе растет суперлинейно с размером сообщества [3]:

$$\forall i \in \mathbb{N} \Rightarrow d_{C_i} \propto n_{C_i}^{1+\gamma},\ \gamma \in (0, 1) \tag{9}$$

10. распределение пользователей по сообществам подчиняется степенному закону с экспонентой β_1 [3]:

$$\forall i \in \mathbb{N} \Rightarrow p(m_i) \sim m_i^{-\beta_1} \tag{10}$$

11. распределение размеров сообществ подчиняется степенному закону с экспонентой β_2 [17]:

$$\forall i \in \mathbb{N} \Rightarrow p(n_{C_i}) \sim n_{C_i}^{-\beta_2} \tag{11}$$

12. пересечение сообществ более плотно, чем их непересекающаяся часть [1]:

$$\forall i, j \in \mathbb{N}\ (i \neq j), C_{i \cap j} = C_i \cap C_j \Rightarrow \frac{d_{C_{i \cap j}}}{n_{C_{i \cap j}}(n_{C_{i \cap j}} - 1)} > \frac{d_{C_i}}{n_{C_i}(n_{C_i} - 1)} \tag{12}$$

Предложенный метод

Основные шаги генератора СКВ следующие:

1. Пользователи распределяются по сообществам с использованием модифицированных модели Чунг–Лу [6, 7] и модели конфигураций для двудольных графов [4, 5];
2. Связи между пользователями внутри каждого сообщества генерируются по модели Эрдёша–Реньи [8].

Генерация двудольного графа «пользователь–сообщество»

Двудольным называется граф, вершины которого могут быть разделены на 2 непересекающихся множества U и V таких, что каждое ребро соединяет вершину в U с вершиной в V. В нашем случае V ($|V| = N_1$) является множеством пользователей, а U ($|U| = N_2$) — множеством сообществ. Связи между пользователями и сообществами моделируются ребрами в двудольном графе.

Таблица 1. Входные параметры для генерации графа

Параметр	**Значение**	**По умолчанию**
N_1	Количество вершин	—
d_{mean}	Средняя степень вершин	—
$x_{\min}$	Минимальное число вхождений пользователя в сообщества	1
$m_{\min}$	Минимальный размер сообщества	2
$x_{\max}$	Максимальное число вхождений пользователя в сообщества	10 000
$m_{\max}$	Максимальный размер сообщества	10 000
$\beta_1 > 1$	Экспонента степенного распределения числа вхождений пользователей в сообщества	2,5
$\beta_2 > 1$	Экспонента степенного распределения размеров сообществ	2,5
$\alpha > 0$	Влияет на вероятность ребра внутри сообщества	4
$0 < \gamma < 1$	Влияет на вероятность ребра внутри сообщества	0,5
ε	Управляет количеством ребер в ε-сообществе	$2N_1^{-1}$

1. Рассчитывается количество сообществ N_2 исходя из уравнения:

$$M_0 = N_1 \cdot \mathbb{E}[m] = N_2 \cdot \mathbb{E}[x], \tag{13}$$

где $\mathbb{E}[m]$ и $\mathbb{E}[x]$ – математические ожидания числа вхождений пользователей в сообщества и размеров сообществ соответственно и вычисляется по формуле

$$\mathbb{E}[x] = \int\limits_{x_{\min}}^{x_{\max}} xp(x)dx = \frac{(1-\beta)(x_{\max}^{2-\beta} - x_{\min}^{2-\beta})}{(x_{\max}^{1-\beta} - x_{\min}^{1-\beta})(2-\beta)}, \tag{14}$$

2. Генерируются две степенные последовательности для размеров сообществ и количества сообществ у пользователя: каждой вершине ставится в соответствие степень (d_i^1 для i-й вершины-пользователя и d_j^2 для j-й вершины-сообщества).
3. Вычисляются значения $D_1^1 = d_1^1$, $D_2^1 = D_1^1 + d_2^1$, ... , $D_{k+1}^1 = D_k^1 + d_{k+1}^1$, ..., $D_{N_1}^1 = D_{N_1-1}^1 + + d_{N_1}^1$ и $D_1^2 = d_1^2$, $D_2^2 = D_1^2 + d_2^2$, ... , $D_{k+1}^2 = D_k^2 + d_{k+1}^2$, ... ,$D_{N_2}^2 = D_{N_2-1}^2 + d_{N_2}^2$.
4. Вычисляется количество генерируемых ребер между долями биграфа:

$$M = M_0 + \mathbb{E}[Y], \tag{15}$$

где $\mathbb{E}[Y]$ – математическое ожидание количества кратных ребер (см. раздел «Кратные ребра»).
5. Для последовательности натуральных чисел

$$[M] = \{1, 2, 3, \ldots, \lfloor M \rfloor\},$$

где $\lfloor x \rfloor = \max\{n \in \mathbb{Z} | n \leqslant z\}$,
выполняется в цикле для $t = 1$ до $\lfloor M \rfloor$:

(a) Выбирается случайное число p и q из $[M]$ равномерно;

(б) Находится интервал $[D_i^1, D_{i+1}^1]$, которому принадлежит p;
(в) Находится интервал $[D_j^2, D_{j+1}^2]$, которому принадлежит q;
(г) Если $i \neq j$, то в двудольный граф добавляется ребро (i, j).

6. Все сгенерированные ребра сортируются, кратные ребра удаляются.

Вычислительная сложность этого шага – $O(M \log(N_1 N_2))$.

Генерация ребер внутри сообществ

На этом шаге создаются ребра в соответствии с принадлежностями вершин к сообществам. Для этого исходя из заданной средней степени вычисляется вероятность ребра в каждом сообществе [3]:

$$p_{c_k} = \frac{\alpha}{x_{c_k}^{\gamma}}, \tag{16}$$

где x_{c_k} — размер сообщества, $\gamma \in (0; 1)$ – параметр модели, α определяется средней степенью ($\alpha > 0$). Более подробный алгоритм вычисления параметров алгоритма исходя из заданной средней степени ниже в соответствующем разделе «Средняя степень».

Стоит отметить, что при этом суммарная вероятность ребра между i и j во всем графе растет в зависимости от количества общих сообществ между этой парой вершин [3]:

$$p(i, j) = 1 - \prod_{c_k \in C_{ij}} (1 - p_{c_k}), \tag{17}$$

где C_{ij} — множество сообществ, которым принадлежит i и j.

Далее количество ребер в сообществе C_j сэмплируется из биномиального распределения с учетом вероятности кратного ребра:

$$M_{c_j} = (1 + \mathbb{P}_{c_k}^{\text{mult}}) \text{Bin}(x_{c_k}, p_{c_k}), \tag{18}$$

где $\mathbb{P}_{c_k}^{\text{mult}}$ — это вероятность кратного ребра (см. раздел «Кратные ребра»).

Затем генерируются M_{c_j} ребер с использованием модели Эрдёша–Реньи. В конце все ребра сортируются, а кратные ребра удаляются. Петли фильтруются в процессе генерации.

ε-сообщество

Поскольку генерация ребер на втором шаге происходит только внутри сообществ, то ребра между сообществами появляются только за счет связей между вершинами, состоящими в нескольких сообществах. Чтобы увеличить вероятность появления ребер между сообществами, к множеству сообществ добавляется так называемое ε-сообщество [1], которое соединяет случайную пару вершин из всего графа с некоторой вероятностью ε. Этот шаг также необходим для того, чтобы обеспечить существование вершин с нулевым количеством сообществ и ненулевой степенью в графе. Другими словами, некоторая часть пользователей может образовывать связи, при этом не являясь членами каких-нибудь сообществ. Обратное тоже верно — пользователь может входить в некоторые сообщества, но при этом не иметь никаких связей.

Количество генерируемых в ε-сообществе ребер:

$$M_\varepsilon = \frac{N_1(N_1 - 1)}{2}\varepsilon, \tag{19}$$

где ε – параметр.

Вычислительная сложность этапа генерации ребер внутри сообществ – $O(K_m)$, где $K_m = \sum_{c_j} M_{c_j}$.

Кратные ребра

Поскольку ребра в предложенном методе генерируются независимо, то после удаления кратных ребер из общего списка по окончании генерации количество ребер несколько отличается от ожидаемого значения, заданного в соответствии с входными параметрами. Таким образом, учитывание вероятности кратного ребра на каждом шаге генерации позволяет уменьшить погрешность, связанную с удалением кратных ребер.

Так, для процесса генерации двудольного графа «пользователь-сообщество» вероятность кратного ребра равна

$$P_{c_i,j}^{\geqslant 2} \approx \left(\frac{x_{c_i} m_j}{M_0}\right)^2. \tag{20}$$

Математическое ожидание $\mathbb{E}[Y]$ количества ребер, появившихся два или более раз, в этом случае:

$$\mathbb{E}[Y] \approx \frac{1}{2}\sum_c \sum_i \left(\frac{x_{c_i} m_j}{M_0}\right)^2 = \frac{1}{2M_0^2}\sum_{c_i} x_{c_i}^2 \sum_i m_j^2. \tag{21}$$

Для процесса генерации ребер внутри сообществ вероятность кратного ребра может быть посчитана как

$$\mathbb{P}_{c_k}^{mult} \sim 1 - e^{-\frac{p_c}{2}} - \frac{p_c}{2} e^{-\frac{p_c}{2}} \sim \frac{p_c}{2} - \frac{p_c}{2}\left(1 - \frac{p_c}{2}\right) = \frac{p_c^2}{4} = \frac{\alpha^2}{4n^{2\gamma}}. \tag{22}$$

Средняя степень

Так как средняя степень является важным свойством для анализа графов и тестирования алгоритмов поиска сообществ, была исследована зависимость между входными параметрами α и γ и средней степенью.

Рассмотрим случайную величину ξ_{c_k}:

$$\xi_{c_k} = \begin{cases} 1, & \text{если } (i,j) \in C_k; \\ 0, & \text{если } (i,j) \notin C_k, \end{cases} \tag{23}$$

где C_k – размер сообщества.

p_{i_k} – вероятность того, что i-я вершина содержится в C_k:

$$p_{i_k} = p_{i_k}(i \in C_k) = 1 - \left(1 - \frac{x_c}{\sum_c x_c}\frac{m_i}{\sum_k m_k}\right)^M, \tag{24}$$

где M – количество ребер в двудольном графе.

Учитывая, что $x_c m_i / M^2 \ll 1$ и $\sum_{c_k} x_{c_k} = \sum_k m_k = M$, получаем:

$$p_{i_k} = p_{i_k}(i \in C_k) = p_{i_k}(m_i, x_{c_k}) = 1 - \left(1 - \frac{x_{c_k} m_i}{M^2}\right)^M \approx \frac{x_{c_k} m_i}{M} \tag{25}$$

Вероятность, что ребро (i,j) сгенерируется в C_k:

$$p(\xi_{c_k} = 1) = p_{i_k} p_{j_k} p_{c_k}, \tag{26}$$

где $p_{c_k} = \alpha / x_{c_k}^{\gamma}$ – это вероятность ребра в сообществе, а x_{c_k} — размер сообщества.

Рассмотрим новую с.в. ξ_{ij}:

$$\xi_{ij} = \begin{cases} 1, & \text{если } \exists c : \xi_c = 1; \\ 0, & \text{если } \forall c : \xi_c = 0. \end{cases} \tag{27}$$

Математическое ожидание кратного ребра во всем графе (используя принцип включений-исключений):

$$\mathbb{E}\xi_{ij} = \sum_{i=1}^{k} (-1)^{i+1} S_i, \tag{28}$$

где k – максимальная кратность ребра в графе ($k \leqslant N_2$), а

$$S_r = \alpha^r \mathbb{E}\left[\sum_{c_1 < \ldots < c_r} \prod_{k=\{1,\ldots,r\}} \frac{1}{x_{c_k}^{\gamma}} \prod_{t=\{i,j\}} \left(\frac{x_{c_k} m_t}{M} \right) \right] \tag{29}$$

При этом необходимо отметить, что

$$\mathbb{E}[\sum_{c_1 < c_2 < \ldots < c_p} x_{c_1}^{\theta_1} x_{c_2}^{\theta_2} \cdots x_{c_p}^{\theta_p}] = \binom{N_2}{p} \mathbb{E}[x_{c_1}^{\theta_1}] E[x_{c_2}^{\theta_2}] \cdots \mathbb{E}[x_{c_p}^{\theta_p}] \tag{30}$$

Поэтому средняя степень при условии заданных размеров сообществ может быть задана как

$$\mathbb{E}[d | x_1, x_2 \ldots x_{N_2}] = \mathbb{E}\xi_{ij} N_1 \tag{31}$$

Таким образом, средняя степень равна

$$d_{\text{mean}} = \mathbb{E}[\mathbb{E}[d | x_1, x_2 \ldots, x_{N_2}]] \approx \sum_{x_1, x_2 .. x_{N_2}} \mathbb{E}\xi_{ij} N_1 \prod_{j=1}^{N_2} p(x_j), \tag{32}$$

где N_1 — это количество вершин во всем графе, а $p(x_j)$ — вероятность того, что сгенерируется сообщество размера x_j.

Теперь, после решения уравнения k-ой степени[6] с переменной α (и фиксированной γ), можно вычислить вероятность p_{c_i}, необходимую для достижения заданной средней степени (см. раздел «Генерация ребер внутри сообществ»).

Связность сообществ

Используя результат работы Эрдёша–Реньи [7, 8, 24], который можно сформулировать в виде следующей теоремы

Теорема 1. Рассмотрим модель $G(n, p)$. Пусть $p = (c \log n)/n$. Если $c > 1$, то почти всегда случайный граф связен. Если $c < 1$, то почти всегда случайный граф не является связным.

[6]Для упрощения вычислений в уравнении 28 можно рассматривать не более трех слагаемых, так как из эмпирического анализа следует, что $S_i = o(S_1)\ \forall i \geqslant 4$. Это означает, что количество ребер кратности более 3 незначительно по сравнению с общим количеством ребер.

можно утверждать, что

Теорема 2. Сообщество C_i связано с большой вероятностью для

$$\alpha > \ln(x_{c_i}) x_{c_i}^{\gamma-1} \tag{33}$$

Распределенная реализация

Предложенный метод был реализован на языке программирования Scala с использованием Apache Spark[7] - фреймворка для распределенных вычислений в распределенной среде. Данный фреймворк позволяет эффективно выполнять различные операции за счет использования структурной абстракции данных, именуемой *сбоеустойчивые распределенные наборы данных* (англ. *Resilient Distributed Datasets*, или *RDD*). RDD – это коллекция объектов, распределенных между множеством машин, которая может быть восстановлена, если какая-то часть объектов утеряна. По умолчанию все RDD хранятся в оперативной памяти. При обновлении данных старый RDD не изменяется, но создается новый, содержащий ссылку на предыдущую версию и список изменений. Перечисленные особенности Spark позволяют увеличивать скорость работы программы в несколько раз по сравнению с другими платформами для распределенных вычислений, в частности, Apache Hadoop.

Схема работы генератора на вычислительном кластере показана на рис. 1.

Главный узел (мастер) – главная машина в вычислительном кластере. *Ведомыми узлами (слэйвами)* называются остальные машины кластера. Файлы HDFS (Hadoop Distributed File System) распределены в локальной файловой системе на слэйвах. Нераспределенная часть вычислений выполняется на главном узле. В процессе распределенных вычислений главный узел назначает задачи слэйвам, координирует их работу и агрегирует результаты.

На первом шаге процесса генерации создаются две степенные последовательности, которые фактически являются распределением размеров сообществ и распределением количества вхождений пользователей в сообщества. Далее мастер отправляет эти последовательности на каждый из слэйвов. На следующем шаге каждый из s слэйвов независимо генерирует $\frac{M}{s}$ ребер (см. раздел «Генерация двудольного графа «пользователь–сообщество»») между пользователями и сообществами в двудольном графе. Все ребра объединяются в один список ребер, и на его основе создается список сообществ, который сохраняется в распределенном файловом хранилище (в данном случае – HDFS).

После этого мастер считывает из хранилища сообщества, создает v копий каждого сообщества, равномощно группирует их и равномерно распределяет их по слэйвам. Далее каждый из ведомых узлов генерирует M_{c_j}/v ребер каждого полученного сообщества C_j, а также M_ε/v ребер при генерации ε-сообщества (см. раздел «Генерация ребер внутри сообществ»). Все ребра объединяются в общий список и записываются в HDFS.

Масштабируемость

Для оценки масштабируемости алгоритма был использован кластер Amazon EC2 из 2, 4, 8 и 16 машин типа *m1.large*[8]. На рис. 2 показано, что алгоритм имеет близкую к линейной масштабируемость, что озволяет создавать синтетические сети больших размеров

[7] `http://spark.incubator.apache.org/`

[8] `http://aws.amazon.com/ec2/previous-generation/`

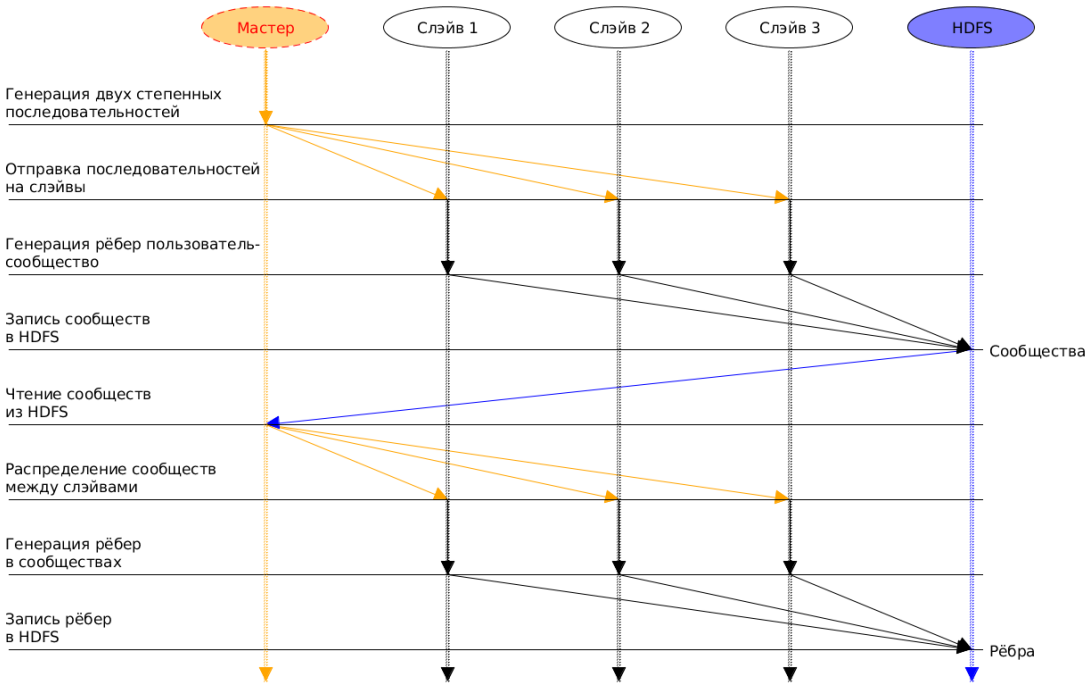


Рис. 1. Схема работы распределенной реализации генератора СКВ

за приемлемое время. Так, например, генерация графа с 1 миллиардом вершин заняла 150 мин на 150 машинах кластера Amazon EC2.

Однако в некоторых случаях может иметь место недостаточное ускорение работы генератора с ростом числа машин. Так, например, локально алгоритм отработал быстрее, чем на двух слэйвах. В данном случае это связано с тем, что на пересылку данных по сети требуется не меньше времени, чем собственно на процесс генерации.

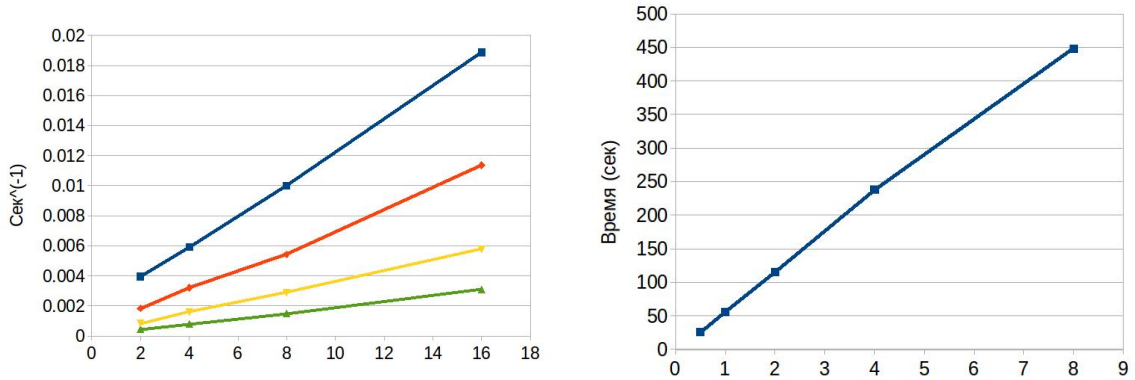


Рис. 2. Слева: масштабируемость на кластере Amazon EC2 из машин типа *m1.large*. Синяя линия — граф с $4 \cdot 10^6$ вершин, красная линия — $8 \cdot 10^6$ вершин, желтая линия — $16 \cdot 10^6$ вершин, зеленая линия — $32 \cdot 10^6$ вершин. **Справа: временная сложность**

Сравнение с другими шаблонными сетями

Таблица 2 содержит результаты сравнения сгенерированных графов с реальными шаблонными сетями на основе данных LiveJournal, ORKUT и YouTube из Stanford Large Network Dataset Collection[9], а также с синтетическими сетями, созданными популярным генератором LFR. На рис. 3, 4, 5, 6 изображены распределения степеней вершин, размеров сообществ и числа вхождений пользователей в сообщества для сравниваемых сетей.

По результатам сравнения можно заключить, что среди сравниваемых синтетических шаблонных сетей CKB имеют наиболее схожую структуру с реальными сетями. Единственным отличием сетей CKB от реальных сетей является небольшое значение коэффициента кластеризации[10], что объясняется использованием модели Эрдёша–Реньи для генерации ребер внутри сообществ. Достижение более реалистичного коэффициента кластеризации требует изменения процесса генерации ребер и является объектом дальнейшей работы.

Восстановление известной структуры сообществ

Также была протестирована эффективность различных алгоритмов определения структуры сообществ на сгенерированных графах при различных значениях параметра β_1 — экспоненты степенного распределения числа вхождений пользователей в сообщества. Увеличение β_1 при неизменных значениях остальных параметров приводит к увеличению среднего числа сообществ у пользователя, что делает структуру сообществ больше сложной для определения.

Для сравнения были выбраны следующие алгоритмы, являющиеся представителями различных классов методов определения структуры сообществ в социальном графе: OSLOM[11], GCE[12], SLPA[13], MOSES[14].

Алгоритмы получали на вход список ребер и возвращали найденное *покрытие* — множество сообществ в исходном графе. Для того, чтобы установить близость известного и найденного покрытий (X, Y), используется NMI — *нормализованная взаимная информация* [17]:

$$NMI(X,Y) = 1 - \frac{1}{2}[H(X|Y)_{\text{norm}} + H(Y|X)_{\text{norm}}]. \tag{34}$$

Для каждого сообщества X_k находится ближайшее Y_k в смысле неопределенности информации $H(X_k|Y_j) \to \min\limits_j$, где X_k — случайная переменная, соответствующая вероятности возникновения вершины в сообществе k, $H(X_k|Y_j)$ — условная энтропия X_k при условии Y_j. H_{norm} вычисляется как нормализация $H(X_k|Y_j)$ от количества всей информации о X_k, усредняя по всем сообществам в X.

---

[9] http://snap.stanford.edu/data/index.html#communities

[10] Коэффициент кластеризации $\bar{C}$ задается следующим образом: $\bar{C} = \frac{1}{n}\sum_{i=1}^{n} C'_i$, где n – количество вершин в графе, $C'_i = \frac{2|\{e_{jk}: j,k \in N_i, e_{jk} \in E\}|}{d_i(d_i-1)}$ – локальный коэффициент кластеризации вершины i, $N_i = \{j : e_{ij} \in E\}$ – множество соседей для вершины i, E – множество ребер, d_i – степень вершины i.

[11] http://www.oslom.org/

[12] https://sites.google.com/site/greedycliqueexpansion/

[13] https://sites.google.com/site/communitydetectionslpa/

[14] http://www.cliquecluster.org/moses

Таблица 2. Сравнение шаблонных сетей CKB ($\beta_1 = \beta_2 = 2.5$), LFR, LiveJournal, ORKUT и YouTube

	Orkut	**LiveJournal**	**YouTube**	**CKB**		**LFR**
Количество вершин	3M	4M	1,1M	3M	97,5K	100K
Средняя степень	76,2	17,3	5,3	109,9	68,8	66,7
Экспонента степенного распределения размеров сообществ β_2	2,12	2,14	2,36	2,19	2,57	2,54
Экспонента степенного распределения принадлежности пользователей к сообществам β_1	1,59	2,22	2,83	2,28	2,62	–
Экспонента степенного распределения степеней β	1,58	2,15	2,53	2,22	2,54	2,56
Медиана распределения размеров сообществ	16	2	3	5	49	40
Медиана распределения принадлежности пользователей к сообществам	14	2	1	7	1	1
Средний коэффициент кластеризации	0,169	0,353	0,172	0,039	0,055	0,226
Эффективный диаметр d_{eff}	4,8	6,4	6,5	4,38	3,88	3,98
Время генерации (с)	—	—	—	160	11	863

NMI лежит в промежутке $[0; 1]$. Минимальное значение соответствует абсолютно разным покрытиям, максимальное – совпадающим покрытиям.

На рис. 7 представлены результаты тестирования. Видно, что лишь 2 из 4 алгоритмов — MOSES и OSLOM — смогли восстановить сгенерированную структуру сообществ с приемлемой точностью. В табл. 3 более подробно исследована зависимость NMI от параметров генерации входных графов для алгоритма MOSES. Можно отметить, что точнее всего сообщества определяются, когда их размер ограничен снизу 10 вершинами, а параметр γ достаточно велик.

N_1	m_{min}	x_{min}	x_{max}	m_{max}	β_1	β_2	α	γ	ε	NMI	d_{mean}
10 000	1	2	1000	1000	2,5	2,5	1	0,3	0	0,59	67,4
10 000	1	2	1000	1000	3,1	3,1	1	0,3	0	0,72	32,3
10 000	1	2	1000	1000	3,1	3,1	2	0,3	0	0,75	57,3
10 000	1	10	1000	1000	2,5	2,5	1	0,5	0	0,88	90,8
10 000	1	10	1000	1000	2,5	2,5	1	0,5	0,0005	0,84	107,1
10 000	2	10	1000	1000	2,5	2,5	0,5	0,5	0	0,52	72,0
10 000	1	10	1000	1000	2,5	2,5	0,2	0,2	0	0,65	88,9

Таблица 3. Зависимость NMI от параметров СКВ для алгоритма MOSES

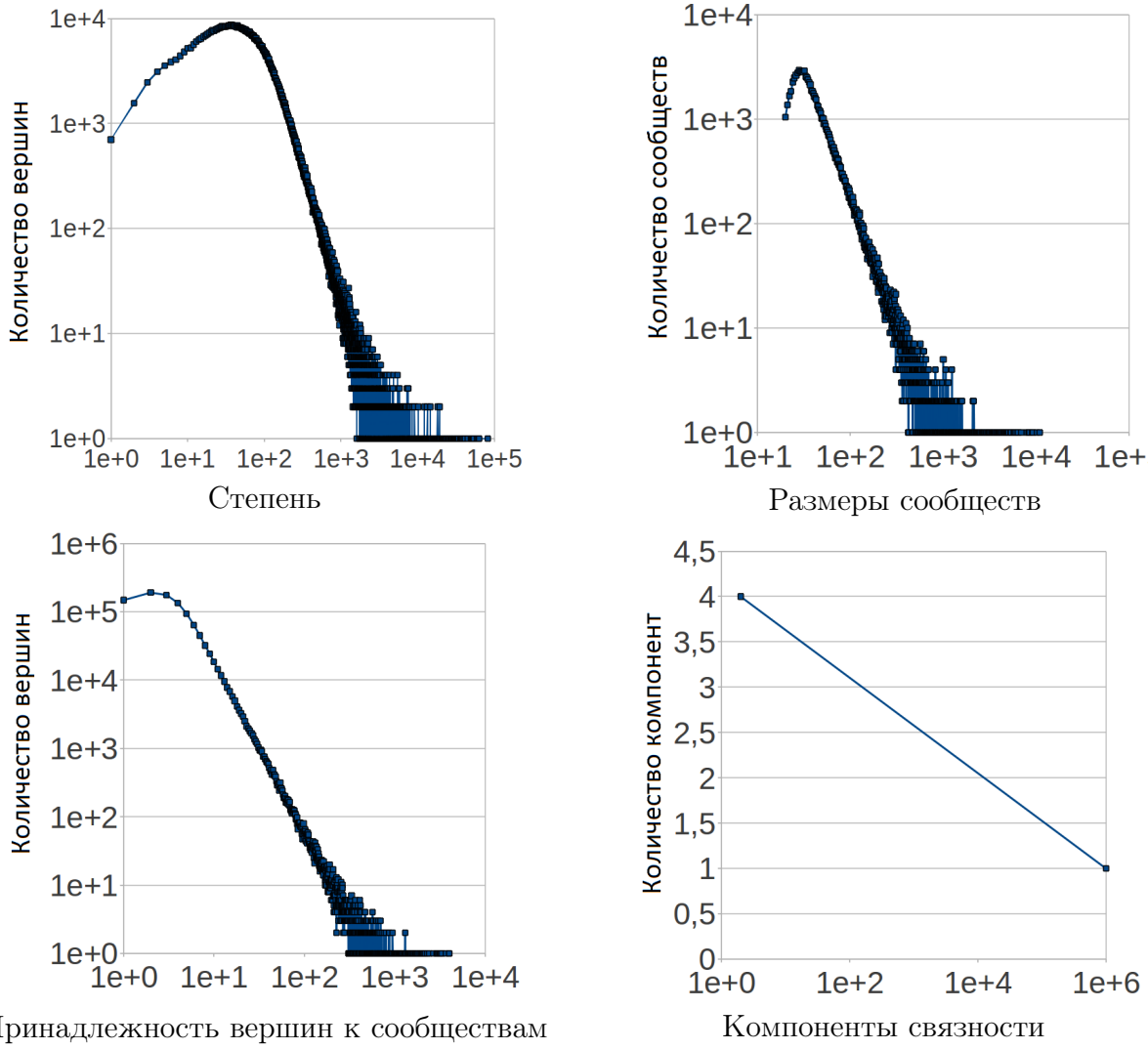


Рис. 3. Распределения степеней, размеров сообществ, принадлежностей вершин к сообществам и размеров компонент связности в сети СКВ с параметрами $N_1 = 10^6$, $\beta_1 = \beta_2 = 2.5$

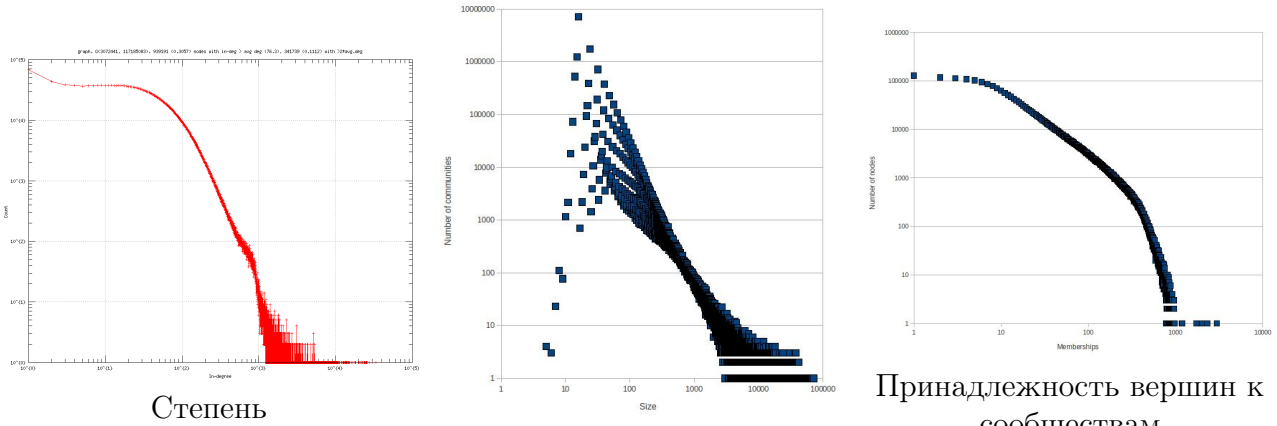
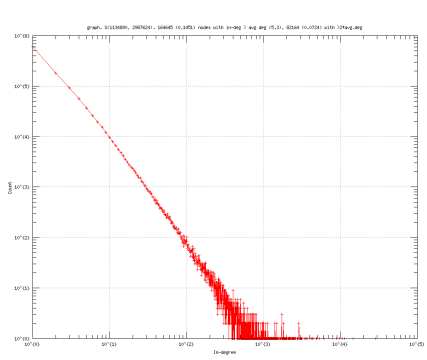
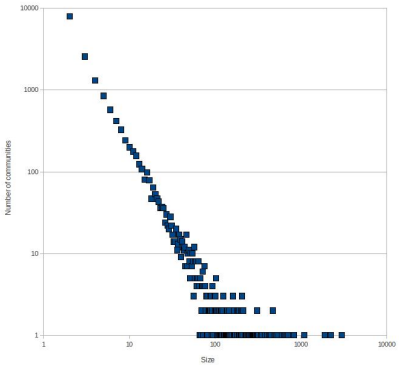


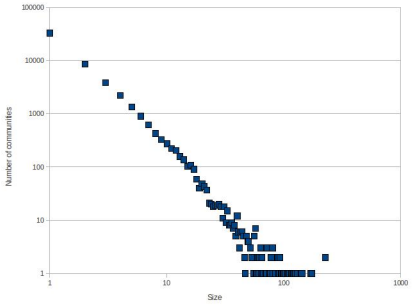
Рис. 4. Распределения степеней, размеров сообществ и принадлежностей вершин к сообществам для сети ORKUT



Степень

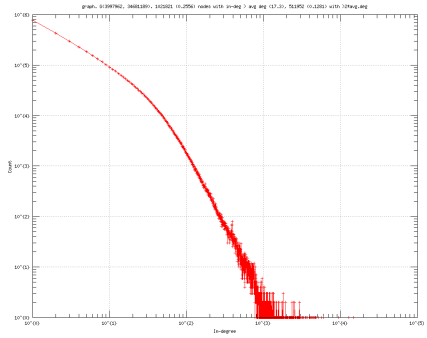


Размеры сообществ

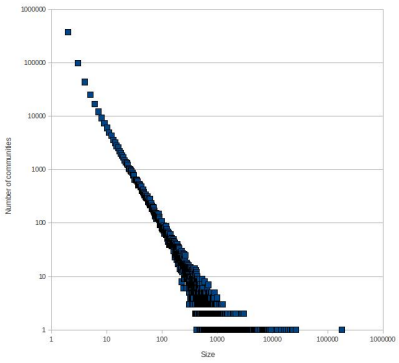


Принадлежность вершин к сообществам

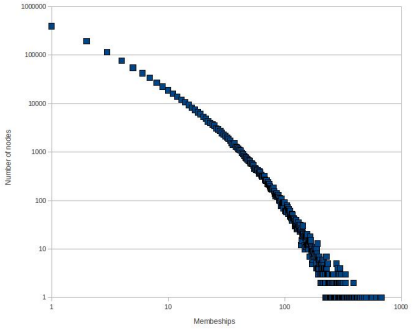
Рис. 5. Распределения степеней, размеров сообществ и принадлежностей вершин к сообществам для сети YouTube



Степень



Размеры сообществ



Принадлежность вершин к сообществам

Рис. 6. Распределения степеней, размеров сообществ и принадлежностей вершин к сообществам для сети LiveJournal

Выводы

В ходе работы был разработан, реализован и экспериментально исследован метод для распределенной генерации больших шаблонных сетей с реалистичными свойствами социальных графов и структурой сообществ.

Возможные направления дальнейшей работы:

— распределенное вычисление мер для сравнения покрытий сообществ;
— возможность контролировать коэффициент кластеризации;
— генерация иерархических, направленных и взвешенных сетей;
— генерация атрибутов пользователей с поддержкой свойств гомифилии в сообществах.

Литература

[1] *Yang J., Leskovec J.* Community-affiliation graph model for overlapping network community detection // *IEEE 12th Conference (International) on Data Mining*, 2012.

[2] *Yang J., Leskovec J.* Defining and evaluating network communities based on ground-truth // *ACM SIGKDD Workshop on Mining Data Semantics*, 2012.

[3] *Yang J., Leskovec J.* 2012. Structure and overlaps of communities in networks // *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2012.

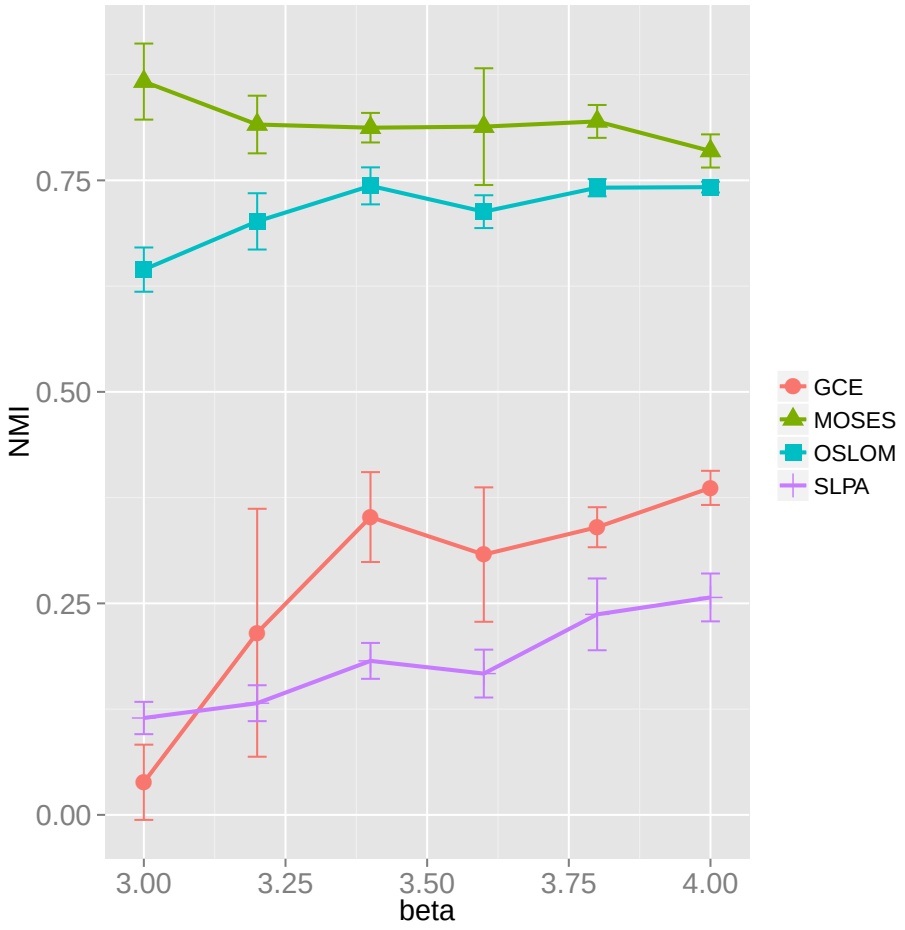


Рис. 7. Зависимость NMI от β_1 для различных алгоритмов определения структуры сообществ. Параметры СКВ: $N_1 = 10,000$, $x_{\min} = 1$, $m_{\min} = 2$, $x_{max} = m_{\max} = 1000$, $\alpha = 1$, $\gamma = 0.1$, $\varepsilon = 0$

[4] *Guillaume J.-L., Latapy M.* Bipartite graphs as models of complex networks // *Phys. A Stat. Mech. Appl.*, 2006. Vol. 371. P. 795.

[5] *Molloy M., Reed B.*. A critical point for random graphs with a given degree sequence // *Random Structures Algorithms*, 1995. Vol. 6. P. 161–180.

[6] *Aiello W., Chung F., Lu L.*. A Random Graph Model for massive graphs // *32nd Annual ACM Symposium on Theory of Computing*, 2000. P. 171–180.

[7] *Бернювский М.М., Кузюрин Н.Н.* Случайные графы, модели и генераторы безмасштабных графов // *Тр. Института системного программирования РАН*, 2012. Т. 22. С. 419–434.

[8] *Erdos P., Renyi A.* On the evolution of random graphs // *Bull. Inst. Int. Statist. Tokyo*, 1961. Vol. 38. P. 343–347.

[9] *Girvan M., Newman M.*. Community structure in social and biological networks // *Proc. Natl. Acad. Sci.*, 2002. Vol. 99.

[10] *Danon L., Díaz-Guilera A., Arenas A.* The effect of size heterogeneity on community identification in complex networks // *J. Stat. Mech. Theory Experiment*, 2006. Vol. 11.

[11] *Sawardecker E.N., Sales-Pardo M., Amaral L.A.N.* 2009. Detection of node group membership in networks with group overlap // *Eur. Phys. J. B*, 2009. Vol. 67. No. 3. P. 277–284.

[12] *Arenas A., Díaz-Guilera A., Pérez-Vicente C. J.* Synchronization reveals topological scales in complex networks // *Phys. Rev. Lett.*, 2006. Vol. 96. No. 11. P. 114102.

[13] *Fan Y., Li M., Zhang P., Wu J., Di Z.* Accuracy and precision of methods for community identification in weighted networks // *Phys. A Stat. Mech. Appl.*, 2007 Vol. 377. No. 1. P. 363–372.

[14] *Seshadhri C., Kolda T. G., Ali Pinar.* Community structure and scale-free collections of Erdos–Renyi graphs // *Phys. Rev. E*, 2012. Vil. 85. No. 5.

[15] *Singer K.* Random intersection graphs. PhD Thesis. Johns Hopkins University, 1995.

[16] *Deijfen M., Kets W.* Random intersection graphs with tunable degree distribution and clustering // *Probab. Eng. Inform. Sci.*, 2009. Vol. 23. No. 4. P. 615–623.

[17] *Lancichinetti A., Fortunato S.* Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities // *Phys. Rev.*, 2009. Vol. 80. No. 1.

[18] *Orman G. K., Labatut V.* The effect of network realism on community detection algorithms // *ASONAM*, 2010. P. 301–305.

[19] *Sala A., Gaito S., Rossi G.P., Zheng H., Zhao B. Y.* Revisiting degree distribution models for social graph analysis. *arXiv:1108.0027*. 2011.

[20] *Faloutsos M., Faloutsos P., Faloutsos C.* On power-law relationships of the Internet topology // *SIGCOMM*, 1999. P. 251–262.

[21] *Albert R., Jeong H., Barabasi A.-L.* Diameter of the world wide web // *Nature*, 1999. Vol. 401. P. 130–131.

[22] *Lancichinetti A., Fortunato S., Kertész J.* Detecting the overlapping and hierarchical community structure in complex networks // *New J. Phys.*, 2009. Vol. 11.

[23] *Leskovec J., Lang K. J., Dasgupta A., Mahoney M. W.*. Statistical properties of community structure in large social and information networks. 2008.

[24] *Райгородский А.М.* 2010. Модели случайных графов и их применения // *Труды МФТИ*, 2010. Vol. 2. No. 4. P. 130–140.

References

[1] *Yang J., Leskovec J.* 2012. Community-affiliation graph model for overlapping network community detection. *IEEE 12th Conference (International) on Data Mining.*

[2] *Yang J., Leskovec J.* 2012. Defining and evaluating network communities based on ground-truth. *ACM SIGKDD Workshop on Mining Data Semantics.*

[3] *Yang J., Leskovec J.* 2012. Structure and overlaps of communities in networks. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining.*

[4] *Guillaume J.-L., Latapy M.* 2006. Bipartite graphs as models of complex networks. *Phys. A Stat. Mech. Appl.* 371:795.

[5] *Molloy M., Reed B.* 1995. A critical point for random graphs with a given degree sequence. *Random Structures Algorithms* 6:161–180.

[6] *Aiello W., Chung F., Lu L.* 2000. A Random Graph Model for massive graphs. *32nd Annual ACM Symposium on Theory of Computing.* 171–180.

[7] *Bernovskiy M.M., Kuzyurin N.N.* 2012. On random graphs, models and scale-free graphs generators. *Proc. Institute for System Programming of RAS* 22:419–434.

[8] *Erdos P., Renyi A.* 1961. On the evolution of random graphs. *Bull. Inst. Int. Statist. Tokyo* 38:343–347.

[9] *Girvan M., Newman M.* 2002. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* 99.

[10] *Danon L., Díaz-Guilera A., Arenas A.* 2006. The effect of size heterogeneity on community identification in complex networks. *J. Stat. Mech. Theory Experiment* 11.

[11] *Sawardecker E.N., Sales-Pardo M., Amaral L.A.N.* 2009. Detection of node group membership in networks with group overlap. *Eur. Phys. J. B* 67(3):277–284.

[12] *Arenas A., Díaz-Guilera A., Pérez-Vicente C. J.* 2006. Synchronization reveals topological scales in complex networks. *Phys. Rev. Lett.* 96(11):114102.

[13] *Fan Y., Li M., Zhang P., Wu J., Di Z.* 2007. Accuracy and precision of methods for community identification in weighted networks. *Phys. A Stat. Mech. Appl.* 377(1):363–372.

[14] *Seshadhri C., Kolda T.G., Ali Pinar* 2012. Community structure and scale-free collections of Erdos–Renyi graphs. *Phys. Rev. E* 85(5).

[15] *Singer K.* 1995. Random intersection graphs. PhD Thesis. Johns Hopkins University.

[16] *Deijfen M., Kets W.* 2009. Random intersection graphs with tunable degree distribution and clustering. *Probab. Eng. Inform. Sci.* 23(4):615–623.

[17] *Lancichinetti A., Fortunato S.* 2009. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev.* 80(1).

[18] *Orman G. K., Labatut V.* 2010. The effect of network realism on community detection algorithms. *ASONAM.* 301–305.

[19] *Sala A., Gaito S., Rossi G. P., Zheng H., Zhao B. Y.* 2011. Revisiting degree distribution models for social graph analysis. *arXiv:1108.0027*

[20] *Faloutsos M., Faloutsos P., Faloutsos C.* 1999. On power-law relationships of the Internet topology. *SIGCOMM.* 251–262.

[21] *Albert R., Jeong H., Barabasi A.-L.* 1999. Diameter of the world wide web. *Nature* 401:130–131.

[22] *Lancichinetti A., Fortunato S., Kertész J.* 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys.* 11.

[23] *Leskovec J., Lang K. J., Dasgupta A., Mahoney M. W.* 2008. Statistical properties of community structure in large social and information networks.

[24] *Raigorodskiy A.M.* 2010. Models of random graphs and their application *Proc. MIPT* 2(4):130–140.