Наука и Образование
МГТУ им. Н.Э. Баумана
Сетевое научное издание
ISSN 1994-0408

Ссылка на статью:
// Наука и Образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2015. №9. С. 188–199.
DOI: **10.7463/0915.0811704**
Представлена в редакцию: 23.09.2015

УДК 519.178; 51-77

Выделение сообществ в социальных графах по множеству признаков с частичной информацией

Чесноков В. О.[1,*], **Ключарёв П. Г.**[1]

[*]v.o.chesnokov@yandex.ru

[1]МГТУ им. Н.Э. Баумана, Москва, Россия

В рамках данной статьи рассматривается метод получения сообществ (кластеров) социального графа ближайшего окружения пользователя социальной сети, т.е. графа, вершинами которого являются друзья пользователя, с диаметром, равным двум и с центром в вершине, соответствующей пользователю. В результате данной работы разработано программное обеспечение, позволяющее собрать данные из социальной сети и провести анализ соответствующего социального графа. Собраны социальные графы 2000 случайных пользователей социальной сети ВКонтакте. Произведена оценка качества кластеризации с помощью трех внутренних метрик: ожидаемой плотности, индекса силуэтов и нормализованной гамма-статистики Хуберта.

Ключевые слова: социальные сети; социальный граф; кластеризация; выделение сообществ

Введение

Важное место в анализе социальных сетей занимает проблема выделения сообществ — нахождение такого покрытия множества вершин социального графа, что между вершинами из одного подмножества «много» ребер, а между вершинами из разных — «мало» [1], или вершины в подмножествах расположены более плотно, чем в графе в целом [2]. Если подмножества не пересекаются, то они называются кластерами. В социальном графе люди из одного сообщества обычно имеют что-то общее между собой, например, общие интересы, место жительства, специальность и т.п. Кластерный анализ используется в информационных системах для обнаружения закономерности в данных.

Классический способ выделения сообществ состоит в применении одного из алгоритмов кластеризации, например максимизации модулярности или минимального остовного дерева, к социальному графу [3, 4, 5]. Основное отличие различных алгоритмов между собой — определение эвристики для получения подмножеств и определение понятий «много вершин внутри кластера», «мало вершин между кластерами» и плотности [2]. В таких алгоритмах при создании кластеров не используется информация о вершинах. Эта информация может использоваться в дальнейшем для именования кластеров, однако часто эта операция осуществляется человеком вручную.

Социальные графы социальных сетей характеризуются тем, что о пользователях известно достаточно много данных — сведения об образовании, местах работы, прохождения воинской службы, членстве в виртуальных сообществах по интересам (которые часто являются отражением реальных) и т.п. Сама социальная сеть побуждает пользователя указать как можно данных о себе. Существуют алгоритмы выделения сообществ, которые концетрируются на таких данных, например, алгоритмы, основанные на латентном размещении Дирихле [6]. Тем не менее лишь немногие из них используют как атрибуты вершин, так и связи между ними, одним из немногих является CESNA [9].

Все алгоритмы выделения сообществ подразумевают, что предоставлена полная информация о графе и атрибутах вершин. Однако не все пользователи социальной сети выставляют информацию о себе на публичное обозрение, и она может быть скрыта настройками приватности, что затрудняет ее анализ. В таком случае возникает проблема определения неуказанной или отсутсвующей информации.

Похожая проблема решается в работах [7] и [8]. В работе [7] предлагается жадный алгоритм выделения сообщества, соответствующего одному признаку, основанный на максимизации нормализованной проводимости. Поскольку алгоритм перебирает все вершины, не принадлежащие сообществу, для получения всех сообществ в худшем случае потребуется время, квадратично зависящее от числа вершин. Для тестирования алгоритма используется две собранные выборка из социальной сети Facebook — подграф, соответствующий университету Rice и подграф, соответствующий Новому Орлеану. В работе [8] предложен алгоритм, позволяющий получить значение одного атрибута пользователя по ограниченому числу обращений $maxFacts$ к другим узлам ограниченным обходом в ширину. Каждый атрибут может принимать конечное множество значений. Кроме того, в работе [8] описан модифицированный алгоритм, дополнительно использующий информацию о публикациях в сообществах и просмотрах публикаций пользователями. Для тестирования алгоритма используются данные из сети Facebook.

В данной работе предложен итеративный алгоритм, линейно зависящий от количества ребер. Данный алгоритм работает быстрее, чем предложенный в [7], и позволяет использовать уже полученные значения атрибутов, в отличие от алгоритма из [8]. Отметим также, что в [8] одна вершина может повлиять не более чем на $maxFacts$ ближайших вершин, в то время как в разработанном алгоритме влияние практически не ограничено. Для тестирования алгоритма используются графы ближайшего окружения случайно выбранных пользователей социальной сети Вконтакте.

1. Постановка задачи

Пусть имеется некоторый неориентированный невзвешенный граф $G'(V', E')$ с диаметром, равным двум, и центром в вершине u, каждая вершина которого может иметь до нескольких признаков (атрибутов), причем у каждой вершины p-й признак может нахо-

диться в одном из трех состояний: присутствует, отсутствует, неопределен. Пусть при этом невозможно различить два последних состояния. Графы, в которых значения всех признаков определены, рассматриваются, например, в [10].

Целью данной работы является разработка алгоритма, позволяющего выдвинуть предположение о состоянии неопределенных (отсутствующих) признаков для максимального количества вершин и выделить по этим признакам сообщества в графе. Таким образом, имея граф друзей пользователя, можно будет получить его разделение по сферам деятельности пользователя: сокурсники, коллеги и т.п.

2. Метод анализа социального графа

Поскольку для любой вершины v из V' существует ребро к u и это может затруднить выделение сообществ, далее будем рассматриваться граф $G(V, E)$, такой что $V = V' \setminus u$, $E = E' \setminus \{e|u \in e\}$. Пусть у каждой вершины $v \in V'$ есть набор признаков $attrs$.

Поскольку центром графа является пользователь u, логично будет предположить, что каждая его вершина v имеет хотя бы один общий признак с u. В таком случае можно упросить модель, рассматривая только такие признаки. Если же общего признака нет, то, скорее всего, он не указан. Если у вершины v несколько общих признаков с u, то выбор можно сделать исходя из количества вершин, которые обладают каждым из признаков. Если необходима общая картина с малым количеством больших кластеров, то выбирается признак, который имеют наибольшее количество вершин. В противном случае выбирается признак, который имеют наименьшее количество вершин. Выбранный признак $\mathcal{C}$ будем называть главным. Определение главных признаков производится посредством алгоритма 1, пример работы которого представлен на рис. 1.

Алгоритм 1. Определение главных признаков

$counts := \varnothing$
for v **from** V **do**
  for p **from** $v.attrs$ **do**
    if p **from** $counts$ **then**
      $counts[p] := counts[p] + 1$
    else
      $counts.add(p, 1)$
for v **from** V **do**
  $common := v.attrs \cap u.attrs$
  if $length(common) = 1$ **then**
    $v.attrs.add(\mathcal{C}, common[0])$
  else if $length(common) > 1$ **then**
    $v.attrs.add(\mathcal{C}, choose(v, common, counts))$
  else
    $v.attrs.add(\mathcal{C}, nil)$

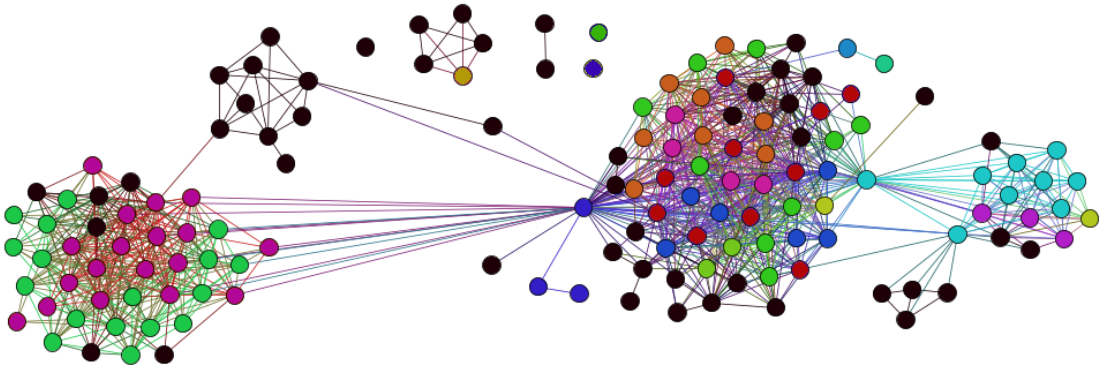

Рис. 1. Пример работы алгоритма определения главных признаков. Черным обозначены вершины, для которых главный признак неизвестен

Тогда для вершин, для которых не указан главный признак, его можно определить по признакам соседей по мажоритарному признаку, так как если у большинства соседей есть одинаковый признак, то и у самой вершины признак, скорее всего, будет таким же, что является следствием триадной структуры [11]. Отметим также, что для того, чтобы работал выбор по мажоритарному признаку нужно, чтобы хотя бы один сосед имел такой же признак, если у соседей они вообще есть (за это отвечает функция $choose$, алгоритм 2).

Алгоритм 2. Мажоритарное выделение сообществ по признакам соседей

$S := \{v \in V | v.attrs[\mathcal{C}] = nil\}$
for v **from** S **do**
  $\mathcal{N} := neighbours(v)$
  $p := most\_common(\mathcal{N})$
  $v.attrs[\mathcal{C}] := p$

В этом алгоритме не учтена ситуация, когда присутствуют связные компоненты, не содержащие хотя бы один известный признак или когда большинство соседей не имеют признака. Кроме того, возможна ситуация, когда на начальном этапе был выбран «неправильный» главный признак. Должна присутствовать возможность изменить его. Эти проблемы можно решить, если сделать алгоритм итеративным и не учитывать вершины без признака при выборе признака для текущей (см. алгоритм 3 и пример работы на рис. 2). Коэффициент α определяет относительное количество вершин из $\mathcal{N}'$, которые должны обладать признаком p, чтобы изменить главный признак $\mathcal{C}$ текущей вершины v на p. Очевидно, что величина коэффициента α лежит в полуинтервале $[0; 1)$.

На каждой итерации алгоритма 3 производится обход всех вершин. Для каждой вершины v определяется множество ее соседей $\mathcal{N}$ и множество соседей $\mathcal{N}'$ с присутствующим главным признаком. Затем производится определение признака p, который есть у большинства вершин из $\mathcal{N}'$. Если количество соседей превышает пороговое, то у текущей вершины главный признак изменяется на p и вершина удаляется из множества вершин S с неопределенным признаком. Алгоритм прекращает свою работу, либо когда множество S опустеет,

Алгоритм 3. Модифицированный алгоритм

$changed := \textbf{true}$
$S := \{v \in V | v.attrs[\mathcal{C}] = nil\}$
while $iter < iter_{max}$ **and** $changed$ **and** $S \neq \varnothing$ **do**
    $iter := iter + 1$
    $changed := \textbf{false}$
    for v **from** V **do**
        $\mathcal{N} := neighbours(v)$
        $\mathcal{N}' := neighbours(v) \setminus \{s | s.attrs[\mathcal{C}] = nil\}$
        if $\mathcal{N}' = \varnothing$ **then**
            continue
        $counts := \varnothing$
        for t **from** $\mathcal{N}'$ **do**
            $p := t.attrs[\mathcal{C}]$
            if p **from** $counts$ **then**
                $counts[p] := counts[p] + 1$
            else
                $counts.add(p, 1)$
        $p := x : count[x] \to max$
        if $counts[p] > \alpha|\mathcal{N}'|$ **then**
            $v.attrs[\mathcal{C}] := p$
            $changed := \textbf{true}$
            $S := S \setminus \{v\}$

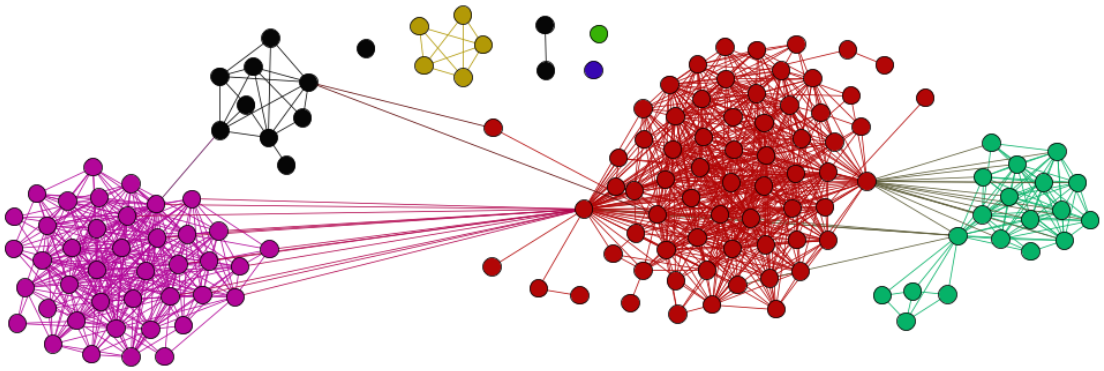

Рис. 2. Пример работы разработанного алгоритма. Черным обозначены вершины, для которых главный признак неизвестен

либо когда на итерации не произошло изменений, либо при достижении максимального числа итераций. Вершины, которые имеют одинаковый главный признак, будем считать принадлежащими одному сообществу. Таким образом, если в кластере преобладает какой-то главный признак, то все вершины внутри кластера его приобретут из-за того, что у них много связей между собой, а вершины вне кластера — нет, так как число связей между кластерами меньше, чем внутри кластеров.

На каждой итерации во внутреннем цикле рассматриваются все соседи вершины не более чем по одному разу. Во внешнем цикле производится обход всех вершин, следовательно

каждое ребро будет посещено не более чем 2 раза. Таким образом, сложность каждой интерации можно оценить как $O(|E|)$, а сложность всего алгоритма — $O(|E|I)$, где I — количество итераций.

3. Оценка качества кластеризации

Для оценки качества полученной кластеризации $\mathcal{C}$ использованы три нормированные метрики. Использование нормированных метрик позволяет: во-первых, сравнить качество двух разбиений с разным количеством кластеров; во-вторых, оценить качество разбиения без сравнения с другими возможными разбиениями; в-третьих, оценить качество работы алгоритма выделения сообществ на выборке из нескольких графов через среднее значение метрик.

Ожидаемая плотность показывает, насколько кластеры «плотнее», чем сам граф [12]:

$$\overline{\rho}(\mathcal{C}) = \sum_{i=1}^{k} \frac{|V_i|}{|V|} \frac{\omega(G_i)}{|V_i|^{\vartheta}},$$

где $\omega(G) = |V| + \sum\limits_{e \in E} \omega(e)$ — вес графа (поскольку граф невзвешенный, то $\omega(e) = 1$); ϑ — плотность всего графа, т.е. $\omega(G) = |V|^{\vartheta}$; $G_i(V_i, E_i)$ — подграф, соответствующий i-му кластеру. Очевидно, что значение $\overline{\rho}(\mathcal{C})$ при хорошем качестве кластеризации стремится к 2, что соответствует кластерам с высокой плотностью, причем чем выше значение, тем кластеризация лучше.

Индекс силуэтов (silhouette index) показывает, насколько хорошо каждый объект (вершина) лежит в своем кластере и вычисляется как [13, 14]

$$S = \frac{1}{|V|} \sum \frac{b_i - a_i}{\max(a_i, b_i)},$$

где a_i — мера схожести i-й вершины со своим кластером; b_i — мера схожести i-й вершины со ближайшим к ней (но не своим) кластером. В качестве меры схожести вершины с кластером взято среднее расстояние от вершины до вершин кластера. Для каждой вершины индекс принимает значения в интервале от -1 до 1. Значения, близкие к -1, показывают, что вершина была отнесена к неверному кластеру. Значения, близкие к 1 показывают, что кластер определен хорошо. Положительные значения, близкие к нулю, показывают, что вершина находится близко к границе кластера (и, соответственно, близко к границе ближайшего кластера). Средний индекс имеет значения в диапазоне от -1 до 1, причем чем ближе значение к 1, тем лучше качество полученных кластеров.

Нормализованная Hubert's Γ Statistic представляет собой поэлементную корреляцию двух матриц [14]:

$$\hat{\Gamma} = \frac{\frac{1}{M} \sum\limits_{i=1}^{|V|-1} \sum\limits_{j=1}^{|V|} (X(i,j) - \mu_X)(Y(i,j) - \mu_Y)}{\sigma_X \sigma_Y},$$

где X — матрица расстояний; μ_X — среднее расстояние; Y — матрица расстояний между центрами кластеров; μ_Y — среднее расстояние между центрами кластеров; σ_X, σ_Y — среднеквадратичные отклонения для соответствующих матриц; $M = |V|(|V| - 1)/2$ — общее количество вершин в верхней треугольной матрице (так как графы неориентированные, рассматривается только она). При этом $\hat{\Gamma} \in [-1; 1]$ и большие значения подразумевают лучшую структуру кластеров.

4. Метод доступа к социальным связям социальной сети

Для получения социальных графов социальной сети использованы подходы, аналогичные описанным в [15]. Данные социального графа собраны через открытое публичное API, расположенный по адресу http://api.vk.com, методы *group.getMembers* и *friends.get*. К сожалению, не все данные доступны через API, поэтому информацию о пользователях пришлось собирать путем парсинга HTML-страниц профилей и поисковой выдачи ВКонтакте.

Социальный граф пользователя собирался по следующему алгоритму. Сначала был получен список друзей пользователя. Затем для каждого пользователя был получен список его друзей, после чего обход графа прекращался. После обхода графа собиралась информация о группах, образовании и местах работы пользователя. Наконец, для каждой группы, школы, кафедры и места работы собирался список участников, обучавшихся и работавших соответственно.

5. Программное обеспечение

Для решения задачи сбора тестовой выборки доработано программное обеспечение для сбора, разработанное в [15]. Данные скачивались в несколько потоков через анонимную сеть Tor, предварительно обрабатывались и записывались в реляционную базу данных PostgreSQL. Распределение задач так же осуществлялось с помощью БД ключ-значение Redis. Из БД граф экспортировался скриптом в формат GEXF для возможности просмотра в Gephi [16]. Еще один скрипт обновлял GEXF, добавляя разбиение на сообщества по разработанному алгоритму.

6. Результаты работы

Для тестовой выборки собраны социальные графы 2000 случайных пользователей и произведена их обработка со значениями коэффициента α в интервале от 0.1 до 0.9 с шагом 0.1. Для всех графов подсчитаны оценки качества кластеризации и подсчитаны средние значения. Кроме того, рассчитано среднее количество количество вершин с отсутствующими признаками до (равное 66.7%) и после применения разработаного алгоритма. Результаты работы представлены в табл. 1.

Алгоритм показал очень высокое значение Hubert's Γ Statistic, близкое к единице, для всех значений коэффициента α. При уменьшении α растет плотность кластеров, что видно

Таблица 1

Результаты работы. Средние значения оценок качества кластеризации

α	кол-во вершин без главного признака, %	S	$\overline{\rho}$	$\hat{\Gamma}$
0.1	25.6	0.139	1.527	0.955
0.2	36.9	0.159	1.483	0.953
0.3	46.9	0.186	1.387	0.947
0.4	55.3	0.194	1.295	0.946
0.5	61.4	0.192	1.227	0.945
0.6	62.4	0.195	1.203	0.945
0.7	63.5	0.196	1.184	0.945
0.8	63.9	0.197	1.175	0.945
0.9	64.0	0.197	1.172	0.945

по значению метрики $\overline{\rho}$, однако они становятся более размытыми, поскольку уменьшается значение индекса силуэтов. Значения индекса силуэтов получились положительным, но невысоким. Это можно объяснить сравнительно небольшим размером кластеров в тестовой выборке, т.е. многие вершины были интерпретированы как лежащие близко к границе кластеров. Значения $\overline{\rho}$ получились больше 1, следовательно, вершины в кластерах расположены плотнее, чем в целом в графах.

Заключение

В данной работе продемонстрирован метод выделения сообществ в социальном графе на основе признаков вершин при частично отсутствующих признаках. Алгоритм опробован на тестовой выборке из 2000 графов, собранной из социальной сети ВКонтакте, продемонстрировав неплохие показатели внутренних оценок качества кластеризации.

Дальнейшее улучшение алгоритма может быть направлено на определение нечетких сообществ, т.е. таких, где вершина может принадлежать нескольким сообществам одновременно. Кроме того, необходимо повысить верифицируемость результата. Для этого необходимо построить модель социального графа ближайшего окружения пользователя и оценить качество выделения сообществ с использованием внешних метрик. При наличии модели можно будет определить порог отношения вершин с отсутствующими признаках, препятствующий корректному выделению сообществ алгоритмом.

Список литературы

1. Fortunato S. Community detection in graphs // Physics Reports. 2010. Vol. 486, no. 3-5. P. 75–174. DOI: 10.1016/j.physrep.2009.11.002
2. Mucha P.J., Richardson T., Macon K., Porter M.A., Onnela J.P. Community structure in time-dependent, multiscale, and multiplex networks // Science. 2010. Vol. 328, no. 5980. P. 876–878. DOI: 10.1126/science.1184819

3. Newman M.E.J., Girvan M. Finding and evaluating community structure in networks // Physical Review E. 2004. Vol. 69. Art. no. 026113. DOI: 10.1103/PhysRevE.69.026113
4. Schaeffer S.E. Graph clustering // Computer Science Review. 2007. Vol. 1, no. 1. P. 27–64. DOI: 10.1016/j.cosrev.2007.05.001
5. Scott J. Social network analysis: A handbook. 2nd ed. London: SAGE, 2000.
6. Blei D.M., Ng A.Y., Jordan M.I. Latent Dirichlet Allocation // Journal of Machine Learning Research. 2003. Vol. 3, P. 993–1022.
7. Mislove A., Viswanath B., Gummadi K.P., Druschel P. You are who you know: inferring user profiles in online social networks // Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, New York, NY, USA, February 03–06, 2010. New York: ACM, 2010. 452 p. P. 251–260. DOI: 10.1145/1718487.1718519
8. Dougnon R., Fournier-Viger P., Nkambou R. Inferring User Profiles in Online Social Networks Using a Partial Social Graph // Advances in Artificial Intelligence / ed. by D. Barbosa, E. Milios. Springer International Publishing, 2015. P. 84–99. DOI: 10.1007/978-3-319-18356-5_8 (Ser. Lecture Notes in Computer Science; vol. 9091).
9. Yang J., McAuley J.J., Leskovec J. Community Detection in Networks with Node Attributes // 2013 IEEE 13th International Conference on Data Mining (ICDM). IEEE Publ., 2013. P. 1151–1156. DOI: 10.1109/ICDM.2013.167
10. Mcauley J., Leskovec J. Discovering Social Circles in Ego Networks // ACM Transactions on Knowledge Discovery from Data. 2014. Vol. 8, no. 1. Art. no. 4. DOI: 10.1145/2556612
11. Granovetter M. The Strength of Weak Ties // American Journal of Sociology. 1973. Vol. 78, no. 6. P. 1360–1380.
12. Stein B., zu Eissen S.M., Wißbrock F. On Cluster Validity and the Information Need of Users // Proceedings of the 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA). Benalmadena, Spain, September 8-10, 2003. ACTA Press, 2003. P. 216–221.
13. Rousseeuw P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis // Journal of Computational and Applied Mathematics. 1987. Vol. 20. P. 53–65. DOI: 10.1016/0377-0427(87)90125-7
14. Zaki M.J., Meira Jr.W. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press, 2014. 562 p.
15. Ключарёв П.Г., Чесноков В.О. Исследование спектральных свойств социального графа сети LiveJournal // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2013. № 9. С. 391–400. DOI: 10.7463/0913.0603441
16. Gephi — the Open Graph Viz Platform: webcite. Режим доступа: http://gephi.github.io (дата обращения 20.09.2015).

Science&Education of the Bauman MSTU
Electronic journal
ISSN 1994-0408

Science and Education of the Bauman MSTU, 2015, no. 9, pp. 188–199.
DOI: **10.7463/0915.0811704**
Received: 23.09.2015

Social Graph Community Differentiated by Node Features with Partly Missing Information

Chesnokov V. O.[1,*], **Klyucharev P. G.**[1]

[*]v.o.chesnokov@yandex.ru

[1]Bauman Moscow State Technical University, Russia

Keywords: social networks, social graph, clustering, community detection

This paper proposes a new algorithm for community differentiation in social graphs, which uses information both on the graph structure and on the vertices. We consider user's ego-network i.e. his friends, with no himself, where each vertex has a set of features such as details on a workplace, institution, etc. The task is to determine missing or unspecified features of the vertices, based on their neighbors' features, and use these features to differentiate the communities in the social graph. Two vertices are believed to belong to the same community if they have a common feature. A hypothesis has been put forward that if most neighbors of a vertex have a common feature, there is a good probability that the vertex has this feature as well. The proposed algorithm is iterative and updates features of vertices, based on its neighbors, according to the hypothesis. Share of neighbors that form a majority is specified by the algorithm parameter. Complexity of single iteration depends linearly on the number of edges in the graph.

To assess the quality of clustering three normalized metrics were used, namely: expected density, silhouette index, and Hubert's Gamma Statistic. The paper describes a method for test sampling of 2.000 graphs of the user's social network "VKontakte". The API requests addressed "VKontakte" and parsing HTML-pages of user's profiles and search results provided crawling. Information on user's group membership, secondary and higher education, and workplace was used as features. To store data the PostgreSQL DBMS was used, and the gexf format was used for data processing. For the test sample, metrics for several values of algorithm parameter were estimated: the value of index silhouettes was low (0.14-0.20), but within the normal range; the value of expected density was high, i.e. 1.17-1.52; the value of Hubert's gamma statistic was 0.94–0.95 that is close to the maximum. The number of vertices with no features was calculated before and after applying the algorithm. With two-third of vertices before using the algorithm their number has fallen to 25-64% depending on algorithm parameter after applying it. The proposed algorithm can be used for clustering the social graphs, but it should be modified to differentiate overlapping communities.

References

1. Fortunato S. Community detection in graphs. *Physics Reports*, 2010, vol. 486, no. 3-5, pp. 75–174. DOI: 10.1016/j.physrep.2009.11.002
2. Mucha P.J., Richardson T., Macon K., Porter M.A., Onnela J.P. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 2010, vol. 328, no. 5980, pp. 876–878. DOI: 10.1126/science.1184819
3. Newman M.E.J., Girvan M. Finding and evaluating community structure in networks. *Physical Review E*, 2004, vol. 69, art. no. 026113. DOI: 10.1103/PhysRevE.69.026113
4. Schaeffer S.E. Graph clustering. *Computer Science Review*, 2007, vol. 1, no. 1, pp. 27–64. DOI: 10.1016/j.cosrev.2007.05.001
5. Scott J. *Social network analysis: A handbook. 2nd ed*. London, SAGE, 2000.
6. Blei D.M., Ng A.Y., Jordan M.I. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 2003, vol. 3, pp. 993–1022.
7. Mislove A., Viswanath B., Gummadi K.P., Druschel P. You are who you know: inferring user profiles in online social networks. *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, New York, NY, USA, February 03–06, 2010*. New York, ACM, 2010, pp. 251-260. DOI: 10.1145/1718487.1718519
8. Dougnon R., Fournier-Viger P., Nkambou R. Inferring User Profiles in Online Social Networks Using a Partial Social Graph. In: Barbosa D., Milios E., eds. *Advances in Artificial Intelligence*. Springer International Publishing, 2015, pp. 84–99. DOI: 10.1007/978-3-319-18356-5_8 (Ser. Lecture Notes in Computer Science; vol. 9091).
9. Yang J., McAuley J.J., Leskovec J. Community Detection in Networks with Node Attributes. *2013 IEEE 13th International Conference on Data Mining (ICDM)*. IEEE Publ., 2013, pp. 1151–1156. DOI: 10.1109/ICDM.2013.167
10. Mcauley J., Leskovec J. Discovering Social Circles in Ego Networks. *ACM Transactions on Knowledge Discovery from Data*, 2014, vol. 8, no. 1, art. no. 4. DOI: 10.1145/2556612
11. Granovetter M. The Strength of Weak Ties. *American Journal of Sociology*, 1973, vol. 78, no. 6, pp. 1360–1380.
12. Stein B., zu Eissen S.M., Wißbrock F. On Cluster Validity and the Information Need of Users. *Proceedings of the 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA), Benalmadena, Spain, September 8-10, 2003*. ACTA Press, 2003, pp. 216–221.
13. Rousseeuw P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 1987, vol. 20, pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7

14. Zaki M.J., Meira Jr.W. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press, 2014. 562 p.

15. Klyucharev P.G., Chesnokov V.O. Study of the spectral properties of LiveJournal's social graph. *Nauka i obrazovanie MGTU im. N.E. Baumana = Science and Education of the Bauman MSTU*, 2013, no. 9, pp. 391–400. DOI: 10.7463/0913.0603441 (in Russian).

16. Gephi — the Open Graph Viz Platform: webcite. Available at: http://gephi.github.io, accessed 20.09.2015.