Математические структуры и моделирование
2010, вып. 21, с. 11–16

УДК 519.17

ПОИСК СВЯЗАННЫХ СТРУКТУР ВО ВЗВЕШЕННЫХ ГРАФАХ

С.В. Белим, А.В. Сорокин

В данной работе проводится исследование алгоритмов выявления связанных структур во взвешенных графах. На основе компьютерного эксперимента сравниваются алгоритмы полного перебора и «жадный» алгоритм.

Введение

В настоящее время моделирование многих систем опирается на теорию графов, то есть множества вершин и связей между ними. Особенно большое количество моделей разработано для социальных сетей [1–3], компьютерных сетей [4, 5], биологических систем [3] и сетей обслуживания [6]. Ранние исследования были ориентированы на изучение статистических свойств, вытекающих из топологии сети. Такие свойства получили название «эффекты малых сетей», так как исследуемые сети имели малые расстояния между вершинами [7,8], обычно растущие логарифмически с увеличением количества вершин.

Во многих сетях проявляется свойство образования связанных структур (community structure). Это свойство иногда называют кластеризацией, однако мы не будем пользоваться этим термином, так как его принято использовать несколько в ином смысле. Термин «связанные структуры» первоначально появился при исследовании социальных сетей и в дальнейшем получил распространение на другие аналогичные сети. Мы, по аналогии с работами [9–12], под связанными структурами будем понимать подмножество вершин, связанных между собой сильнее, чем с остальными вершинами графа. В данном определении недостаточно четко выглядит понятие «сильнее связаны». В различных сетях для характеристики величины связи вводятся разные функции.

На сегодняшний день разработано несколько подходов к поиску связанных структур:

1. *Полный перебор* состоит в выделении подмножеств вершин и вычислении функции силы связности соответствующей структуры. Этот подход имеет одно преимущество – алгоритм является точным. Однако, как легко понять, сложность алгоритма растет экспоненциально с увеличением числа вершин, в силу чего алгоритм становится не пригодным для достаточно больших графов.

Омский государственный университет им. Ф.М. Достоевского.
E-mail: sbelim@mail.ru

2. *Иерархическая кластеризация* дает более быстрый алгоритм, однако не всегда правильный ответ [11,12]. Метод состоит в том, что сначала вычисляется вес связи для каждой пары вершин. Затем строится многоуровневое дерево, листьями которого являются исходные вершины. На первом шаге построения дерева появляется новая вершина, связанная дугами с двумя наиболее сильно связанными между собой вершинами. Далее в исходном графе образуется стяжка двух вершин, выделенных ранее при построении дерева. Таким образом в графе две вершины заменяются одной, которая наследует все связи с остальными вершинами. То есть если у одной из вершин, попавших в стяжку, была дуга к какой-либо вершине, то она будет и в новом графе. Далее задача образования стяжки из двух вершин решается в новом графе и так далее. Полученное многоуровневое дерево в социологии получило название дендрограммы [12].

3. Генетические алгоритмы [13, 14] были разработаны именно для больших графов и используют метод, аналогичный построению многоуровневого дерева. Вероятность правильного разбиения графа на связанные структуры при этом, как и следовало ожидать, еще ниже, чем во втором случае.

Целью данной статьи является построение и исследование с помощью компьютерного эксперимента «жадного» алгоритма поиска связанных структур.

1. Описание графов

Начнем с представления графов, на которых будут выявляться связанные структуры. Будем считать, что в графе заданы веса как ребер, так и вершин. Такие графы могут быть заданы с помощью матрицы смешения E, предложенной в работе [10]. Элементы матрицы смешения задаются следующим образом. Диагональный элемент E_{ii} показывает вес вершины с номером i (v_i). Элемент E_{ij} ($i \neq j$) показывает величину связи вершины v_i с вершиной v_j. Как показано в работе [10], более удобным является приведенный вид матрицы смешения $e = E/m$, где $m = \sum E_{ij}$. В приведенной матрице смешения элемент e_{ij} показывает долю веса заданного ребра в общем весе графа. В дальнейшем под матрицей смешения будет пониматься именно приведенный вид. Легко увидеть, что $\sum e_{ij} = 1$.

Способы задания матрицы смешения могут быть разными и зависят от алгоритма определения величины связи вершин. Так, в работе [10] в качестве величины связи двух вершин используется вес связывающего их ребра. При этом вес вершины можно трактовать как суммарный вес петель с концами в этой вершине. В работе [15] в качестве величины связи двух вершин используется сумма весов всех путей в графе, ведущих из одной вершины в другую с весовыми коэффициентами, зависящими от длины пути. Влияние способа построения матрицы смешения на выявление связанных структур до сих пор остается не до конца исследованным.

2. Мера связности вершин

Как уже было сказано во введении, для выявления связанных структур необходимо определить некоторую функцию от элементов матрицы смешения, численно определяющую силу связности. Будем считать, что такая функция задана при постановке задачи, и будем обозначать ее $Q(e)$ и называть мерой связности.

В ряде работ было предложено несколько функций меры связности вершин.

1. Метод парных корреляций был предложен в работе [16] и состоит в вычислении коэффициентов Жаккарда и индексов Ранда для пар вершин, взятых из различных подграфов исходного графа.

2. Метод кластеризации, основанный на метрике Донгена [17].

3. Теоретико-информационный подход [18, 19], рассматривающий меру связности как интенсивность обмена информацией. Далее на основе вычисления взаимной энтропии выделяются связанные структуры, внутри которых обмен информацией происходит интенсивнее, чем с остальными вершинами.

4. Метод Ньюмана, исследованный в работах [10–12]. В качестве меры связанности используется величина

$$Q(e) = \sum e_{ii} - \sum a_i b_i,$$

где

$$a_i = \sum e_{ij}, b_i = \sum e_{ji}.$$

Выбор функции меры связанности графа зависит от постановки задачи. В данной работе в качестве функции $Q(e)$ мы будем использовать выражение, предложенное в работах Ньюмана.

3. Задача поиска связанных структур

Определим более строго процедуру образования связанной структуры в графе. Начнем с алгоритма образования стяжек как преобразования графа G в граф $G1$. Выделим в графе G подграф G' и заменим все входящие в него вершины одной вершиной, при этом вершины подграфа $G \setminus G'$ остаются неизменными. Образованная вершина связана дугами с теми вершинами графа $G1$, с которыми были связаны вершины, вошедшие в стяжку. Вес вершины, вошедшей в стяжку, равен сумме весов вершин и дуг, вошедших в стяжку. При этом, если граф не ориентированный, при образовании стяжки каждую дугу заменяем двумя дугами с противоположной ориентацией и одинаковым весом.

Под связанной структурой будем понимать подграф исходного графа, который при образовании из него стяжки максимизирует меру связанности графа. Будем различать две задачи выявления связанных структур — частную и общую.

Частная задача:
Поиск связанной структуры в исходном графе, включающей в себя заданную вершину.

Общая задача:
Выявление всех связанных структур в графе.

В данной работе задачи выявления связанных структур в графах решались с помощью компьютерного эксперимента. Рассматривались взвешенные графы с различным количеством вершин. Для каждого размера графа случайным образом генерировалось по 100 матриц смешения. После чего решалась задача выявления связанных структур.

Достаточно сложным является вопрос: является ли частная задача частью общей задачи? То есть всегда ли связанная структура, образованная при решении частной задачи, сохранится при решении общей задачи. Для проверки этой гипотезы был проведен компьютерный эксперимент. Последовательно для всех вершин графа решалась частная задача. Затем для всего графа решалась общая задача и проверялось, все ли связанные структуры, полученные при решении частных задач, присутствуют в решении общей задачи.

4. «Жадный» алгоритм поиска связанных структур

Несложно заметить, что точный алгоритм, построенный на полном переборе, имеет экспоненциальную сложность. Поэтому возникает задача построения других алгоритмов, дающих точное решение задачи либо решение, близкое к точному.

Рассмотрим следующий «жадный» алгоритм решения частной задачи для вершины v.

1. Ищем вершину v_1, связанную с v дугой, которая при образовании стяжки с v дает наибольшее увеличение меры связности Q.

2. Образуем стяжку из вершин v_1 и v, обозначаем ее через v и переходим к пункту 1.

3. Пункты 2 и 3 повторяем до тех пор, пока существуют вершины, стяжка с которыми увеличивает Q.

Для определения эффективности «жадного» алгоритма был проведен компьютерный эксперимент решения частной задачи поиска связанных структур с помощью «жадного» алгоритма и точного алгоритма (перебора).

«Жадный» алгоритм решения общей задачи поиска связанных структур выглядит следующим образом.

1. Выбираем одну из вершин графа и решаем для нее частную задачу поиска связанных структур.

2. В графе, полученном в результате стяжки связанной структуры, найденной в первом пункте, выбираем вершину, отличную от выбранной ранее, и для нее решаем частную задачу поиска связанных структур.

3. Повторяем пункт 2 до тех пор, пока все вершины не будут определены в связанные структуры (связанные структуры могут содержать и одну вершину).

На рисунке приведены сравнительные результаты компьютерного эксперимента. Тест 1 показывает процент случаев, в которых решение частной задачи присутствует в решении общей задачи для матриц различного размера. Тест 2 демонстрирует процент совпадения решений частной задачи поиска связанных структур «жадным» алгоритмом и полным перебором.

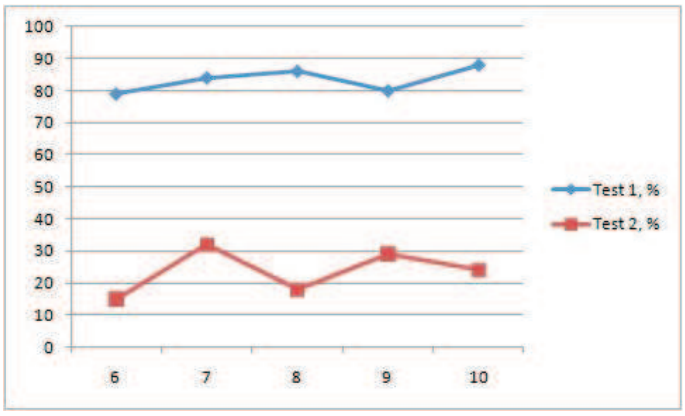

Рис. 1. Сравнение результатов «жадного» алгоритма и точного решения для общей задачи

5. Обсуждение результатов

Результаты компьютерного эксперимента показывают, что объединение в связанные структуры, выгодное одной вершине, не всегда выгодно при полном разбиении графа на связанные структуры (тест 1). Отсюда следует, что решение общей задачи с помощью «жадного» алгоритма не всегда будет совпадать с точным решением.

«Жадный» алгоритм не всегда приводит к точному решению частной задачи выявления связанных структур (тест 2). Однако учитывая, что в случаях несовпадения решений отклонение меры связанности Q «жадного» алгоритма от точного алгоритма составляет не более 5%, можно считать, что «жадный» алгоритм дает хороший результат, не уступающий в точности другим приближенным методам [19].

Литература

1. Wasserman S., Faust K. Social Network Analysis. Cambrige.: Cambrige University Press, 1994.
2. Scott J. Social Network Analysis: A Handbook. London.: Sage Publication, 2000.
3. Watts D. J., Strogatz S. H. Collective dynamics of 'small-world' networks // Nature. 1998. V. 393. P. 440–442.
4. Faloutsos M., Faloutsos P., Faloutsos C. On power-law relationships of the internet topology // Computer Communication Review. 1999. V. 29. P. 251–262.
5. Albert R., Jeong H., Barabasi A.-L. Diameter of the world-wide web // Nature. 1999. V. 401. P. 130–131.
6. Newman M. E. J. The structure of scientific collaboration network // Proc. Natl. Acad. Sci. USA. 2001. V. 98. P. 404–409.
7. Pool I., Kochen M. Contact and influence // Social network. 1978. V. 1. P. 1–48.
8. Milgram S. The small world problem // Psychology Today. 1967. V. 2. P. 60–67.
9. Girvan M., Newman M. E. J. Community structure in social and biological networks // arXiv:cond-mat/0112110v1. (2001)

10. Newman M. E. J., Girvan M. Finding and evaluting community structure in networks // arXiv:cond-mat/0308217v1. (2003)
11. Newman M. E. J. Fast algorithm for detecting community structure in networks // arXiv:cond-mat/0309580v1. (2003)
12. Newman M. E. J. Mixing patterns in networks // Phys. Rev. E. 2003. V. 67. P. 026126-1–026126-13.
13. Berryman M. J., Allison A., Abbott D. Optimizing genetic algorithm strategies for evolving networks // arXiv:cs/0404019v1. (2004)
14. Tasgin M., Herdagdelen A., Bingol H. Community detection in complex network using genetic algorithms // arXiv:0711.0491v1. (2007)
15. Leicht E. A., Holme P., Newman M. E. J. Vertex similarity in networks // arXiv:physics/0510143v1. (2005)
16. Meilia M. Comparing clasterings-an information based distance // Journal of Multivariate Analysis. 2007. V. 98. P. 873–895.
17. Dongen S. V. Performance criteria for graph clastering and Markov cluster experiments. National Research Institute for Mathematics and Computer Science in the Netherlands, 2000.
18. Meilia M. Comparing clusterings: an axiomatic view. // ICML '05: Proceedings of 22nd International Conference on Machine Learning, New York:ACM Press, 2005. P. 577–584.
19. Meilia M. Comparing clasterings // Technical report, University of Washington, 2002.