электронное научно-техническое издание
НАУКА и ОБРАЗОВАНИЕ
Эл № ФС 77 - 30569. Государственная регистрация №0420900025. ISSN 1994-0408

Повышение информативности сайта с помощью ориентированных графов

 # 11, ноябрь 2010

авторы: Сорокин А. В., Белим С. В.

УДК 681.326(075)

e-mail: only4andrey@mail.ru, sbelim@mail.ru

Омский государственный университет им. Ф. М. Достоевского,

пр. Мира, 55-а, г.Омск, 644077 (Россия)

Введение

Скорость доступа к информации определяется не только пропускной способностью каналов связи, но и структурой связей. Другими словами способ хранения данных существенно сказывается на скорости предоставления информации пользователю. Количество и архитектура связей начинают играть определяющую роль при низкой пропускной способности линий связи.

Рассмотрим информационную структуру, состоящую из набора связанных текстов, которую принято называть web-сайтом. Поставим задачу повышения информативности отдельных страниц с помощью выделения наиболее сильно связанных данных. Информация, отображаемая на страницах web-сайта, может быть разбита на два вида по способу представления. Первая часть (статическая), представляет собой html-страницу с заранее четко определенной форматом текста и содержанием. Вторая часть (динамическая) формируется в процессе функционирования сайта по определенному алгоритму. Связь между страницами осуществляется с помощью гиперссылок. Будем считать, что количество переходов между страницами прямо пропорционально количеству гиперссылок. Это предположение выполняется не всегда строго, так как предполагает, что сайт организован «разумно», то есть присутствие

каждой гиперссылки оправдано необходимостью перехода. На любом сайте можно выделить группы страниц наиболее тесно связанные друг с другом и слабее связанные с остальными страницами. Для повышения информативности сайта и удобства пользования им в ряде случаев имеет смысл объединить информацию, находящуюся на разных страницах. Поискам возможных алгоритмов автоматической оптимизации сайтов и посвящена эта статья.

Статические и динамические части страниц предъявляют разные требования к используемым алгоритмам. Оптимизация статических страниц может быть осуществлена с использованием «медленного» алгоритма полного перебора с получением точного ответа. Динамическая же часть сайта должна оптимизироваться «быстрым» алгоритмом с возможным понижением точности решения. Для решения задачи оптимизации удобно использовать представление сайта в виде ориенитированного графа.

Во многих ориентированных графах 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 [1,2,3,4] 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.

2. Иерархическая кластеризация дает более быстрый алгоритм, однако не всегда правильный ответ [3,4]. Метод состоит в том, что сначала вычисляются 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 [4].

3. Генетические алгоритмы [5, 6] 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.

1. Модель сайта

Для оптимизации сайта рассмотрим задачу автоматической компоновки информации. А именно, пусть информация разделена на *N* атомарных частей. Причем существуют направленные связи между частями, которые характеризуют взаимное влияние их друг на друга, точнее причинно-следственные связи между ними. Поставим задачу объединения частей в группы с целью повышения информативности содержания групп по сравнению с отдельными частями. Одним из типичных прикладных случаев данной задачи и является набор страниц в сети Интернет со связями в виде гиперссылок. В качестве других примеров можно привести журналы аудита компьютеров одной локальной сети, где связи устанавливаются событиями взаимодействия компьютеров, а задача группировки объектов необходима для объединения рабочих станций в подсети. Еще одним примером могут служить связанные таблицы в одной базе данных.

Решение данной задачи достаточно не очевидно. Например, тривиальное решение, сводящееся к объединению всех частей в одну группу, является, очевидно, «плохим», так как приводит к засорению полезной информации большим количеством второстепенных данных. Подобный подход, по сути, представляет собой не оптимизацию представления информации, а стеганографическое сокрытие ее. Если же обратится к прикладным задачам, то подобное тривиальное решение приводит и вовсе к абсурдным результатам. Так например, оптимизация содержания сайта сведется к размещению всей имеющейся информации на одной странице. Учет же связей вне заданного сайта приведет к сведению всей информации, имеющейся в Интернете в одно место. Аналогичная ситуация будет наблюдаться и в сети компьютеров, которая не будет разбиваться на подсети, что технически не всегда возможно. Для баз данных тривиальное решение приводит к сведению любой базы данных к единственной таблице.

Описанные выше примеры показывают, что группировка информации требует введения некой целевой функции *Q*, показывающей «силу связности». Задача группировки при этом сводится к поиску экстремума функции *Q*. Можно выделить следующие требования на целевую функцию *Q*:

1. Функция *Q* зависит от количества (интенсивности) связей между разными группами информации и связей между элементами одной группы.

2. Вклад связей между элементами одной группы в значение Q превышает вклад аналогичной связи между разными группами.

3. Функция *Q* характеризует преобладание связей внутри групп над связями между группами.

Исходя из этих требований, можно подобрать функцию *Q* в виде разности функции от внутренних связей Q_{intro} и функции от внешних связей Q_{inter}:

$$Q = Q_{intro} - Q_{inter}.$$

В этом случае задача оптимизации сводится к поиску максимума *Q*.

Формализуем задачу, проведя моделирование структуры сайта с помощью ориентированного взвешенного графа. Вершины графа будут соответствовать страницам сайта, а дуги связям в виде гиперссылок. Припишем веса как дугам так и вершинам графа. Вес дуги будет определяться количеством гиперссылок от одной страницы к другой, а вес вершины

количеством гиперссылок на саму себя. То есть вес вершины вычисляется как вес петель с концами в этой вершине.

Будем описывать построенные графы с помощью матрицы смешения *E*, предложенной в работе [2]. 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 [2], более удобным является приведенный вид матрицы смешения $e = \frac{E}{m}$, 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 [2] 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 [7] 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.

Как уже было сказано выше, 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 предложена несколько функций меры связности вершин.

1. Метод парных корреляций был предложен в работе [8] 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 [9].

3. Теоретико-информационный подход [10, 11], 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 [2, 3, 4]. В качестве меры связанности используется величина

$$Q(e) = \sum_{i=1}^{N} e_{ii} - \sum a_i b_i \ ,$$

где $a_i = \sum_{j=1}^{N} e_{ij}$, $b_i = \sum_{j=1}^{N} e_{ji}$.

Выбор функции меры связанности графа зависит от постановки задачи. В данной работе в качестве функции *Q(e)* мы будем сначала использовать выражение, предложенное в работах Ньюмана, а затем рассмотрим некоторые его модификации.

2. Задача поиска связанных структур

Определим более строго процедуру образования связанной структуры в графе. Начнем с алгоритма образования стяжек, как преобразования графа *G* в граф G_1. Выделим в графе *G* подграф *G′* и заменим все входящие в него вершины одно вершиной, при этом вершины подграфа *G\G′* остаются неизменными. Образованная вершина связана дугами с теми вершинами графа G_1, 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, то 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 *Q(e)*. 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.

Задачи выявления связанных структур в графах в данной работе решалась 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.

Как легко видеть, 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 из первого пункта, 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 1 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.

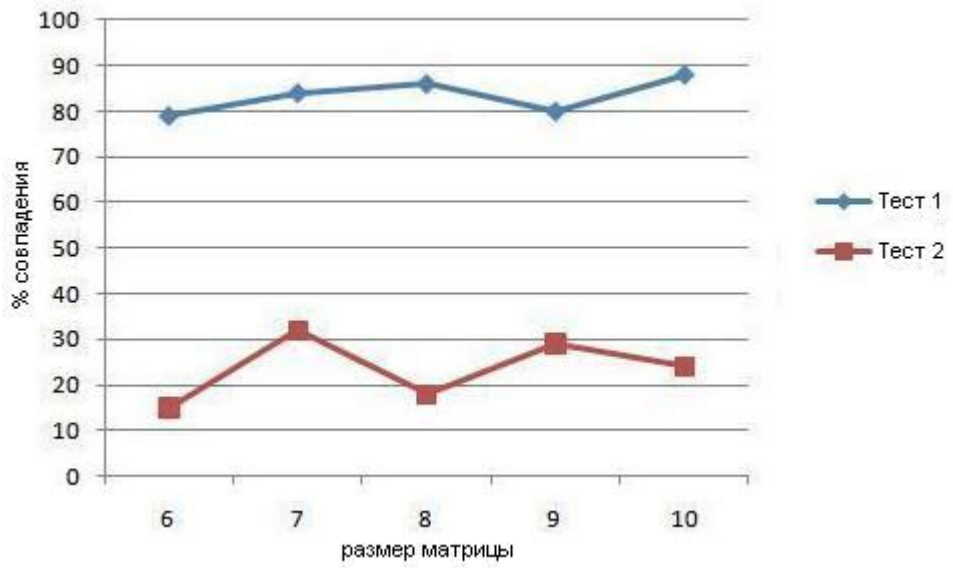

Рис. 1. Результаты компьютерного эксперимента

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 не совпадения решений отклонение меры связанности Q, жадного алгоритма от точного алгоритма составляет не более 20 %, можно считать, что «жадный»

алгоритм дает хороший результат не уступающий в точности другим приближенным методам [11].

Заключение

Программный комплекс, разработанный на основе изложенных выше алгоритмов, позволил выявлять страницы сайта, которые рекомендуется объединять для повышения информативности. Тестирование в реальных условиях показало, что для любого достаточно большого сайта программа выдает рекомендации на не менее чем три объединения. В одном случае рекомендовалось объединить сразу пять страниц.

Данный программный комплекс также был использован для анализа существующих компьютерных сетей пяти средних предприятий. В качестве вершин использовались рабочие станции, а в качестве связей линии коммуникаций. Вес дуг определялся по интенсивности обмена информацией в течение недели, которая определялась на основе служебных log-файлов. Связь определялась на основе IP-адресов.

Список литературы

1. Girvan M., Newman M. E. J. Community structure in social and biological networks // arXiv:cond-mat/0112110v1.(2001)

2. Newman M. E. J., Girvan M. Finding and evaluting community structure in networks // arXiv:cond-mat/0308217v1.(2003)

3. Newman M. E. J. Fast algorithm for detecting community structure in networks // arXiv:cond-mat/0309580v1.(2003)

4. Newman M. E. J. Mixing patterns in networks // Phys. Rev. E. 2003. V.67. P.026126-1 -026126-13.

5. Berryman M. J., Allison A., Abbott D. Optimizing genetic algorithm strategies for evolving networks // arXiv:cs/0404019v1.(2004)

6. Tasgin M., Herdagdelen A., Bingol H. Community detection in complex network using genetic algorithms // arXiv:0711.0491v1.(2007)

7. Leicht E. A., Holme P., Newman M. E. J. Vertex similarity in networks // arXiv:physics/0510143v1.(2005)

8. Meilia M. Comparing clasterings-an information based distance // Journal of Multivariate Analysis. 2007. V.98. P.873-895.

9. Dongen S. V. Performance criteria for graph clastering and Markov cluster experiments. - National Research Institute for Mathematics and Computer Science in the Netherlands, 2000.

10. 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.

11. Meilia M. Comparing clasterings // Technical report, University of Washington, 2002.