УДК 004.021 + 004.42

ИСПОЛЬЗОВАНИЕ МАТРИЦ СМЕЖНОСТИ ДЛЯ ВИЗУАЛИЗАЦИИ БОЛЬШИХ ГРАФОВ

З. В. Апанович[1]

[1]*Институт систем информатики им. А.П. Ершова Сибирского отделения Российской академии наук, пр. Академика Лаврентьева, 6, Новосибирск, Новосибирская обл., 630090*

[1] apanovich@iis.nsk.su

Аннотация

Экспоненциальный рост размеров таких графов, как социальные сети, интернет-графы и др., требует новых подходов к их визуализации. Наряду с представлениями типа «диаграммы связей вершин» все чаще используются визуализации матриц смежностей, а также разнообразные комбинации этих представлений. В данном обзоре рассмотрены новые подходы к визуализации графов большого объема при помощи матриц смежностей и приведены примеры приложений, где эти подходы применяются. Описаны различные типы шаблонов, возникающие при упорядочении матриц смежностей, соответствующих современным сетям, и алгоритмы, позволяющие выделять эти шаблоны. В частности, продемонстрировано, как использование методов упорядочения матриц совместно с алгоритмами поиска таких шаблонов, как звезды, ложные звезды, цепи, почти клики, полные клики, двудольные ядра и почти двудольные ядра, позволяют создавать понятные визуализации графов, имеющих миллионы вершин и ребер. Также приведены примеры гибридных визуализаций, использующих диаграммы связей вершин для представления неплотных частей графа, а матрицы смежностей – для представления плотных частей и их приложений. Гибридные методы используются для визуализации сетей соавторства, глубоких нейронных сетей, сравнения сетей связности человеческого мозга и др.

Ключевые слова*: графы большого объема, визуализация, матрицы смежности, жгуты ребер, гибридная визуализация*

ВВЕДЕНИЕ

В последние годы размер графов, нуждающихся в обработке, возрастает экспоненциально. Например, в англоязычной части Википедии насчитывается более 5,6 миллионов взаимосвязанных статей, Amazon предлагает миллионы продуктов с ребрами, соединяющими каждый элемент с другими похожими продуктами. Граф YahooWeb охватывает более 1,4 миллиарда веб-страниц и 6,6 миллиарда ссылок, а сеть связности типичного человеческого мозга насчитывает 100 миллиардов взаимосвязанных нейронов. Размеры и сложность этих графов являются проблемой для существующих алгоритмов обработки, в частности, для алгоритмов визуализации. В последнее время достигнут большой прогресс алгоритмов визуализации графов, относящихся к классу диаграмм связей вершин (node-link), изображающих графы при помощи глифов, соответствующих вершинам, соединенным прямыми или ломаными линиями, соответствующими ребрам. Эти алгоритмы позволяют визуализировать графы, содержащие несколько миллионов вершин, но они применимы в основном к большим разреженным графам. В связи с этой особенностью в последние годы в очередной раз усилился интерес к визуализации графов при помощи матриц смежностей. Надо сказать, что визуализация и упорядочение матриц, так называемые «визуальные матрицы», используются с конца девятнадцатого века для представления и анализа табличных данных [1]. Первый метод упорядочения матриц был предложен британским египтологом W.V. Flinders Petrie в 1899 году для хронологического упорядочения древних захоронений [2]. Польский антрополог Ян Чекановский [3] использовал упорядочение и визуализацию как метод классификации и кластеризации ископаемых черепов. В тридцатые годы двадцатого века визуализация матриц смежности так называемых социограмм применялась в такой науке, как социометрика [4]. С тех пор эти представления успешно применяются в биологии, неврологии, системах управления поставками и перевозками, системах искусственного интеллекта. В работе [5] показано, что визуализации матриц смежности превосходят диаграммы связей узлов при изображении больших и плотных графов. Также весьма информативные визуализации возникают при сочетании диаграмм связей вершин и матриц смежностей.

В данном обзоре рассмотрены визуальные шаблоны и целевые функции, используемые для упорядочения матриц смежности, методы визуализации очень больших графов на основе упорядочения матриц смежности, а также гибридные методы визуализации графов, использующие различные комбинации изображений типа «диаграммы связей вершин» и матриц смежности.

1. ЗАВИСИМОСТЬ ШАБЛОНОВ, ВСТРЕЧАЮЩИХСЯ В МАТРИЦАХ СМЕЖНОСТИ, ОТ СТРУКТУРЫ ГРАФА И ПРИМЕНЯЕМЫХ АЛГОРИТМОВ УПОРЯДОЧЕНИЯ

Матрица смежности графа G – это квадратная матрица A, где каждый элемент матрицы $a_{ij} \in A$ указывает на наличие или отсутствие ребра между вершинами i и j соответствующего графа G (см. рисунок 1(а)).

(а) (б)

Рис. 1. Диаграмма связей вершин графа и его матрица смежности

Элемент a_{ij} равен единице, если в графе имеется ребро $e = (v_i, v_j)$, и нулю в противном случае. В случае неориентированного графа матрица A является симметричной, а в случае ориентированного графа – асимметричной. В случае взвешенного графа элемент a_{ij} отображает вес ребра. При визуализации матриц смежности обычно используют заполненные и пустые ячейки для представления «нулей» и «единиц», а в случае взвешенного графа применяется цветовое кодирование весов элементов (рис. 1(б)).

Основным шагом при визуализации матриц смежности является упорядочение строк и столбцов, направленное на выявление скрытых структурных шаблонов

в графовых данных. Под *упорядочением* будем понимать функцию $\varphi(v)$, которая присваивает уникальный целочисленный индекс каждой вершине графа *G*.

В настоящее время известно огромное количество алгоритмов упорядочения матриц, которые условно можно разделить на две большие группы: либо оптимизация некоторой целевой функции, либо попытка выделения блочной структуры.

Алгоритмы упорядочения матриц пытаются оптимизировать некоторую целевую функцию, полезную для операций, связанных с сетью. В работе [6] описаны наиболее распространенные целевые функции, такие, как минимальное линейное упорядочение (Minimum Linear Arrangement, MinLA), ширина ленты (BandWidth), ширина разреза, профиль, бисекция ребер, бисекция вершин и др. Одной из наиболее популярных функций, используемых для оценки качества упорядочения матриц, является MinLA, минимизирующая сумму расстояний между концевыми вершинами ребер графа:

$$\mathrm{LA}(\phi, G) = \sum_{(u,v)\in E} |\varphi(u) - \varphi(v)|.$$

Вторая группа алгоритмов ориентирована на выделение блочно-диагональных структур. Они тоже известны достаточно давно [3]. Большинство этих методов, под названием «разбиение матриц и кластеризация блоков», происходит из области биоинформатики.

В настоящее время в литературе по упорядочению матриц не существует консенсуса по поводу целевой функции, которую надо использовать. Поэтому с точки зрения визуализации графов было бы правильнее смотреть на задачу упорядочения как на задачу выявления шаблонов данных, имеющих смысл для пользователя. С этой точки зрения очень важный шаг был сделан в работах [7–9], в которых была предпринята попытка создать каталог визуальных шаблонов, встречающихся в матрицах смежностей и их графовых интерпретаций. Были выделены шаблоны *звезда*, *диагональный блок* и *вне-диагональный блок*, а также *полоса, параллельная диагонали*.

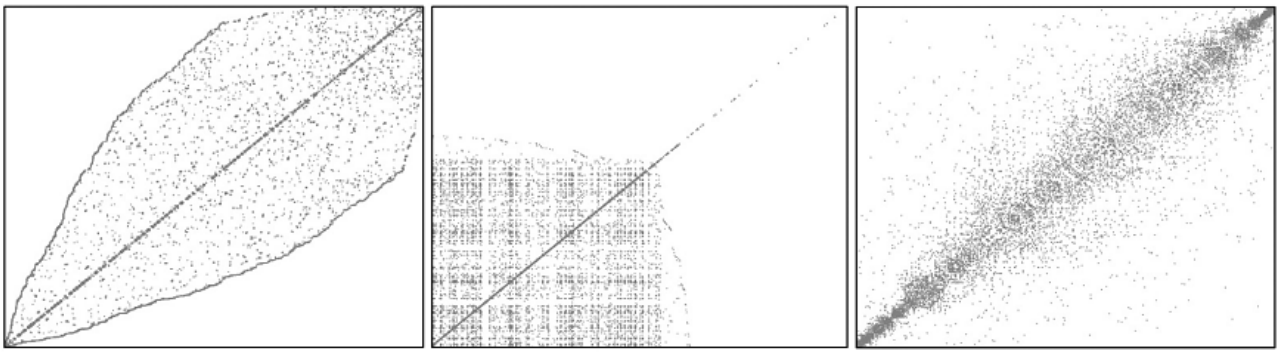
Затем была создана коллекция графов, включавшая как программно сгенерированные графы, так и реальные графы. Сгенерированные графы имели три разных размера (100, 500 и 1000 вершин) и относились к трем разным группам:

случайные графы Erdős-Rényi, графы малого мира (small world graphs) и графы, распределение степеней вершин которых подчиняется закону степенного распределения (power low graphs). Также для экспериментов использовалось несколько реальных графов с размерами от 1770 вершин и 290 тысяч ребер до 42 750 вершин и 3,29 миллиона ребер. Для каждого из графов сначала было создано два начальных упорядочения: исходное и случайное, а затем к каждому начальному упорядочению был применен каждый из ниже перечисленных алгоритмов упорядочения:

— упорядочение методом поиска в ширину;
— упорядочение методом поиска в глубину;
— упорядочение по степеням вершин;
— обратный алгоритм Катхилла–Макки [10];
— алгоритм Кинга [11];
— алгоритм Слоана [12];
— алгоритм на основе дерева разделителей [13];
— алгоритм на основе спектрального упорядочения [14].

Изображения, полученные одним из вышеуказанных алгоритмов, оценивались на предмет *стабильности* и *интерпретируемости*.

Для оценки *стабильности* сравнивалось изображение, полученное одним из перечисленных выше алгоритмов, с исходным и случайным упорядочениями. Изображение считалось стабильным, если в исходных и вновь построенных изображениях обнаруживались сходные структуры из словаря структур. Для оценки *интерпретируемости* изображения анализировалось наличие структур, перечисленных в словаре структур (шаблонов). Эти ранние эксперименты показали, что стабильность зависит и от графа, и от алгоритма, при этом наиболее стабильным оказался алгоритм Слоана, который выдавал похожие результаты независимо от начального упорядочения для всех типов графов. Что касается интерпретируемости, то помимо шаблонов, характеризующих структуру соответствующих графов, были обнаружены специфические шаблоны, характерные для каждого из алгоритмов упорядочения матриц. Эти шаблоны были названы *следами*. Примеры наиболее характерных следов показаны на рисунке 2.



(а) (б) (с)

Рис. 2. Следы, характерные для алгоритмов упорядочения разных типов [9].
(а) след «оболочка», (б) след «горизонт», (с) след «галактики»

На рисунке 2(а) показан след типа «оболочка», который характерен для алгоритмов упорядочения на основе поиска в ширину. След типа «горизонт» показан на рисунке 2(б), он выделяет путь, пройденный алгоритмами, основанными на поиске в глубину. «Горизонт» определяется сплошной диагональной линией, которая следует за главной диагональю и в какой-то момент делает резкий отворот от главной диагонали. Линия может продолжать следовать вдоль основной диагонали, но большинство ребер в графе будет попадать между «горизонтом» и горизонтальной осью. Граф может содержать несколько «горизонтов», расширяющихся от главной диагонали. Это будет происходить, когда в графе есть несколько несвязанных или слабо связанных кластеров. След «галактика» показан на рисунке 2(с), он может быть результатом работы таких методов упорядочения, как спектральные методы или упорядочение на основе степени вершин. В симметричной матрице смежности след «галактика» выглядит как набор точек, которые кажутся притянутыми к центру тяжести на диагонали или самой диагонали. Их внешний вид напоминает эллиптические галактики.

Наиболее распространенные шаблоны, которые присутствуют во многих хорошо упорядоченных матрицах смежности и соответствуют структурным компонентам, часто встречающимся в реальных графах, показаны на рисунке 3:

а) *блочно-диагональный шаблон* состоит из блоков на диагонали матрицы. Когерентные прямоугольные области появляются в упорядоченных матричных

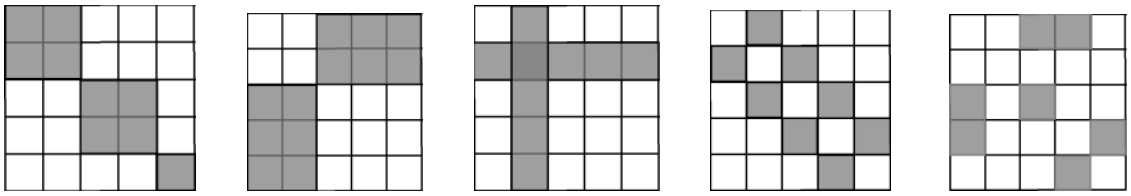
изображениях всякий раз, когда сильно связанные компоненты или клики присутствуют в базовой топологии. Четкие блочные шаблоны позволяют подсчитывать количество кластеров, оценивать наложение кластеров, сравнивать размеры кластеров. Во многих сетях встречаются блочные шаблоны с отсутствующими вершинами или внедиагональные точки, соответствующие связям с другими кластерами;

б) *блочно-внедиагональный шаблон*, соответствует либо двусвязной компоненте, либо подшаблонам блочного шаблона. В случае двусвязной компоненты это может соответствовать, например, отношениям между авторами и их произведениями.

в) *шаблон крест/звезда* возникает в случае, когда вершина связана с большим количеством других вершин (вершина-хаб);

г) *шаблон полоса* выглядит как вне-диагональные непрерывные линии, параллельные диагонали и соответствуют путям или цепям в графе;

д) *анти-шаблон шум* появляется в случаях, когда алгоритм упорядочения не способен выявить имеющуюся структуру графа или никакой структуры нет.



(а) (б) (в) (г) (д)

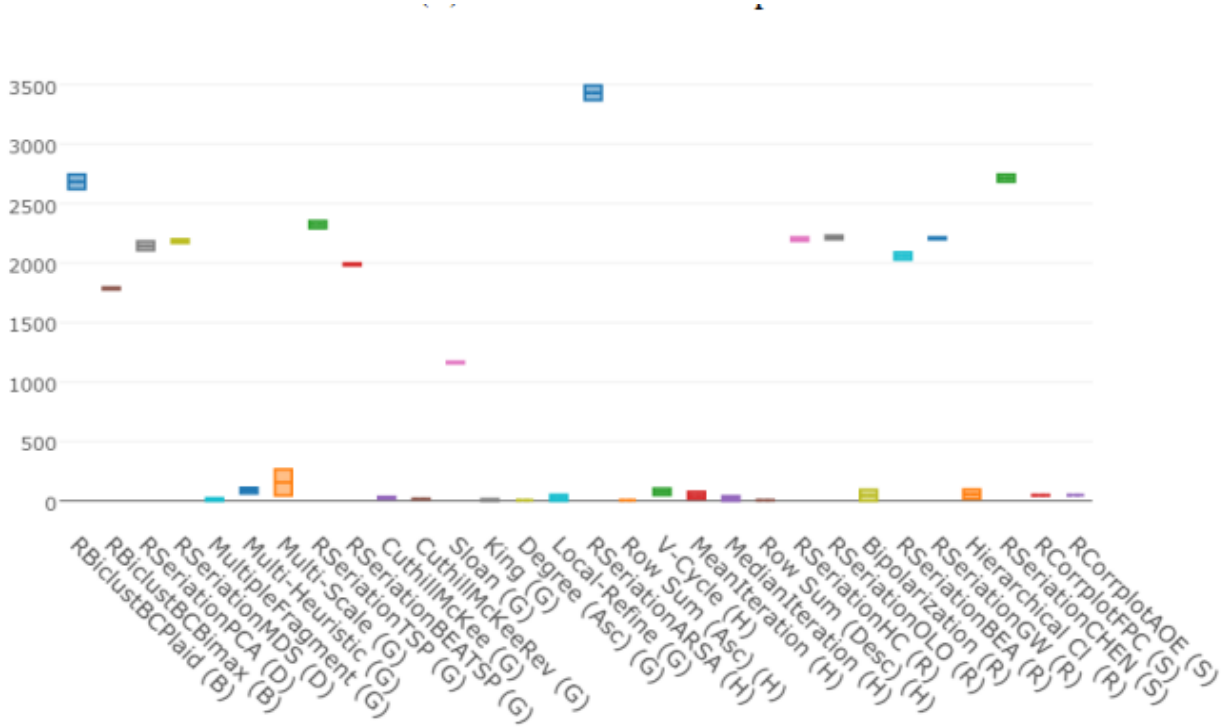
Рис. 3. Примеры шаблонов и анти-шаблонов (а) блочно-диагональный шаблон, (б) блочно-внедиагональный шаблон, (в) шаблон крест/звезда, (г) шаблон полоса, (д) анти-шаблон шум

2. ИЗВЕСТНЫЕ РЕАЛИЗАЦИИ АЛГОРИТМОВ УПОРЯДОЧЕНИЯ МАТРИЦ И ИХ СРАВНЕНИЕ

Большинство алгоритмов переупорядочения доступно в публичных библиотеках, хотя ни одна библиотека пока что не реализует все известные алгоритмы упорядочения. Пакеты R *corrplot* [15], *biclust* [16] и *seriation* [17] обеспечивают большое количество алгоритмов упорядочения для табличных данных. Несколько графовых алгоритмов доступны в C++ библиотеке Boost [18], а библиотека Reorder.js [19] предоставляет множество алгоритмов упорядочения в JavaScript.

Несмотря на большое количество обзоров, проблема оценки качества упорядочения и понимания того, какие шаблоны являются артефактами алгоритмов, а какие шаблоны представляют определенные структуры в данных, имеет решающее значение и по сей день. С этой точки зрения огромный интерес представляет один из последних обзоров [20], использующий следующую методологию.

Для эмпирического сравнения были собраны 35 реализаций разных алгоритмов матричного переупорядочивания, взятых из разных библиотек. Собранные реализации алгоритмов были разбиты на семь основных категорий, таких, как Робинзоновские (R) , Спектральные (S), Уменьшение размерности (D), Эвристические (H), Теоретико-графовые (G), Би-кластеризации (B) и интерактивные (управляемые пользователем.



.

Рис. 4. Время выполнения алгоритмов для больших и плотных графов [20]

Затем были выбраны 150 графов различного происхождения, различающиеся как по размеру от малых (25–100 вершин) до относительно больших (100–1500 вершин), так и по плотности, от разреженных (плотность 0,05–0,28) до плотных (плотность 0,28–0,6). Для всех 150 графов были созданы изображения упорядоченной матрицы смежности, которые использовались в качестве входных данных для

оценки показателей эффективности. Это позволило получить в общей сложности 4348 изображений для 5250 переупорядоченных матриц в качестве основы для сравнения (некоторые упорядочения выдали ошибочные результаты). Все алгоритмы сравнивались по быстродействию, а полученные изображения – по качеству упорядочения (http://matrixreordering.dbvis.de). Сравнение времени выполнения алгоритмов для больших и плотных графов показано на рисунке 4 в виде диаграммы размахов. Время выполнения указано в миллисекундах. Наиболее быстрыми оказались алгоритмы, отнесенные к теоретико-графовой группе, такие, как алгоритм Катхилла–Макки и обратный алгоритм Катхилла–Макки [10], алгоритм Кинга [11], много-шкальные [21] и некоторые робинзоновские алгоритмы [22, 23].

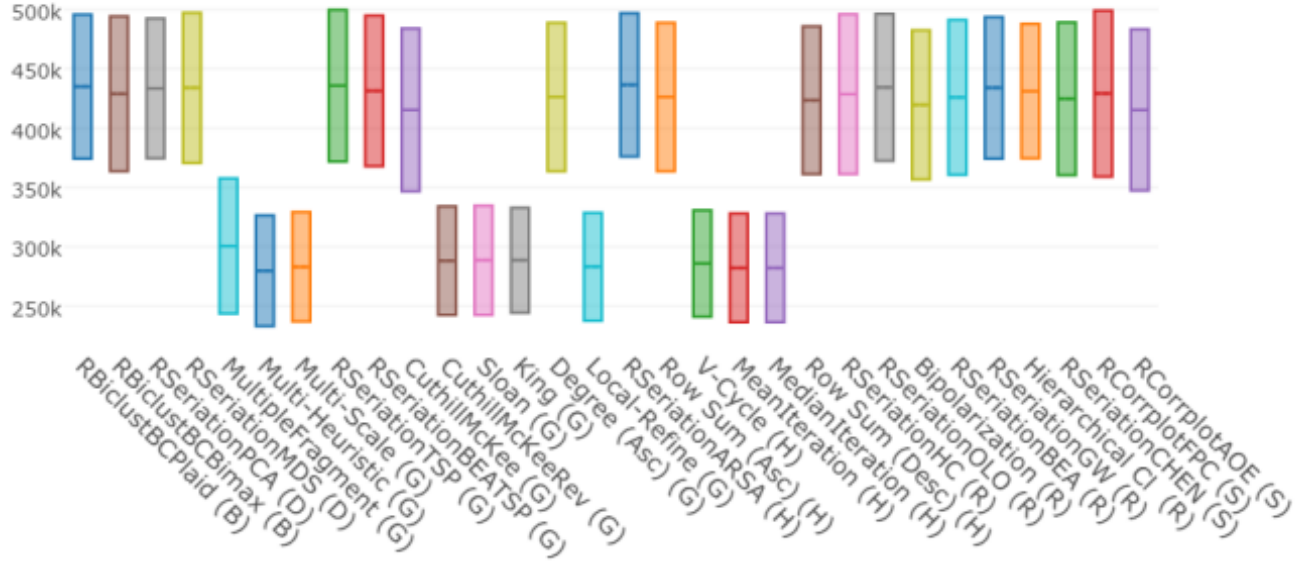


Рис. 5. Значения функции MinLA для больших и плотных графов [20]

Для оценки *качества* упорядочения использовалась функция LA (Linear Arrangement), мера компактности блоков в матрицах. Низкие значения функции LA характерны для упорядочений, которые выявляют когерентные блоки в матрице. В случае упорядочения, выдающего шум, значение функции LA оказывается высоким. Эксперименты показали, что и для этой меры лучше всего себя опять показали теоретико-графовые методы, такие, как [11, 24], превосходя такие алгоритмы, как спектральные методы, робинзоновские и методы би-кластеризации. Диаграмма размахов, отражающая сопоставление значений функции MinLA для больших и

плотных графов, полученных после применения различных алгоритмов, показана на рисунке 5.

Тем не менее, было отмечено, что хотя в идеале хотелось бы иметь конкретное руководство по выбору алгоритма и того, какие параметры следует использовать по отношению к определенным данным и задачам, осталось слишком много открытых исследовательских вопросов, чтобы обеспечить формальные и надежные рекомендации на данном этапе.

Современные исследования ведутся в двух направлениях: с одной стороны продолжается поиск новых алгоритмов упорядочения, направленных на выявление новых шаблонов; с другой стороны, явное выделение топологических структур, таких, как хабы и двудольные подграфы, используется для построения новых упорядочений. Про это более подробно будет сказано в следующих разделах.

3. ИСПОЛЬЗОВАНИЕ МАТРИЦ СМЕЖНОСТИ ДЛЯ ПОИСКА И КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ ГРАФОВ

В последние годы появились работы, позволяющие использовать вышеописанные шаблоны для анализа коллекций матричных визуализаций. Матрицы смежности позволяют конвертировать графовые данные в изображения, а затем применять к ним методы анализа и классификации изображений. Так, например, подход Magnostics [25] оценивает матричные представления в соответствии с наличием специфических визуальных шаблонов, таких, как блоки и линии, указывающих на существование таких структур, как кластеры, двудольные ядра или хабы. Для этого строятся векторные дескрипторы изображений матриц, и сходство между изображениями матриц оценивается как близость между соответствующими векторами. Реализованный в данной системе интерфейс Query-by-Sketch для визуального исследования больших коллекций матриц показан на рисунке 6. Пользователь может изобразить приблизительный эскиз ожидаемого шаблона матрицы (1) и получить от системы ранжированный список результатов (2) в соответствии с выбранным вектором признаков (3). В данном случае ищется блочный шаблон, состоящий из четырех блоков.

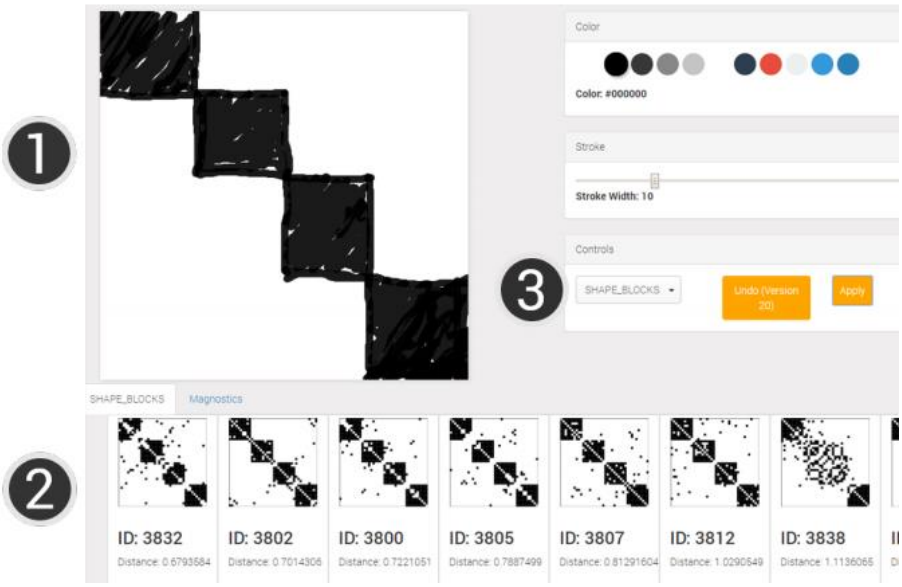


Рис. 6. Интерфейс Query-by-Sketch для визуального исследования больших коллекций матриц [25]

Также эта методика применима для анализа изменения во времени динамических сетей. Следует отметить, что данная методика сравнения изображений матриц смежности сильно зависит от алгоритмов, применяемых для упорядочения матриц. Можно эффективно искать блоки, если эти блоки уже выделены при помощи некоторого алгоритма упорядочения. Поэтому для эффективного сведения задачи поиска и классификации графов к задаче поиска и классификации изображений нужны алгоритмы упорядочения, позволяющие однозначно отобразить структурную информацию в изображение. Так называемый *инвариантный относительно перестановок* метод упорядочения представлен в работах [26, 27]. Этот метод упорядочения использовался для решения следующей задачи.

Дан небольшой подграф большого графа, можно ли по этому подграфу идентифицировать исходный граф?

Для решения этой задачи использовалась следующая схема:

1. Из больших графов, принадлежащих разным типам, извлекались подграфы заданного размера.
2. Для каждого извлеченного подграфа строилась упорядоченная матрица смежности при помощи метода, инвариантного относительно перестановок.
3. Полученные изображения матриц классифицировались при помощи методов глубокого обучения.

Эксперименты по классификации графов на основе изображений матрицы смежности подграфа проводились на общеизвестных тестовых графах, взятых на сайте Стэнфордского проекта по анализу сетей (Stanford Network Analysis Project, http://snap.stanford.edu/data/). Рассматривались такие графы как сеть соавторства DBLP (317 080 вершин, 1 049 866 ребер), сеть совместно покупаемых продуктов Amazon (334 863 вершин, 925 872 ребра), граф гиперссылок между страницами Wikipedia (4 604 вершин, 119 882 ребер), сеть дорог штата Пенсильвания (1 088 092 вершин, 1 541 898 ребер), сеть интернет-страниц с гиперссылками между ними, граф друзей из Facebook (4 039 вершин, 88 234 ребер), граф участников террористической группировки и связей между ними (271 вершина, 756 ребер), социальная сеть Gowalla (196 591вершин, 950 327 ребер), в которой вершинами являются пользователи, а ребрами – общее местоположение, и другие.

Для каждого из вышеперечисленных больших графов выбирались подграфы размера *N* методом случайного блуждания (random walk). Первый узел выбирался случайным образом и добавлялся к текущему набору вершин. На последующих *N*-1 итерациях рассматривались все ребра, имеющие одну вершину в текущем множестве и одну вершину, не принадлежащую текущему множеству вершин. Одно из ребер выбиралось случайным образом, и вершина, инцидентная этому ребру, не принадлежащая текущему множеству вершин, добавлялась к текущему множеству вершин. После того, как все *N* вершин были выбраны, подграф исходного графа, индуцированный этими вершинами, добавлялся к множеству подграфов.

Для каждого из извлеченных подграфов строилось упорядочение, инвариантное относительно перестановок. Алгоритм построения инвариантного относительно перестановок упорядочения работает следующим образом.

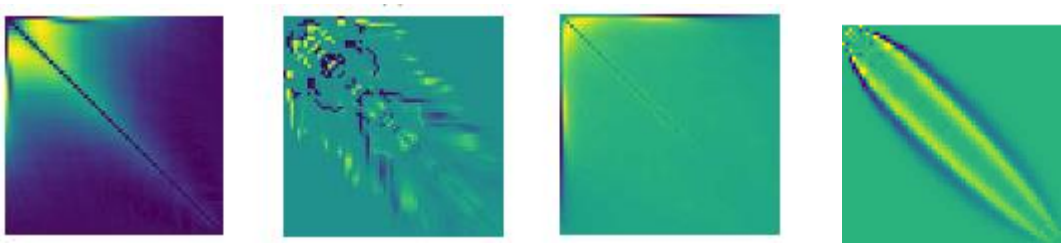
Упорядочение начинается с вершины с наивысшей степенью. Если имеется несколько вершин, имеющих наивысшую степень, конфликт разрешается выбором вершины с наибольшей *k*-окрестностью для *k*=2, затем для *k*=3 и так далее. Если конфликт устранить невозможно, вершина выбирается случайно. Как только первая вершина выбрана, следующей вершиной в упорядочении является вершина с наименьшим кратчайшим путем до первой уже выбранной вершины, в случае конфликта он разрешается по наименьшему кратчайшему пути до второй уже выбранной вершины и т. д. Если двусмысленность все еще существует после сравне-

ния длин кратчайших путей, выбирается вершина с наибольшей степенью. Если конфликт еще не устранен после рассмотрения степени и размера окрестности каждой вершины, выбор осуществляется случайным образом (в симметричных структурах некоторые вершины эквивалентны). Данный способ упорядочения обладает следующими свойствами:

1. Вершины, принадлежащие одному кластеру, будут расположены рядом в матрице смежности.
2. Более важные вершины (с более высокой степенью) предшествуют менее важным вершинам.

Для решения задачи классификации, то есть для выяснения вопроса о том, какой подграф соответствовал какому исходному графу, применялись методы глубокого обучения (предобученные сверточные сети из библиотеки Caffe [28]).

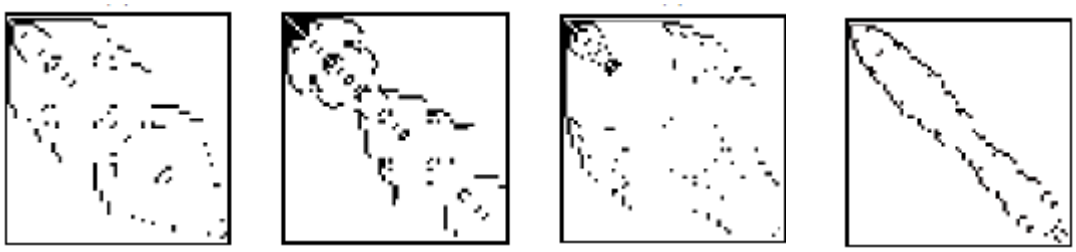
Для обучения классификаторов выбирались случайные подграфы размера 8, 16, 32 и 64 вершины. Для каждого типа графа генерировалось 5 000 выборок (случайных подграфов). Эксперименты показали, что при величине подграфа, равной 64 вершине, удается предсказать граф, из которого извлечен этот подграф размером до миллиона вершин, с точностью, превышающей 90 процентов. Это превосходит результаты классификации графов, получаемые так называемыми ядерными (kernel) методами. На рисунке 7 показаны примеры главных компонент для таких классов графов, как DBLP 7(а), террористическая сеть 7(б), социальная сеть Gowalla 7 (в), граф дорог 7(г).



(а) (б) (в) (г)

Рис. 7. Главные компоненты нескольких больших графов. (а) DBLP, (б) террористическая сеть, (в) социальная сеть Gowalla, (г) граф дорог [27]

На рисунке 8 показаны изображения упорядоченных матриц подграфов, извлеченных из больших графов DBLP 8(а), террористическая сеть 8(б), социальная сеть Gowalla 8(в), граф дорог 8(г).



(а) (б) (в) (г)

Рис. 8. Изображение соответствующих матриц смежности подграфов, извлеченных из больших графов и упорядоченных. (а) DBLP, (б) террористическая сеть, (в) социальная сеть Gowalla, (г) граф дорог [27]

4. ВИЗУАЛИЗАЦИЯ МАТРИЦ СМЕЖНОСТЕЙ БОЛЬШОГО И ОЧЕНЬ БОЛЬШОГО ОБЪЕМА

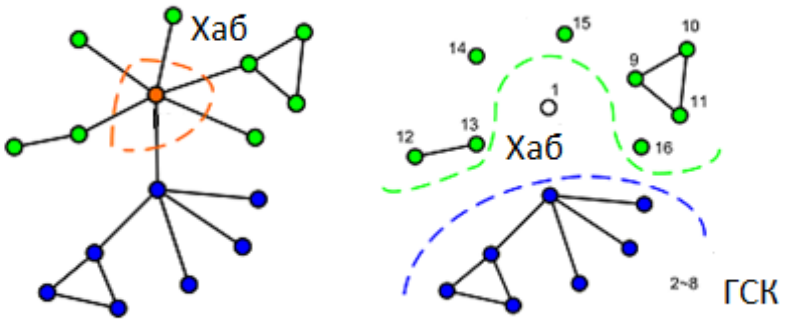
Несмотря на то, что существующие системы визуализации диаграмм связей узлов способны изображать очень большие графы, содержащие миллионы вершин, они плохо справляются с плотными графами, содержащими большое количество ребер. В то же время, благодаря тому, что в матрицах смежности ребро может занимать всего один пиксель на дисплее, они очень популярны при изображении плотных графов. Однако, даже используя один пиксель на каждое ребро, можно визуализировать не более нескольких миллионов ребер. Работы Matrix Zoom [29] и ZAME [30] расширяют подход к визуализации с одним ребром на пиксель путем слияния узлов и ребер в блоки с помощью алгоритмов кластеризации, создавая матрицу смежности, где каждая позиция представляет собой множество ребер в иерархической агрегации.

Однако авторы подхода SlashBurn [31] обратили внимание на то, что большинство графов реального мира изначально не имеют блочной структуры. Они подчиняются степенному закону распределения степеней вершин, то есть в них имеется несколько узлов-«хабов», имеющих очень высокие степени, а большинство узлов имеет низкую степень. Эти хабы хорошо связаны с большинством узлов графа, объединяя все мелкие сообщества в одно огромное сообщество. Поэтому крупные сети легко разрушаются упорядоченным удалением узлов-хабов. После каждого удаления хаба появляется небольшой набор несвязных компонент (спутников), в то время как большинство узлов по-прежнему принадлежит гигантской

связной компоненте. Поэтому упорядочение SlashBurn выполняет итеративно два этапа:

1. узлы с наибольшей степенью удаляются из исходного графа;
2. узлы переупорядочиваются таким образом, что узлы с высокой степенью располагаются в матрице смежности ближе к началу координат, несвязные компоненты смещаются на периферию, а гигантская связная компонента (ГСК) – к середине матрицы.

На следующих итерациях эти же шаги применяются к гигантской связной компоненте. На рисунке 9(а) показана схема выделения хаба и гигантской связной компоненты. На рисунке 9(б) показано присваивание номеров вершин в упорядочении Slashburn. Хаб получает номер один, вершины гигантской связной компоненты (ГСК) – номера со второго по восьмой, а все остальные вершины, попавшие в несвязные компоненты, получают большие номера.



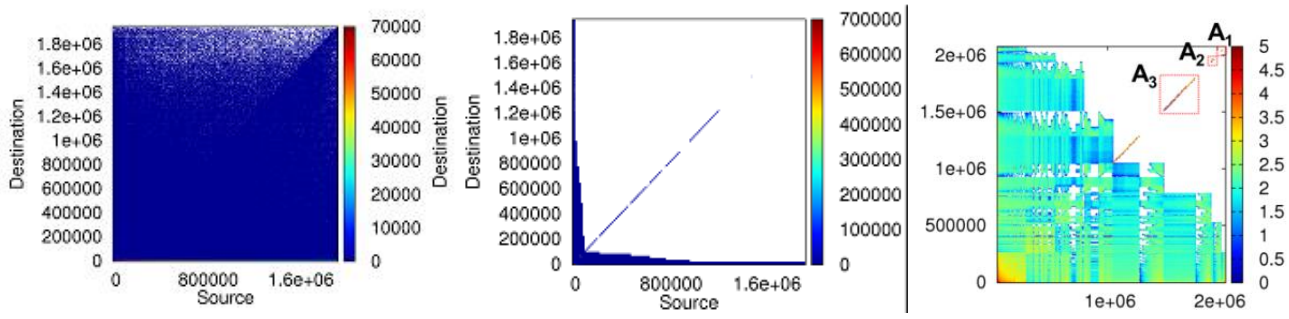
(а) (б)

Рис. 9. (а) граф до применения алгоритма Slashburn (б) граф после применения алгоритма Slashburn [31]

На рисунках 10(а) и 10(б) показаны визуализации графа Weibo-KDD, имеющего 1 944 589 вершин и 50 655 143 ребер, до и после применения алгоритма упорядочения SlashBurn. Можно видеть, что SlashBurn собирает ненулевые элементы матрицы смежностей в левой, нижней и диагональной частях матрицы смежностей, порождая форму, похожую на стрелу.

Упорядочение SlashBurn можно использовать для визуализации очень больших графов, количество вершин в которых превышает миллиард, и, значит, размер матрицы смежности может легко превысить резолюцию типичного экрана. В рабо-

те [32] указанная проблема решается построением проекции исходной матрицы в матрицу меньшего размера, которая может быть показана на типичном экране. Например, матрица размером 1 миллиард на 1 миллиард проецируется в матрицу 1000 на 1000, где значение элемента «уплотненной» устанавливается равным количеству ненулевых элементов в соответствующей подматрице исходной матрицы. Однако эта проекция порождает вторую задачу: «уплотненная» матрица будет почти полной в большинстве случаев, то есть простое линейное масштабирование теряет информацию о тонких различиях малых значений.



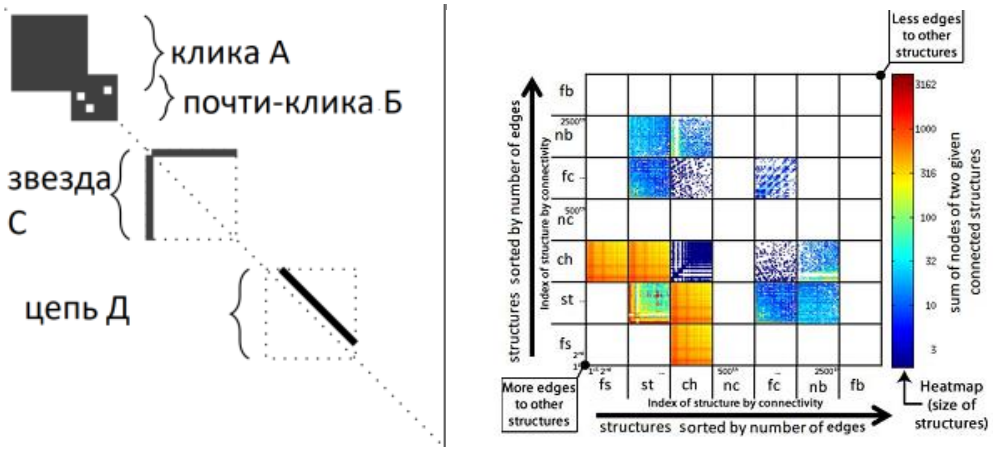
(а) (б) (с)

Рис. 10. (а, б) Матрица смежности графа Weibo-KDD до и после применения алгоритма упорядочения SlashBurn [31]. (с) Матрица смежности графа US Patent упорядоченная при помощи алгоритма Net-Ray [32]

Для решения проблемы полной матрицы перед равномерным сжатием применяется алгоритм SlashBurn, который переупорядочивает узлы и создает огромные пустые области в результирующей уплотненной матрице. В дополнение к переупорядочению узлов авторы решают задачу «полной матрицы» с помощью логарифмического масштабирования числового значения каждого элемента матрицы смежностей (вариант 1-LOG), а также логарифмического масштабирования осей *x* и *y* вместе с числовыми значениями каждого элемента (вариант 3-LOG). В результате удается получить изображение большого графа US patent, имеющего 6 009 555 вершин и 10 565 431 ребер (рис. 10(с)). Такая визуализация позволяет различать плотные и разреженные области графа, а также сообщества, слабо связанные с остальной частью графа (обозначены как A_1, A_2 и A_3).

Как уже говорилось ранее, в реальных графах большого объема имеется значительное количество таких структур, как звезды, двудольные ядра, цепи и т. д. Поэтому естественно возникает идея построения для таких графов матриц

смежности, изображающих связи между разнообразными подструктурами, выделяемыми из исходного графа.



(а) (б)

Рис. 11. (а) схема упорядочения подструктур в матрице смежностей алгоритмом VOG [33], (б) схема упорядочения подструктур в матрице смежностей алгоритмом StructMatrix [34]

В дополнение к общепринятым структурам, таким, как полные клики, полные двусвязные ядра, цепи и звезды, словарь выделяемых структур VOG[33] содержит такие элементы, как *почти клики* и *почти двусвязные ядра*. Почти клика – это структура, содержащая $1-\varepsilon$ ($0< \varepsilon <1$) процентов ребер, которые имела бы полная клика с таким же количеством вершин. Аналогичным образом определяются почти двудольные ядра. Например, если ε=0,2, это означает, что структура считается почти кликой или почти двудольным ядром, если она имеет, по меньшей мере, 80% ребер соответствующей полной структуры.

Основная идея метода VOG состоит в том, чтобы выделить наиболее важные подграфы, которые описывают граф наиболее компактным образом и позволяют пользователю понять основные характеристики графа. Результатом работы алгоритма VOG является упорядоченный список словарных структур, расположенных в матрице смежностей, как это показано на рисунке 11(а). В основу поиска заданных словарных структур положен принцип Минимальной Длины Описания, который требует, чтобы описание графа, использующее выделенные словарные структуры, было самым коротким. Выделенные структуры располагаются на главной диагона-

ли матрицы смежностей. Наиболее важные подграфы располагаются в левом верхнем углу. Заметим, что такой порядок является следствием порядка элементов, заданных алгоритмом кластеризации Slashburn.

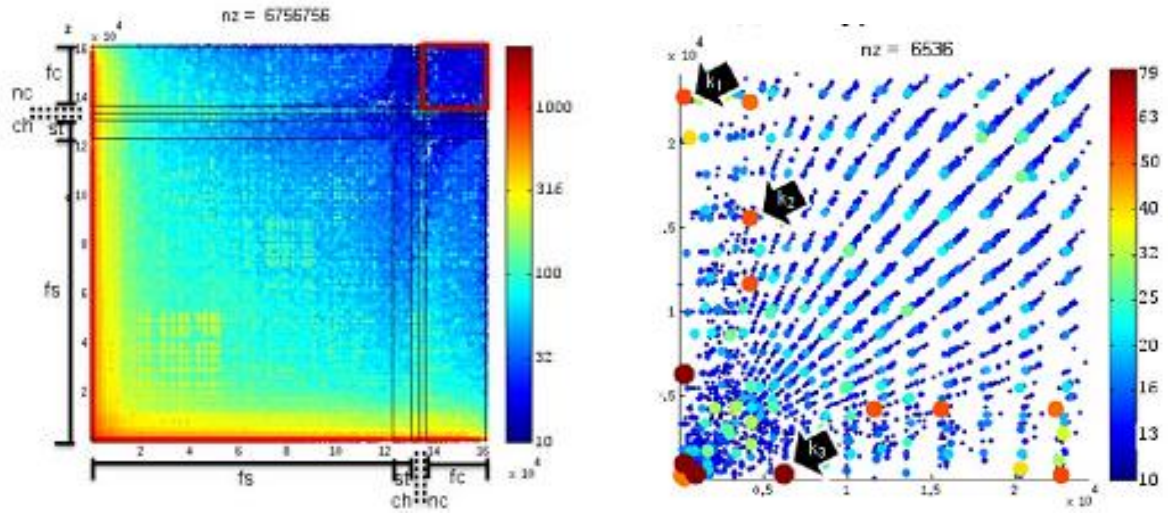
Программа StructMatrix [34] является дальнейшим развитием подхода к упорядочению матриц смежности, предложенного в VOG. Словарь структур программы StructMatrix содержит еще один элемент, называемый «ложная звезда». Ложные звезды – это структуры, подобные звездам (центральная вершина, окруженная спутниками), но спутники центральной вершины имеют ребра, инцидентные другим вершинам, что указывает на то, что эта звезда может быть подструктурой более крупной структуры.

В отличие от VOG, алгоритм выделения подструктур заданного словаря основан не на минимальности описания, а на максимизации выделения заданных словарных структур, которая выполняется сравнением количества вершин и ребер в выделенных подструктурах. Например, подструктура с n вершинами и m ребрами будет классифицирована как полная клика, если $m=n(n\text{-}1)/2$, и будет классифицирована как почти клика, если $m>(1-\varepsilon)(n\text{-}1)n/2$. В случае, если выделенная подструктура оказалась двудольным графом, она будет классифицирована как полное двудольное ядро, если $m=m_1m_2$, почти двудольным ядром, если $m>(1-\varepsilon)m_1m_2$, и звездой, если m_1 или m_2 равно 1.

На рисунке 11(б) показана схема расположения структур, используемая программой StructMatrix [34]. Можно видеть, что StructMatrix располагает подструктуры, имеющие наибольшее количество связей с другими подструктурами в левом нижнем углу, типы подструктур упорядочены слева направо и снизу вверх в таком порядке: ложные звезды, звезды, цепи, почти клики, полные клики, почти двудольные ядра, двудольные ядра. Внутри каждой области, соответствующей типу подструктур, подграфы располагаются по убыванию количества ребер, связывающих их с другими подструктурами.

В дополнение к возможностям масштабирования, связанным с выделением подструктур, алгоритм StructMatrix реализует возможность динамической проекции исходной большой матрицы в матрицу меньшего размера, позволяющей строить изображения графа с различными разрешениями. Поэтому каждый пиксель соответствует нескольким сотням или тысячам ребер. Цвет пикселя соответствует

сумме вершин двух подструктур. Большие количества ребер изображаются в красной цветовой шкале, а небольшие – в синей. На рисунке 12 (а) показано изображение матрицы смежности графа соавторства, сгенерированного авторами подхода на основе набора данных DBLP и насчитывающего 1 366 099 вершин и 5 716654 ребер, построенное программой StructMatrix.



(а) (б)

Рис. 12 (а) матрица смежности графа соавторства DBLP, упорядоченная алгоритмом StructMatrix, (б) фрагмент матрицы смежностей, соответствует связям между полными кликами [34]

На этом изображении можно видеть огромное количество ложных звезд, что соответствует специфике графа соавторства DBLP, в котором многие публикации созданы большим количеством учеников, работающих под чьим-то руководством. Эти ученики, в свою очередь, начинают руководить другими учащимися, появляются новые звезды, и так далее. Меньшинство структур, как видно из матрицы, соответствует авторам, чьи ученики не взаимодействуют с другими учениками, определяя звезды. Наличие полных клик, обозначенных на схеме как *fc*, вполне ожидаемо в таком наборе данных, как граф соавторства DBLP, поскольку каждая статья определяет полную клику среди ее авторов. На рисунке 11(б) показан фрагмент матрицы смежностей с рисунка 11(а), который соответствует связям между полными кликами. Стрелками выделены самые большие клики, клика k_1 имеет 47 авторов, k_2 – 45 авторов, k_3 – 75 авторов.

Эти специфические структуры были замечены из-за их цвета, который указывает на большие размеры. Структуры k_1 и k_3, хотя и большие, в основном изолированы, поскольку они не соединяются с другими структурами, с другой стороны, клика k_2 определяет линии пикселей (вертикальную и горизонтальную) одинаково окрашенных точек, что указывает на то, что она имеет связи с другими кликами.

5. ГИБРИДНЫЕ ИЗОБРАЖЕНИЯ, СОЧЕТАЮЩИЕ МАТРИЦЫ СМЕЖНОСТИ И ДИАГРАММЫ СВЯЗЕЙ УЗЛОВ

Структура социальных сетей может изменяться от очень разреженных (генеалогические деревья) до очень плотных (экспорт и импорт между странами). Известно, что для визуализации разреженных социальных сетей лучше подходят диаграммы связей вершин, а для очень плотных – матрицы смежностей.

К промежуточной категории относятся сети малого мира, которые встречаются очень часто, включая многие сети знакомств, а также глобальный интернет. Для визуализации социальной сети наиболее важным из этих свойств является высокий коэффициент кластеризации, соответствующий наличию многих локально плотных кластеров и небольшого количества узлов-хабов, соединяющих граф, который является глобально разреженным.

В одной из наиболее цитируемых работ по визуализации информации представлена система NodeTrix [35], строящая гибридное представление для визуализации социальных сетей. Авторы попытались преодолеть слабые стороны матриц смежностей, располагая традиционные связи на сторонах матриц смежностей.

В NodeTrix парадигма связи узлов используется для визуализации общей структуры сети, матрицы смежности показывают отдельные сообщества, а межкластерные ребра изображаются в виде кривых, соединяющих границы отдельных матриц. Такое изображение позволяет использовать достоинства обеих парадигм.

Изображение агрегированных вершин при помощи матриц смежностей имеет два преимущества:

Во-первых, у матриц смежностей, все вершины расположены на одной линии, поэтому точки подключения внешних связей хорошо читаемы (также можно использовать изменение порядка вершин для минимизации количества пересечений ребер). Во-вторых, поскольку вершины присутствуют как в столбцах, так и в

строках, на всех четырех сторонах матрицы смежностей, это дает дополнительную свободу при выборе точки подключения внешней связи с тем, чтобы минимизировать количество пересечений ребер.

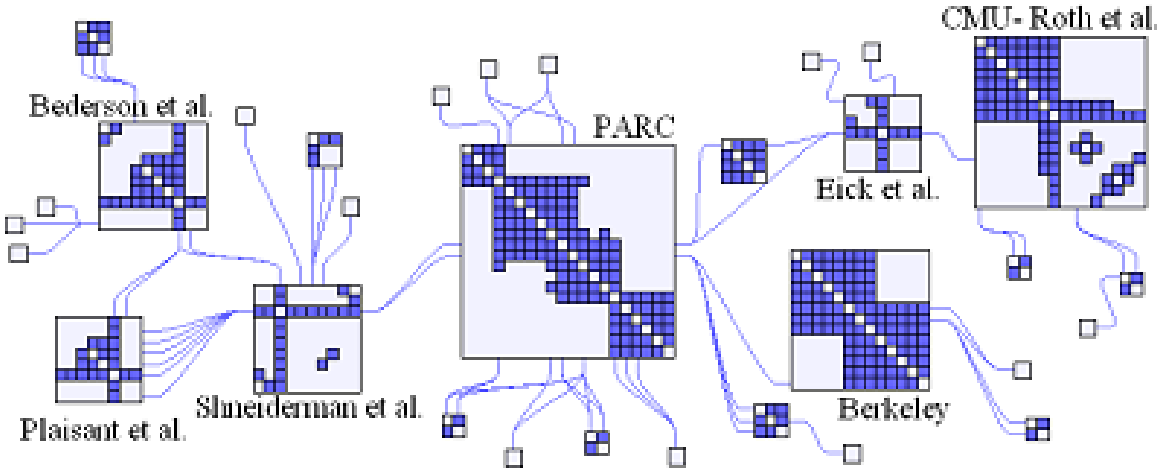


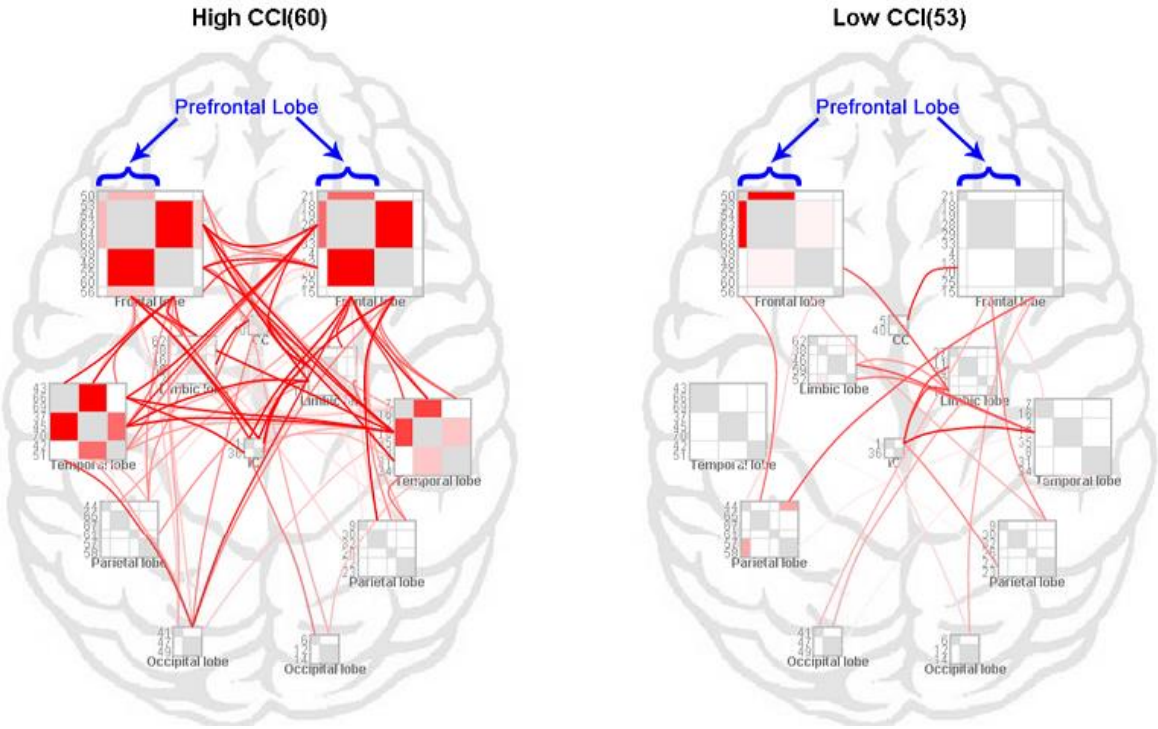
Рис. 13. Фрагмент сети соавторства, изображенной при помощи NodeTrix [35]

Начальное (силовое) размещение вычисляется при помощи алгоритма LinLog, который позволяет быстро идентифицировать кластеры. После этого шага начального размещения пользователю предоставляется возможность менять размещение интерактивно, двигая вершины, группируя множество вершин или удаляя вершины из группы.

На рисунке 13 показан фрагмент сети соавторства, изображенной при помощи NodeTrix. Благодаря гибкости NodeTrix, удается выявить шаблоны сотрудничества в сетях соавторства. Наиболее распространенными шаблонами оказались звезды (кресты в матрице смежности), которые соответствуют, как правило, одному исследователю, который имеет совместные работы со всеми остальными исследователями своей группы, но эти остальные исследователи не сотрудничают друг с другом. Примерами такого шаблона являются матрицы, помеченные именами таких исследователей, как Шнейдерман (Shneiderman), Плэзант (Plaisant), Бедерсон (Bederson) и другие. Блочный шаблон, который можно наблюдать на примере исследователей из Беркли (Berkeley), представляет собой почти клику, что говорит о том, что среди этих исследователей нет главного, и есть несколько групп, внутри которых исследователи сотрудничают друг с другом. Наконец можно

наблюдать промежуточные шаблоны (Roth et al), где есть центральный исследователь, но его коллеги также сотрудничают друг с другом.

Идея NodeTrix оказалось весьма продуктивной, что привело как к значительному количеству приложений, использующих NodeTrix, так и к другим гибридным визуализациям. В частности, на рисунке 14(а) показан пример использования NodeTrix для поблочного сравнения сетей связности головного мозга [36].



(а) (б)

Рис. 14. (а) Визуальное сравнение сетей связности мозга у групп населения с высоким (а) и низким (б) композитным индексом творчества [36]

Для каждой сети связности головного мозга визуальное представление на высоком уровне состоит из нескольких матриц, каждая из которых соответствует функциональному блоку мозга. Каждая строка/столбец в матрице смежности соответствует одной ROI (Region of Interest – высокоспециализированные области мозга), номер которой нарисован как метка на границе матрицы смежности. Поскольку сеть связности между областями ROI может содержать тысячи связей, визуальное сравнение сетей такого объема является трудной задачей. Для упрощения задачи визуального сравнения было осуществлено выявление более крупных блоков

внутри каждой матрицы смежностей, а связи между отдельными блоками собирались в «жгуты» [37–39].

Внутри каждой матрицы элементы, изображенные как закодированные цветом ячейки, указывают на внутри-блочные соединения. Насыщенность красного цвета используется для отображения силы каждого соединения ROI. Для межблочных соединений ROI изогнутые жгуты ребер рисуются между матрицами, где две конечные точки находятся на краю исходного и целевого столбца/строки ROI. Визуализация демонстрирует, что в группе с высоким композитным индексом творчества (CCI) значительно больше связей как внутри матриц смежностей, так и между ними, чем для группы населения с низким композитным индексом творчества.

Понятно, что метод NodeTrix может быть расширен для поддержки визуального сравнения многих других геопространственных сетей (например, динамический трафик и сети миграции), в которых имеются поблочные шаблоны связности, а позиции сетевых узлов фиксируются геопространственно. Например, одно из недавних гибридных приложений MapTrix [40] изображает потоки людей или ресурсов между различными географическими локациями в ситуациях, когда имеется много грузов, а также много пунктов отправления и пунктов назначения. В то время как карты потоков, изображающие пункты отправления и пункты назначения точками на карте, а движение потоков товаров – линиями или стрелками, хорошо справляются с задачей изображения потоков из одного источника, они быстро становятся загроможденными и трудно понимаемыми, когда количество источников товаров увеличивается. Вторым традиционным способом изображения таких многие-ко-многим потоков являются матрицы отправления-назначения (origin-destination matrices, OD-matrices), в которых каждая строка соответствует одному пункту отправления, а каждый столбец – одному пункту назначения. При этом каждый элемент матрицы изображает значение потока от пункта отправления в пункт назначения. Серьезным недостатком такого представления является отсутствие географической привязки пунктов отправления и пунктов назначения. Поэтому появляются гибридные приложения, пытающиеся совместить достоинства обычных географических карт и матриц смежностей. Пример одного из таких гибридных приложений показан на рисунке 15.

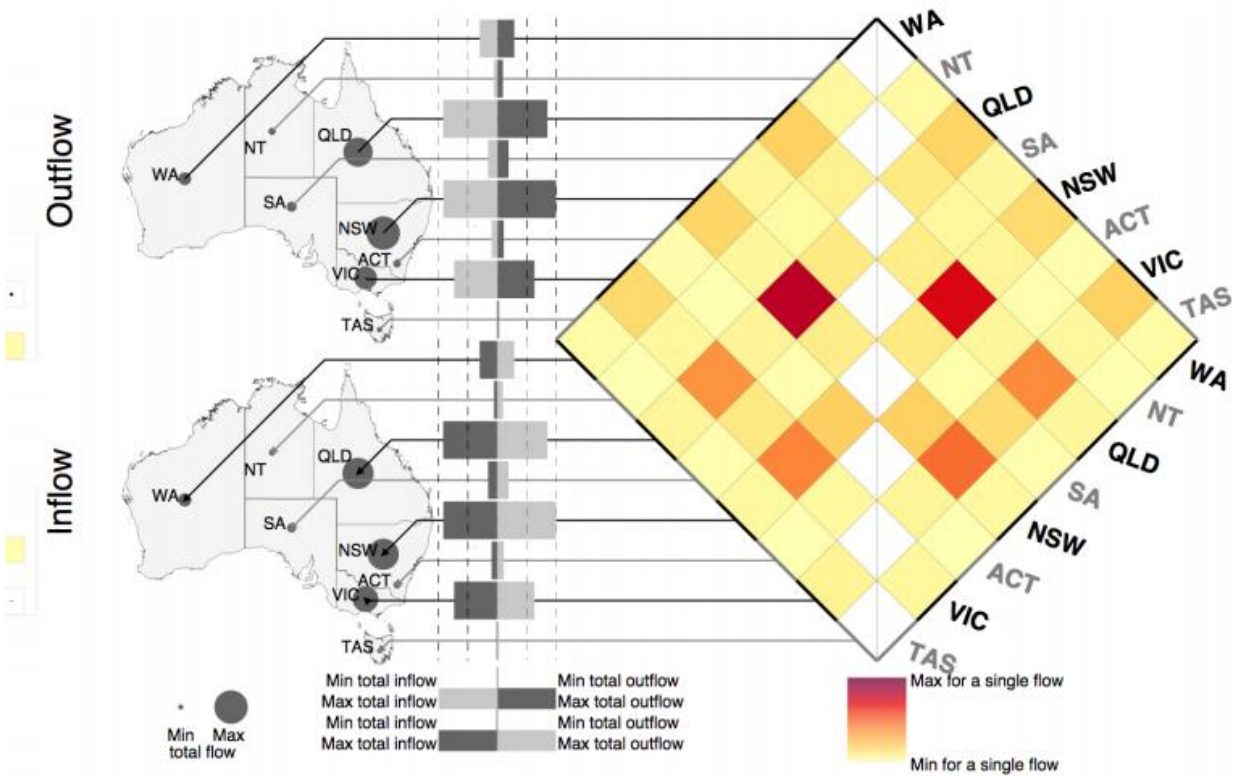


Рис. 15. Гибридная визуализация транспортировки потоков товаров между многими пунктами отправления и многими пунктами назначения [40]

Можно видеть, что для повышения читаемости изображения пункты отправления и пункты назначения изображены на двух идентичных географических картах. На верхней карте размеры окружностей пропорциональны потоку отправляемых грузов, а на нижней – потоку принимаемых грузов. При этом от каждого пункта отправления или пункта назначения имеется линия помощи, соединяющая его со строкой или столбцом матрицы смежностей. С алгоритмической точки зрения, важно так разместить линии помощи, чтобы они не пересекались друг с другом, и при этом порядок строк и столбцов матрицы смежностей был одинаковым. Пользовательские эксперименты показали, что такой способ визуализации позволяет построить понятную визуализацию 51*51 потоков между всеми штатами США, что является невыполнимой задачей для обычной карты потоков.

Еще один тип гибридной визуализации, использующий матрицы смежности в качестве вершин поуровневого изображения графов, а также объединение ребер в жгуты, реализован в инструменте визуальной аналитики CNNVis [41]. CNNVis поз-

воляет анализировать поведение глубоких сверточных нейронных сетей, которые могут содержать десятки или сотни слоев, тысячи нейронов в каждом слое и миллионы связей между нейронами. Исходная сверточная сеть представляет собой ориентированный ациклический граф, в котором каждая вершина соответствует одному нейрону, а ребра – соединениям между отдельными нейронами. Для построения визуализации размеров, приемлемых для пользователя, CNNVis объединяет несколько последовательных слоев в один кластер, и для каждого кластера выбирает один представительный слой, а затем осуществляет кластеризацию нейронов в каждом представительном слое и выбирает несколько представительных нейронов из каждого кластера нейронов. Для найденных кластеров строится два типа изображений. Во-первых, иерархический алгоритм упаковки прямоугольников используется для визуализации фрагментов изображения, распознавать которое обучался данный кластер нейронов. Во-вторых, упорядоченная матрица активации нейронов позволяет выявлять кластерные шаблоны.

С этой целью средние векторы активации нейронов организованы в матрицу активации, где каждая строка является средним вектором активации одного нейрона. Чтобы позволить специалистам исследовать роли разных нейронов по отношению к изображениям разных классов, цвет ячейки в *i*-й строке и *j*-м столбце матрицы активации представляет собой среднюю активацию *i*-го нейрона n_i в классе c_j.

Результирующая гибридная визуализация показана на рисунке 16. Она позволяет анализировать взаимодействия между нейронами и выяснять роль отдельных нейронов по отношению к распознаваемым изображениям, то есть интерактивно можно установить, каким образом активация нейронов (выходные значения нейронов после применения функции активации) влияют на распознавание разных фрагментов изображения. При этом желательно, чтобы визуализация выявляла кластерные шаблоны в активациях нейронного кластера. Чтобы решить эту проблему, был разработан алгоритм переупорядочения матрицы, который может визуально выявлять кластерные шаблоны в пределах данных. Целью упорядочения является максимизация суммарного сходства между отдельными нейронами в матрице активаций.

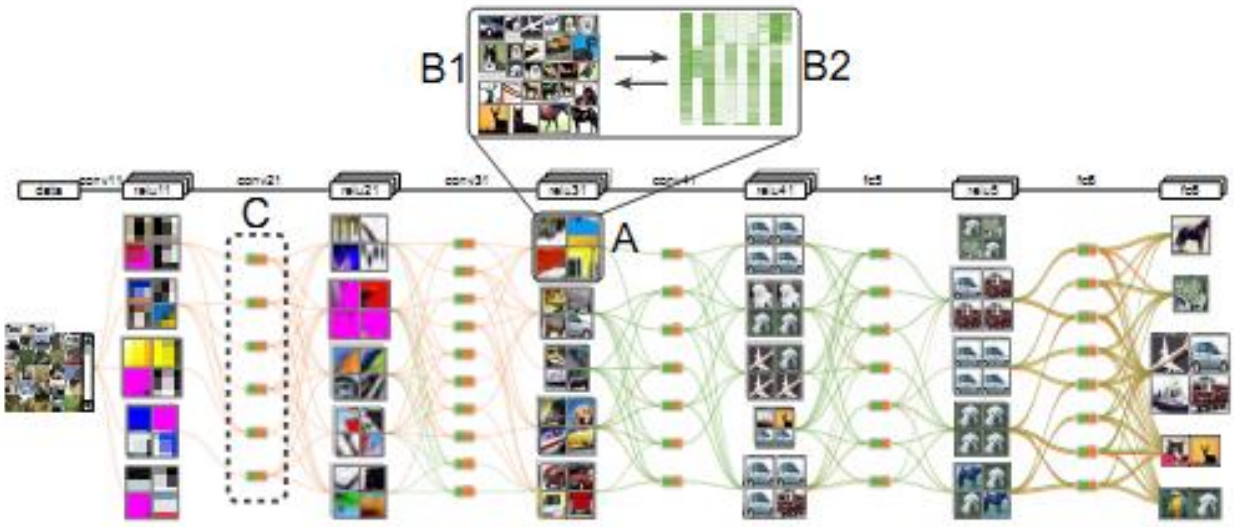


Рис. 16. Гибридное изображение глубоких сверточных нейронных сетей CNNVis [41]

Эксперименты с разработанной программой CNNVis показали, что гибридная визуализация позволяет понимать, диагностировать и улучшать сверточные сети, в частности, она позволяет экспериментально выяснять, каким образом разные архитектуры сетей влияют на производительность, а также выяснять причины, по которым в отдельных случаях процесс обучения не сходится.

ЗАКЛЮЧЕНИЕ

Наряду с изображениями диаграмм связей узлов, визуализация графов при помощи упорядоченных матриц смежностей становится все более популярной, особенно при построении изображений больших и плотных графов. Расширяется множество шаблонов, имеющих смысл для пользователя, возникает необходимость в новых алгоритмах, выявляющих новые шаблоны и их комбинации. Особенно перспективными выглядят гибридные изображения графов, сочетающие диаграммы связей узлов и матрицы смежностей, соединенные жгутами ребер.

СПИСОК ЛИТЕРАТУРЫ

1. *Liiv I.* Seriation and matrix reordering methods: An historical overview// Statistical analysis and data mining. 2010. V. 3, No. 2. P. 70–91.

2. *Petrie F.W.M.* Sequences in prehistoric remains// J. Anthropol. Inst. Great Britain and England. 1899. V. 29, No. 3/4. P. 295–301.

3. *Czekanowski J.* Zur differentialdiagnose der Neandertalgruppe// Korrespondenzblatt Deutsch Ges Anthropol Ethnol Urgesch XL.1909. V. 6, No. 7. S. 44–47.

4. *Forsyth E., Katz L.* A matrix approach to the analysis of sociometric data: preliminary report// Sociometry. 1946. V. 9, No. 4. P. 340–347.

5. *Ghoniem M., Fekete J.D., Castagliola P.* On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis //Information Visualization. 2005. V. 4, No. 2. P. 114–135.

6. *Díaz J., Petit J., Serna M.* A survey of graph layout problems// ACM Comput. Surv. 2002. V. 34, No. 3. P. 313–356.

7. *Mueller C., Martin B., Lumsdaine A.* A comparison of vertex ordering algorithms for large graph visualization// Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on Visualization. 2007. IEEE. P. 141–148.

8. *Mueller C., Martin B., Lumsdaine A.* Interpreting large visual similarity matrices// 2007 6th International Asia-Pacific Symposium on Visualization, 2007. IEEE. P. 149–152.

9. *Mueller C., Martin B., Cottam J., Lumsdaine A.* Matrix representations of graphs. URL: https://www.slideserve.com/amandla/matrix-res-of-graphs.

10. *Cuthill E., MCkee J.* Reducing the bandwidth of sparse symmetric matrices// Proceedings of the 1969 24th National Conference (New York, NY, USA, 1969), ACM '69, ACM. P. 157–172.

11. *King I.P.* An automatic reordering scheme for simultaneous equations derived from network systems// International J. for Numerical Methods in Engineering. 1970. V. 2, No. 4. P. 523–533.

12. *Sloan S.W.* An algorithm for profile and wavefront reduction of sparse matrices// International J. for Numerical Methods in Engineering. 1986. V. 23, No. 2. P. 239–251.

13. *Blandford D., Blelloch G., Kash I.* Compact representations of separable graphs //Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA). 2003. P. 679–688.

14. *West D.B.* Introduction to Graph Theory, Prentice-Hall, Inc., 1996. P. 436–449.

15. *Wei T. Corrplot.* Visualization of a correlation matrix // r package version 0.73. ed., 2013. URL: https://github.com/taiyun/corrplot.

16. *Kaiser S., Leicsh F.* A toolbox for bicluster analysis in r. 2008. URL: https://www.researchgate.net/publication/33029412_A_Toolbox_for_Bicluster_Analysis_in_R.

17. *Hahsler M., Hornik K., Buchta C.* Getting things in order: An introduction to the r package seriation// J. of Statistical Software. 2008. V. 25, No. 3. P. 1–34.

18. *Siek J.G., Lee L.-Q., Lumsdaine A.* The Boost Graph Library: User Guide and Reference Manual// Pearson Education. 2001. P. 352.

19. *Fekete J.-D. Reorder.js:* A JavaScript Library to Reorder Tables and Networks// IEEE VIS 2015, Oct. 2015. Poster. URL: https://hal.inria.fr/hal-01214274. 5

20. *Behrisch M., Bach B., Henry N. Riche, Schreck T., Fekete J.-D.* Matrix Reordering Methods for Table and Network Visualization // EuroVis 2016. 2016. V. 35, No. 3. P. 1–24.

21. *Koren Y., Harel D.* A multi-scale algorithm for the linear arrangement problem// Revised Papers from the 28th International Workshop on Graph-Theoretic Concepts in Computer Science (London, UK, UK, 2002), WG '02, Springer-Verlag. 2002. P. 296–309.

22. *Hubert L.* Some applications of graph theory and related nonmetric techniques to problems of approximate seriation the case of symmetric proximity measures// British J. of Mathematical and Statistical Psychology. 1974. V. 27, No. 2. P. 133–153.

23. *Gruwaeus G., Wainer H.* Two additions to hierarchical cluster analysis//British J. of Mathematical and Statistical Psychology. 1972. V. 25, No. 2. P. 200–206.

24. *George J.A.* Computer implementation of the finite element method// PhD thesis, Stanford University. 1971. P. 1–228.

25. *Behrisch M. et al.* Magnostics: Image-Based Search of Interesting Matrix Views for Guided Network Exploration //IEEE Transactions on Visualization & Computer Graphics. 2017. V. 23, No. 1. P. 31–40.

26. *Ke Wu, Watters P., Magdon-Ismail M.* Network Classification Using Adjacency Matrix Embeddings and Deep Learning//2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). 2016.

27. *Hegde K., Magdon-Ismail M., Ramanathan R., Thapa B.* Network Signatures from Image Representation of Adjacency Matrices: Deep Transfer Learning for Subgraph Classification. *2018.* URL: https://arxiv.org/abs/1804.06275

28. *Krizhevsky A., Sutskever I., Hinton G.E.* Imagenet classification with deep convolutional neural networks// NIPS. 2012. P. 1–9.

29. *Abello J. van Ham F.* Matrix zoom: A visual interface to semi-external graphs// IEEE InfoVis. 2004. P. 183–190.

30. *Kang U., Faloutsos C.* Beyond 'caveman communities': Hubs and spokes for graph compression and mining // ICDM. 2011. P. 300–309. URL: https://arxiv.org/abs/1406.3411

31. *Kang U., Lee J.-Y., Koutra D., Faloutsos C.* Net-ray: Visualizing and mining billion-scale graphs // Adv in Knowledge Discovery and Data Mining. Springer. 2014. P. 348–361.

32. *Koutra D., Kang U., Vreeken J., Faloutsos C.* Vog: Summarizing and understanding large graphs // Proc. SIAM Int Conf on Data Mining (SDM), Philadelphia, PA. 2014. URL: https://arxiv.org/abs/1406.3411.

33. *Gualdron H., Cordeiro R., Rodrigues J.* StructMatrix: Large-scale visualization of graphs by means of structure detection and dense matrices // The Fifth IEEE ICDM Workshop on Data Mining in Networks. 2015. P. 1–8.

34. *Henry N., Fekete J.-D., McGun M. J.* Nodetrix: a hybrid visualization of social networks// IEEE Transactions on Visualization and Computer Graphics, 2007. URL: https://arxiv.org/abs/1406.3411. V. 13. P. 1302–1309.

35. *Yang X., Shi L., Daianu M., Tong H., Liu Q., Thompson P.* Blockwise human brain network visual comparison using NodeTrix representation// IEEE Trans Vis ComputGraph. 2017. V. 23, No. 1. P. 181–190. doi: 10.1109/tvcg.2016.2598472

36. *Holten D.* Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data// IEEE Transactions on Visualization and Computer Graphics. 2006. V. 12, No. 5. P. 741–748.

37. *Апанович З.В.* Методы построения жгутов ребер для улучшения понимаемости информации// Проблемы управления и моделирования в сложных системах труды XV Международной конференции. 2013. С. 439–445.

38. *Апанович З.В., Винокуров П.С., Кислицина Т.А.* Методы и средства визуализации больших научных порталов//Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2011. Т. 9. № 3. С. 5–14.

39. *Yang Y., Dwyer T., Goodwin S., Marriott K.* Many-to-Many Geographically-Embedded Flow Visualisation: An Evaluation// IEEE Transactions on Visualization & Computer Graphics. 2017. V. 23, No. 1. P. 411–420.

40. *Liu M., Shi J., Li Z., Li C., Zhu J., Liu S.* Towards Better Analysis of Deep Convolutional Neural Networks// IEEE Transactions on Visualization & Computer Graphics. 2017. V. 23, No. 1. P. 91–100. doi:10.1109/TVCG.2016.2598831

USING ADJACENCY MATRICES FOR VISUALIZATION OF LARGE GRAPHS

Z. V. Apanovich

A.P. Ershov Institute of Informatics Systems, Siberian Branch of the Russian Academy of Sciences, Novosibirsk State University, Novosibirsk

apanovich@iis.nsk.su

Abstract

Exponential size growth of such graphs as social networks, Internet graphs, etc. requires new approaches to their visualization. Along with node-link diagram representations, adjacency matrices and various hybrid representations are increasingly used for large graphs visualizations. This survey discusses new approaches to the visualization of large graphs using adjacency matrices and gives examples of applications where these approaches are used. We describe various types of patterns arising when adjacency matrices corresponding to modern networks are ordered, and algo-

rithms making it possible to reveal these patterns. In particular, the use of matrix ordering methods in conjunction with algorithms looking for such graph patterns as stars, false stars, chains, near-cliques, full cliques, bipartite cores and near-bipartite cores enable users to create understandable visualizations of graphs with millions of vertices and edges. Examples of hybrid visualizations using node-link diagrams for representing sparse parts of a graph and adjacency matrices for representing dense parts are also given. The hybrid methods are used to visualize co-authorship networks, deep neural networks, to compare networks of the human brain connectivity, etc.

***Keywords*:** large graphs, visualization, adjacency matrices, edge bundles, hybrid visualization.

REFERENCES

1. *Liiv I.* Seriation and matrix reordering methods: An historical overview// Statistical analysis and data mining. 2010. V. 3, No. 2. P. 70–91.

2. *Petrie F.W.M.* Sequences in prehistoric remains// J. Anthropol. Inst. Great Britain and England. 1899. V. 29, No. 3/4. P. 295–301.

3. *Czekanowski J.* Zur differentialdiagnose der Neandertalgruppe// Korrespondenzblatt Deutsch Ges Anthropol Ethnol Urgesch XL.1909. V. 6, No. 7. S. 44–47.

4. *Forsyth E., Katz L.* A matrix approach to the analysis of sociometric data: preliminary report// Sociometry. 1946. V. 9, No. 4. P. 340–347.

5. *Ghoniem M., Fekete J.D., Castagliola P.* On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis //Information Visualization. 2005. V. 4, No. 2. P. 114–135.

6. *Díaz J., Petit J., Serna M.* A survey of graph layout problems// ACM Comput. Surv. 2002. V. 34, No. 3. P. 313–356.

7. *Mueller C., Martin B., Lumsdaine A.* A comparison of vertex ordering algorithms for large graph visualization// Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on Visualization. 2007. IEEE. P. 141–148.

8. *Mueller C., Martin B., Lumsdaine A.* Interpreting large visual similarity matrices// 2007 6th International Asia-Pacific Symposium on Visualization, 2007. IEEE. P. 149–152.

9. *Mueller C., Martin B., Cottam J., Lumsdaine A.* Matrix representations of graphs. URL: https://www.slideserve.com/amandla/matrix-res-of-graphs.

10. *Cuthill E., MCkee J.* Reducing the bandwidth of sparse symmetric matrices// Proceedings of the 1969 24th National Conference (New York, NY, USA, 1969), ACM '69, ACM. P. 157–172.

11. *King I.P.* An automatic reordering scheme for simultaneous equations derived from network systems// International J. for Numerical Methods in Engineering. 1970. V. 2, No. 4. P. 523–533.

12. *Sloan S.W.* An algorithm for profile and wavefront reduction of sparse matrices// International J. for Numerical Methods in Engineering. 1986. V. 23, No. 2. P. 239–251.

13. *Blandford D., Blelloch G., Kash I.* Compact representations of separable graphs //Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA). 2003. P. 679–688.

14. *West D.B.* Introduction to Graph Theory, Prentice-Hall, Inc., 1996. P. 436–449.

15. *Wei T. Corrplot.* Visualization of a correlation matrix // r package version 0.73. ed.,. 2013. URL: https://github.com/taiyun/corrplot.

16. *Kaiser S., Leicsh F.* A toolbox for bicluster analysis in r. 2008. URL: https://www.researchgate.net/publication/33029412_A_Toolbox_for_Bicluster_Analysis_in_R.

17. *Hahsler M., Hornik K., Buchta C.* Getting things in order: An introduction to the r package seriation// J. of Statistical Software. 2008. V. 25, No. 3. P. 1–34.

18. *Siek J.G., Lee L.-Q., Lumsdaine A.* The Boost Graph Library: User Guide and Reference Manual// Pearson Education. 2001. P. 352.

19. *Fekete J.-D. Reorder.js:* A JavaScript Library to Reorder Tables and Networks// IEEE VIS 2015, Oct. 2015. Poster. URL: https://hal.inria.fr/hal-01214274. 5

20. *Behrisch M., Bach B., Henry N. Riche, Schreck T., Fekete J.-D.* Matrix Reordering Methods for Table and Network Visualization // EuroVis 2016. 2016. V. 35, No. 3. P. 1–24.

21. *Koren Y., Harel D.* A multi-scale algorithm for the linear arrangement problem// Revised Papers from the 28th International Workshop on Graph-Theoretic

Concepts in Computer Science (London, UK, UK, 2002), WG '02, Springer-Verlag. 2002. P. 296–309.

22. *Hubert L.* Some applications of graph theory and related nonmetric techniques to problems of approximate seriation the case of symmetric proximity measures// British J. of Mathematical and Statistical Psychology. 1974. V. 27, No. 2. P. 133–153.

23. *Gruwaeus G., Wainer H.* Two additions to hierarchical cluster analysis//British J. of Mathematical and Statistical Psychology. 1972. V. 25, No. 2. P. 200–206.

24. *George J.A.* Computer implementation of the finite element method// PhD thesis, Stanford University. 1971. P. 1–228.

25. *Behrisch M. et al.* Magnostics: Image-Based Search of Interesting Matrix Views for Guided Network Exploration //IEEE Transactions on Visualization & Computer Graphics. 2017. V. 23, No. 1. P. 31–40.

26. *Ke Wu, Watters P., Magdon-Ismail M.* Network Classification Using Adjacency Matrix Embeddings and Deep Learning//2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). 2016.

27. *Hegde K., Magdon-Ismail M., Ramanathan R., Thapa B.* Network Signatures from Image Representation of Adjacency Matrices: Deep Transfer Learning for Subgraph Classification. *2018.* URL: https://arxiv.org/abs/1804.06275

28. *Krizhevsky A., Sutskever I., Hinton G.E.* Imagenet classification with deep convolutional neural networks// NIPS. 2012. P. 1–9.

29. *Abello J. van Ham F.* Matrix zoom: A visual interface to semi-external graphs// IEEE InfoVis. 2004. P. 183–190.

30. *Kang U., Faloutsos C.* Beyond 'caveman communities': Hubs and spokes for graph compression and mining // ICDM. 2011. P. 300–309. URL: https://arxiv.org/abs/1406.3411

31. *Kang U., Lee J.-Y., Koutra D., Faloutsos C.* Net-ray: Visualizing and mining billion-scale graphs // Adv in Knowledge Discovery and Data Mining. Springer. 2014. P. 348–361.

32. *Koutra D., Kang U., Vreeken J., Faloutsos C.* Vog: Summarizing and understanding large graphs // Proc. SIAM Int Conf on Data Mining (SDM), Philadelphia, PA. 2014. URL: https://arxiv.org/abs/1406.3411.

33. *Gualdron H., Cordeiro R., Rodrigues J.* StructMatrix: Large-scale visualization of graphs by means of structure detection and dense matrices // The Fifth IEEE ICDM Workshop on Data Mining in Networks. 2015. P. 1–8.

34. *Henry N., Fekete J.-D., McGun M. J.* Nodetrix: a hybrid visualization of social networks// IEEE Transactions on Visualization and Computer Graphics, 2007. URL: https://arxiv.org/abs/1406.3411. V. 13. P. 1302–1309.

35. *Yang X., Shi L., Daianu M., Tong H., Liu Q., Thompson P.* Blockwise human brain network visual comparison using NodeTrix representation// IEEE Trans Vis ComputGraph. 2017. V. 23, No. 1. P. 181–190. doi: 10.1109/tvcg.2016.2598472

36. *Holten D.* Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data// IEEE Transactions on Visualization and Computer Graphics. 2006. V. 12, No. 5. P. 741–748.

37. *Apanovich Z.V.* Metody postroeniia zhgutov reber dlia uluchsheniia ponimaemosti informatsii//Problemy upravleniia i modelirovaniia v slozhnykh sistemakh trudy XV Mezhdunarodnoi konferentsii. 2013. S. 439–445.

38. *Apanovich Z.V., Vinokurov P.S., Kislitsina T.A.* Metody i sredstva vizualizatsii bolshikh nauchnykh portalov//Vestnik Novosibirskogo gosudarstvennogo universiteta. Seriia: Informatsionnye tekhnologii. 2011. V. 9. No. 3. S. 5–14.

39. *Yang Y., Dwyer T., Goodwin S., Marriott K.* Many-to-Many Geographically-Embedded Flow Visualisation: An Evaluation// IEEE Transactions on Visualization & Computer Graphics. 2017. V. 23, No. 1. P. 411–420.

40. *Liu M., Shi J., Li Z., Li C., Zhu J., Liu S.* Towards Better Analysis of Deep Convolutional Neural Networks// IEEE Transactions on Visualization & Computer Graphics. 2017. V. 23, No. 1. P. 91–100. doi:10.1109/TVCG.2016.2598831

СВЕДЕНИЯ ОБ АВТОРЕ

АПАНОВИЧ Зинаида Владимировна – старший научный сотрудник Института Систем Информатики СО РАН, доцент Новосибирского государственного университета. Сфера научных интересов – визуализация информации, визуализация графов, Semantic Web.

Zinaida Vladimirovna APANOVICH – senior researcher of the Institute of Informatics Systems of SB RAS, Associate Professor of Novosibirsk State University. Research interests include information visualization, graph visualization, Semantic Web.

email: apanovich@iis.nsk.su

Материал поступил в редакцию 12 декабря 2018 года