Обзор средств визуализации графов.

Выполнила Беленькая София Евгеньевна

Научный руководитель: зав. кафедрой КТ, д. т. н., доцент, зав. лаб. ИАиЭ СО РАН, Зюбин В.Е.

Соруководитель: Розов А. С., старший преподаватель кафедры КТ.

# ОБЗОР

1. GraphViz [5] - программное обеспечение визуализации графов с открытом исходным кодом. Пакет состоит из набора утилит командной строки и программ с графическим интерфейсом, способных обрабатывать файлы на языке DOT, а также виджетов и библиотек, облегчающих создание графов и программ для их построения, включая следующие:

* dot — инструмент создания многоуровневого графа с возможностью вывода изображения результирующего графа во множестве форматов (PNG, PDF, PostScript, SVG и ряда других). Предлагается основной, если ребра имеют направление. Dot располагает ребра в одном направлении (сверху вниз или слева на право), пытается избежать пересечений ребер и уменьшает их длины.
* neato — инструмент создания графа на основе «пружинной» модели («spring model», «energy minimised»). Предлагается основной для небольших графов (около 100 узлов), о структуре которых ничего не известно.
* twopi — инструмент создания графа на основе «радиальной» модели. Расположение узлов на концентрических окружностях зависит от их дистанции от данного корневого узла. Положение корневого узла можно выбрать или предоставить это сделать программе.
* circo — инструмент создания графа на основе «круговой» модели.
* fdp — инструмент создания ненаправленного графа на основе «fdp»-модели.
* dotty — графический интерфейс для создания графов.
* lefty — программируемый графический виджет (на языке EZ[3][4]).
* Sfdp - мультимасштабная версия fdp для верстки больших графов.

GraphViz имеет множество опций для индивидуализации диаграмм. Среди них: выбор цветов, шрифтов, изменение стилей линий и узлов, создание гиперссылок и т.д. Имеет 55 видов выходных форматов файлов. Есть API для взаимодействия со многими языками, такими, как java и python. Форма узлов может быть различной.

1. Gephi [6] - это программный пакет сетевого анализа и визуализации с открытым исходным кодом. В нем есть инструменты раскраски вершин и рёбер по их свойствам, настройка подписей, размеров и прочих параметров. Есть экспорт в основные форматы изображений, включая векторные. Поддерживает импорт из БД (SQLite, MySQL, PostgreSQL и Teradata). Gephi реализует несколько алгоритмов компоновки графа, а также экспорт в svg. Импортирует такие форматы представления графов, как GEXF, GDF, GML, GraphML, Pajek NET, GraphViz DOT, CSV, UCINET DL, Tulip TPL, Netdraw VNA, Spreadsheet. Есть API. Поддерживает только круглые узлы.
2. Igraph - это библиотека с открытым исходным кодом для создания и управления графами и анализа сетей. Она написана на C, а также существует в виде пакетов Python и R. Находится в свободном доступе под лицензией GNU General Public License Version 2. Метки узлов в графе могут быть любыми. Граф может быть представлен в форматах Pajek, edgelist(txt).

Реализуются следующие алгоритмы расположения графов:

* circle - распределяет вершины равномерно по кругу.
* drl - уже упомянутый алгоритм для больших графов.
* fr - силовой алгоритм Fruchterman-Reingold.
* graphort - алгоритм на основе физической модели, разработанный Michael Schmuhl.
* gfr - модификация алгоритма Fruchterman-Reingold для больших графов.
* kk - алгоритм Kamada-Kawai.
* lgl - адгоритм LGL.
* random - равномерное распределение точек.
* rt - алгоритм Reingold-Tilford (хорош для деревьев).
* rt\_circular - модификация предыдущего алгоритма для размещения дерева по кругу.

1. LargeViz
2. Graphistry Реализует только одну укладку. Есть ограничение 800 тысяч на максимальное число вершин или рёбер.
3. OGDF [3] - это автономная библиотека C++ для графических алгоритмов, в частности для автоматического рисования графов. Библиотека доступна под лицензией GNU General Public License.
4. OmniGraffle
5. ZGRViewer
6. VizierFX
7. Yed. Распространяется бесплатно. Поддерживает экспорт в форматы PNG, JPG, SVG, PDF, SWF.
8. NetworkX [7][8] - библиотека для манипулирования графами под python, распространяется свободно под BSD-new license. Поддерживает следующие алгоритмы размещения графов:

* Random
* graphviz
* spectral
* spring
* shell
* circular

Узлы могут быть любыми (например, текстом, изображениями, записями XML). Ребра могут содержать произвольные данные (например, веса, временные ряды). Поддерживаются следующие форматы представления графов: GML, GraphML, edge list text files. Алгоритмы представления графов предоставляются GraphViz.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название | Платно для коммерческого использования | Возможность перетаскивания | Auto layout | Количество алгоритмов укладки графа [2] | Формат представления графов | Количество поддерживаемых форматов | Настройка параметров визуализации | Наличие API | Узлы как на требуемых диаграммах |
| GraphViz | - | - | + | 8 | .dot, .gv? | 55 | + | + | + |
| Gephi | - | - | + | 10+ | GEXF, GDF, GML, GraphML, Pajek NET, GraphViz DOT, CSV, UCINET DL, Tulip TPL, Netdraw VNA, Spreadsheet | 2? | + | + | - |
| igraph | - | - | + | 10+ | Pajek, edge list |  | + | + | + |
| LargeViz |  |  |  |  |  |  |  |  |  |
| Graphistry | - |  | + | 1 |  |  |  |  |  |
| OGDF |  |  |  |  |  |  |  |  |  |
| OmniGraffle |  |  |  |  |  |  |  |  |  |
| ZGRViewer |  |  |  |  |  |  |  |  |  |
| VizierFX |  |  |  |  |  |  |  |  |  |
| Yed [4] | - |  | + |  |  | 5+ |  |  |  |
| NetworkX | - | + | + | 6 | GML, GraphML, edge list | 8 | + | + | + |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

[1] <https://habr.com/ru/company/ods/blog/464715/>

[2] Helen Gibson, Joe Faith and Paul Vickers, “A survey of two-dimensional graph layout techniques for information visualization” <http://leonidzhukov.net/hse/2018/sna/papers/gibson2013>

[3] M. Chimani, C. Gutwenger, M. Jünger, G. W. Klau, K. Klein, P. Mutzel. The Open Graph Drawing Framework (OGDF). Chapter 17 in: R. Tamassia (ed.), Handbook of Graph Drawing and Visualization, CRC Press, 2014.

[4] <https://www.yworks.com/products/yed>

[5] <http://www.graphviz.org>

[6] <https://gephi.org>

[7] <http://blog.esemi.ru/2011/12/networkx-python.html>

[8] <https://networkx.github.io>