COMPACT REPRESENTATION OF SEPARABLE GRAPH

by

Xiang Zhang

Submitted in partial fulfillment of the requirements
for the degree of Master of Applied Computer Science

at

Dalhousie University
Halifax, Nova Scotia
August 2017

Table of Contents

Abstract

This is a test document.

Acknowledgements

Thanks to all the little people who make me look tall.

Chapter 1

Introduction

Nowadays, many applications use graphs to show the relationship and represent connectivity between multiple objects. As the graphs inevitably grow very huge, the space issue becomes ever more important.

In this project, we are interested in improving a compact representation mechanism for separable graphs presented by a previous work [1]. In which work the authors proposed an approach to representing separable graphs compactly. Their representation used $O(n)$ bits, meanwhile using constant time on degree or adjacency query, and neighbour listing for one vertex in constant time per neighbour (They took advantage of $O(\log n)$-bit parallelism computation to access $O(n)$ consecutive bits in one operation). Graphs with good separators got good compression by using their representation, and even graphs that are not strictly separable, their representation still works well because the separable components in those graphs can be compressed.In their paper, they provided detail description for compressing the graph by building two structures: the adjacency table and the root-find structure [1], which used vertex separators to encode the graph into a shadow adjacency table [1], as well as support constant time on query operations. But their experiment implemented the data structure by using edge separator instead of vertex separator, which means the shadow label and root-find structure were not needed. In this project, we implement the data structure by using edge separator, and conduct experiment on it.

We firstly follow their idea to implement the data structure by using edge separator. The data structure is building by recursively partitioning a graph into two subgraphs until only one vertex left. During the partition process, an edge separator tree is built. Then make use of the edge separator tree to renumber the vertexes. We store an adjacency list for each vertex, then concatenate the adjacency lists in the order of renumbered vertex to form an adjacency table. Each pair of vertexes in the adjacency is stored via encoding the difference d between the two vertexes, and all the differences are stored contiguously as a sequence of bits in memory. We show, by using adjacency table, $O(n)$ bits are sufficient to encode the separable graph.

Next, we implement several index structures to support degree and adjacency query in constant time, and neighbour listing in constant time per Neighbour. The index structures contains all the index structures involved in their experiment, as well as two index structures implemented by us via applying succinct data structure on bit vector. In their paper, they encoded the degree of each vertex at the start of each adjacency list. However, we found that, the space of degree can be saved in some cases, which causes a trade-off between the time to support degree queries and the space to encode the graph.

1.1 Related Works

1.2 Preliminaries

Graph Separator. A family of graphs G is define separable if : it is closed under taking subgraphs, and satisfies the $f(.)$-separator theorem [6] if there is a constant α ¡ 1 and β ¿ 0 such that each member graph in G with n vertexes has a separator set S of size $|S| < \beta f(n)$, that partition the graph into two parts A and B, with at most αn vertexes in each '[6]. A graph is separable if it belongs to a separable family of graphs.

In this project we focus on the graphs that satisfy the n^c-separator theorem for some constant $c < 1$. One class of graphs that reach this specifications is the planar graphs, which satisfies the theorem that $c = \frac{1}{2}$. Another example can be well-shaped meshes in $\mathbb{R}^d$, with separators of size $O(n^{1-1/d})$ [3].

Lipton *etal*. [2] prove that all classes of graph which satisfy $n/(\log n)^{1+\epsilon}$-separator theorem have bounded density. The bounded density means that every n-vertex member in a class if graphs has bounded $O(n)$ edges. So we can assert that the separable graphs have bounded density.

By making use of the definitions above, we define a class of graphs G satisfies a $f(n)$-edge separator theorem if there are constant variables α ¡ 1 and β ¿ 0 such that each member graph in G with n vertexes, has an edge separator with at most $\beta f(n)$ edges whose removal partitions the graph into two subgraphs with at most

αn vertexes in each. The edge separator is not as common as the vertex separator, because a graph with an edge separator of size s also has a vertex separator of size s at most, but no similar bounds holds when it is conversed [1].

In this project, we will only consider the connected graph, which means all vertexes in the graph have nonzero degree, and we assume each step of partitioning always returns a edge separator set of size $O(n^c)$.

Queries. Our data structure supports three kinds of queries on separable graphs: adjacency queries, degree queries and neighbour listing. The adjacency query checks whether there is an edge between two vertices, the neighbour listing returns the neighbours of a given vertex, and the degree query reports the number of edges connected to a given vertex.

Bit Vector. The compact representation stores a separable graph as difference-encoded bit sequence in memory. To support the three kinds queries (adjacency queries, degree queries and neighbour listing) in constant time, knowing both the number of encoded adjacency lists in the sequence up to an index position, and the start position of a particular adjacency is necessary. More formally, the compact representation requires data structures that support rank and select operations in constant time: Given a set $B[0...n)$ to represented a subset S of a universe $U = [0...n)$ $= \{0, 1, ..., n-1\}$, where $B[i] = 1\ iff\ i \in S$,

- $rank_q(x) = \{k \in [0...x] : B|k| = q\}$
- $select_q(x) = min\{k \in [0...n) : rank_q(k) = x\}$

In this project, we use two kinds of succinct data structure on bit vector : RRR [5] and SD vector [4], to build the index structure which supports queries on separable graph. Both structures satisfy information-theoretic lower bound, and still supports efficient rank and select query operations.

Chapter 2

Doing It

2.1 Getting Ready

Get all the parts that I need. I can throw in a whole pile of terms like preparation, methodology, forethought, and analysis as examples for me to use in the future.

2.2 Next Step

Do it!

Of course, you have to have pictures to show how you did it to make people understand things better.

Chapter 3

Conclusion

Did it!

Bibliography

[1] Daniel K.Blandfor, Guy E. Blelloch, and Ian A. Ksh. Compact representations of separable graphs. *Proc. SODA, ACM/SIAM*, page 679688, 2003.

[2] R. J. Lipton, D. J. Rose, and R. E. Tarjan. Generalized nested dissection. *SIAM Journal on Numerical Analysis*, (16):346–358, 1979.

[3] G. L. Miller, S.-H. Teng, W. P. Thurston, and S. A.Vavasis. Separators for sphere-packings and nearest neighbor graphs. *Journal of the ACM*, (44):1–29, 1997.

[4] D. Okanohara and K. Sadakane. Practical entropy-compressed rank/select dictionary. *Proceedings of ALENEX*, 2007.

[5] R. Raman, V. Raman, and S. Rao. Succinct indexable dictionaries with applications to representations of k-ary trees and multi-sets. *SODA*, 2002.

[6] R.J.Liton and R.E. Tarjan. A separator theorem for planar graphs. *SIAM J.Applied Mathematics*, (16):177–189, 1979.