On the Effect of Locality in Compressing Social Networks

Panagiotis Liakos[1], Katia Papakonstantinopoulou[1†], and Michael Sioutis[2]

[1] University of Athens, Greece
{p.liakos,katia}@di.uoa.gr
[2] Université Lille-Nord de France, Artois, CRIL/CNRS, Lens, France
sioutis@cril.fr

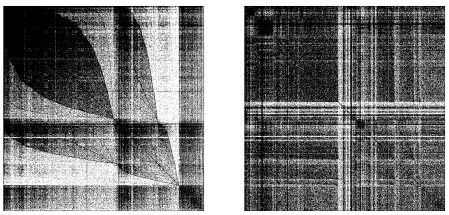
We improve the state-of-the-art method for graph compression by exploiting the locality of reference observed in social network graphs. We take advantage of certain dense parts of those graphs, which enable us to further reduce the overall space requirements. The analysis and experimental evaluation of our method confirms our observations, as our results present improvements over a wide range of social network graphs.

1 Introduction

With the arrival of the Web 2.0 era and the emerging popularity of social network sites, a number of new challenges regarding information retrieval research have been brought to the surface. Users form communities and share mass amounts of high quality information, making the effective and efficient mining of that information an important research direction for modern information retrieval. The structure of the networks studied, i.e., the graphs formed by the relationships among each network's users, is of utmost importance for fields such as user behaviour modelling, sentiment analysis, and social computing. The extraordinary pace at which social network graphs are growing has turned the focus on obtaining space-efficient in-memory representations of them. Information retrieval systems that work directly on those graphs can benefit from such representations.

Graph compression is mostly based on empirical observations of the graph structures. Concerning web graphs, a great number of links is intra-domain, and, thus, their adjacent nodes are close to each other (*locality of reference* or simply *locality*), while nodes close by tend to have similar sets of neighbours (*similarity*). In [6], these two regularities were utilized to decrease space requirements to six bits per edge. Later, Boldi and Vigna proposed a number of techniques that reduced those requirements even further [1]. *Locality* and *similarity* exist naturally in web graphs, as long as their nodes are labelled using a lexicographic order (by URL).

[†]This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) – Research Funding Program: THALES. Investing in knowledge society through the European Social Fund.



(a) (b)

Fig. 1: `youtube-2007` before and after LLP.

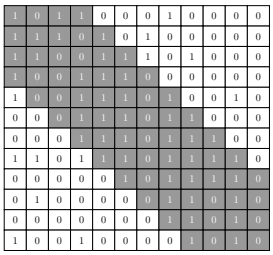


Fig. 2: An adjacency matrix.

Although web graphs exhibit an ordering which exploits additional information – besides the graph itself – that leads to high compression rates, there is no such obvious ordering for social networks. As a consequence, research efforts have focused on testing some well-known permutations of the node labels [3] and discovering heuristics that will obtain effective node orderings [2, 5]. Despite the fact that those attempts proved relatively fruitful, social networks still seem to be harder to compress than web graphs. The increased space requirements of social network graphs can be justified by the existence of a topological difference between the two kinds of graphs [5] that still is to be clarified, as well as by the fact that their compressibility characteristics have not been fully utilized.

In this paper, we concentrate on compressing social network graphs by exploiting the locality property, and build upon the state-of-the-art implementation of Boldi et al., namely, the compression framework of [1] after applying the *Layered Label Propagation* (LLP) algorithm [2] on the input graphs. Our observations not only allow for a greater compression rate – as our experimental evaluation indicates – but leave plenty of room for future exploitation as well.

2 Overview

2.1 Identifying the dense part of the graph

Most compact graph representations are based either on the adjacency matrix representation [4] or on the adjacency lists representation [1] of the graph. For a given graph $G = (V, E)$ the adjacency matrix representation is preferred when G is dense, i.e., when $|E| = \Theta(|V|^2)$, while adjacency lists are preferred when G is sparse, i.e., when $|E| = O(|V|)$.

We combined the two kinds of representations after observing that social network graphs, although rather sparse in general, have a dense part around the main diagonal of the graph's adjacency matrix after the LLP algorithm [2] has been applied on them. This tendency is shown in Figure 1, where the adjacency matrix of a graph from the `youtube` social network is illustrated before (1a) and after (1b) the reordering of its nodes.

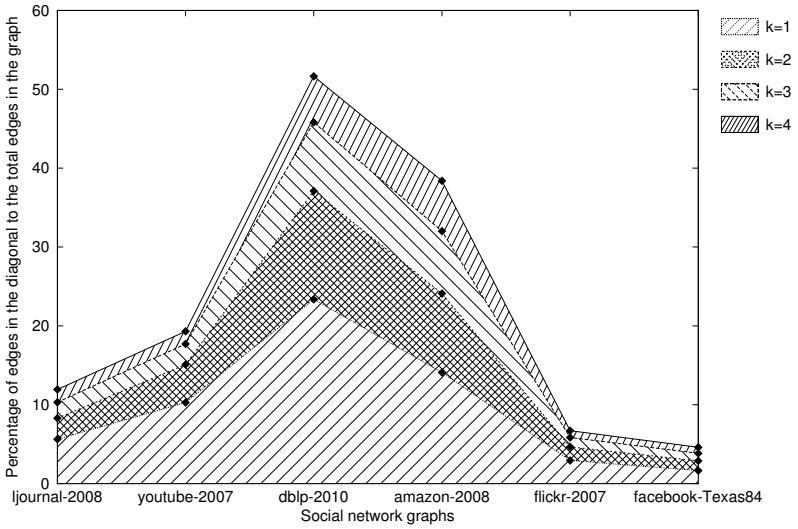


Fig. 3: Percentage of edges contained in the diagonal stripe of various social network graphs for various stripe widths.

More formally, we call this dense area the *diagonal stripe*, and define it as follows: let $k \in \mathbb{Z}_+$, an edge (i, j) is in the k-diagonal stripe, iff $i - k \leq j \leq i + k$. The 3-diagonal stripe of an example adjacency matrix is illustrated in Figure 2.

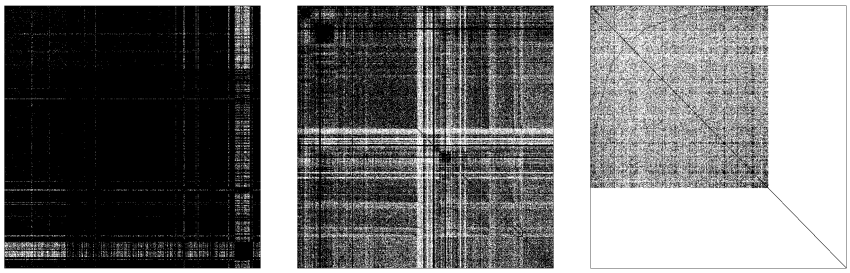
In the graphs we examined experimentally, a large number of edges tends to be in the diagonal stripe, meeting our expectations regarding the locality property. Figure 3 illustrates this trend for $k \in \{1, 2, 3, 4\}$ for the graphs of our dataset, described in detail in Section 3.1.

2.2 Proposing a hybrid method for graph compression

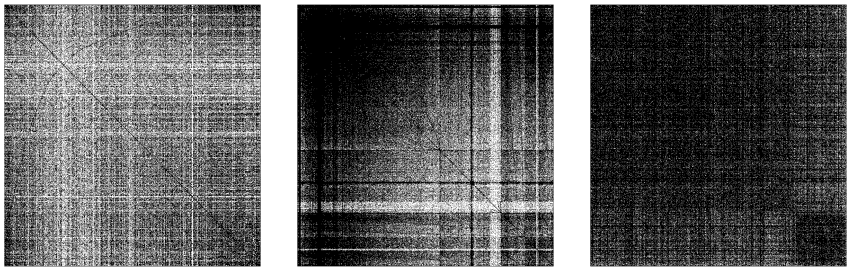
Having identified an opportunity to compress large parts of social network graphs effectively, we propose a hybrid method, which uses a bit vector to represent the diagonal stripe and resorts to the method in [1] to address the issue of compressing the remaining edges. For the rest of this paper we will refer to our method as $\mathsf{BV}_{\mathcal{D}}$ and to the method in [1] as BV.

Every possible pair of nodes (a, b) lying in the diagonal stripe is mapped through a simple function to the bit vector. Thus, the existence of an edge there can be verified in constant time. A big percentage of these pairs represent edges absent from the graph. However, including those pairs in our representation allows us to be aware of the position of every pair and not resort to using an index as in [4], which would not only introduce a similar space overhead, but would dramatically increase the retrieval time as well.

By using BV to compress the rest, sparse part, of the graph, we manage to provide a full graph compression framework and perform comparisons over



(a) `ljournal-2008` (b) `youtube-2007` (c) `dblp-2010`



(d) `amazon-2008` (e) `flickr-2007` (f) `facebook-Texas84`

Fig. 4: Visualizations of the adjacency matrices of some social network graphs.

the whole graph, not only the diagonal stripe. The computational complexity of this approach is approximately equal to the complexity of BV alone, as mapping the diagonal stripe to a bit vector is linear in the number of diagonal edges. Furthermore, this mapping can only decrease the query time on the compressed graph's elements, when compared with the query time of BV alone.

3 Experiments

3.1 Dataset

In order to test our approach we used a dataset of six social network graphs. Figure 4 provides an illustration of their adjacency matrices, where one can clearly see how the diagonal stripe stands out in almost all of the graphs. The origin and characteristics of our graphs are summarized in the following list:

- `ljournal-2008`: *LiveJournal* is a virtual community social website that started in 1999. It comprises $5,363,260$ nodes and $79,023,142$ edges.[3]
- `youtube-2007`: *Youtube* is a video-sharing website that includes a social network. It comprises $1,138,499$ nodes and $5,980,886$ edges.[5]

[3]Collected in [5], retrieved by LAW: `http://law.di.unimi.it/`

graph	# nodes	# edges	% of edges in diagonal	k	compression ratio (bits/edge)	
					BV	$\mathsf{BV}_{\mathcal{D}}$
ljournal-2008	5,363,260	79,023,142	5.62%	1	11.84	11.80
youtube-2007	1,138,499	5,980,886	15.10%	2	14.18	13.79
dblp-2010	326,186	1,615,400	37.12%	2	8.63	7.76
amazon-2008	735,323	5,158,388	43.56%	5	10.77	10.56
flickr-2007	1,715,255	31,110,082	4.66%	2	9.81	9.76
facebook-Texas84	36,371	3,181,310	3.84%	3	8.82	8.80

Table 1: Comparison with BV method.

- dblp-2010: *DBLP* is a bibliography service. Each vertex represents an author, and an edge links two vertices if the corresponding authors have collaborated. It comprises 326,186 nodes and 1,615,400 edges.[4]
- amazon-2008: *Amazon* is a symmetric graph describing similarity among books as reported by the Amazon store, comprising 735,323 nodes and 5,158,388 edges.[4]
- flickr-2007: *Flickr* is a photo-sharing website based on a social network. It comprises 1,715,255 nodes and 31,110,082 edges.[5]
- facebook-Texas84: *Facebook* is the most successful online social networking service. Its Texas84 subnetwork comprises 36,371 nodes and 3,181,310 edges.[6]

The aforementioned graphs vary in size and cover a wide range of social networking services. Thus, they form a thorough evaluation environment for our proposed method.

3.2 Compression ratio comparison

Table 1 shows the number of nodes and edges in each graph, the percentage of edges in the diagonal stripe, the compression ratio achieved by the BV technique [1], and the one achieved by our proposed method ($\mathsf{BV}_{\mathcal{D}}$) for a given k.

As expected, the largest improvement (10%) was achieved for dblp-2010, which has the densest diagonal among all graphs in our dataset. Notable improvements were also observed for graphs youtube-2007 (3%) and amazon-2008 (2%). Surprisingly, for the other three graphs, viz., ljournal-2008, flickr-2007 and facebook-Texas84, $\mathsf{BV}_{\mathcal{D}}$ also managed to surpass the performance of BV, even though the percentage of edges in their diagonal stripes is relatively small.

By outperforming BV for all the graphs in our dataset, we proved that the effect of our observations, even when utilized with a simple approach such as that of $\mathsf{BV}_{\mathcal{D}}$, is very powerful on social network graphs.

[4]Collected by LAW: http://law.di.unimi.it/

[5]Part of the IMC 2007 datasets with LLP [2] applied on it (http://socialnetworks.mpi-sws.org/data-imc2007.html).

[6]The largest of the Facebook100 graphs containing friendships from 100 US universities in 2005 (https://archive.org/details/oxford-2005-facebook-matrix).

3.3 The effect of parameter k

Achieving a good compression ratio with $\mathsf{BV}_{\mathcal{D}}$ depends heavily on choosing an appropriate width for the diagonal stripe of the given graph, defined by k. The optimal values of k for the graphs of our dataset are illustrated in Table 1.

As k increases, more and more edges are included in the diagonal stripe, which, however, becomes progressively sparser. We have found that a good selection of value for this parameter ranges between 1 and 5. The most appropriate value can only be known a posteriori, as it depends on the exact structure of the graph and does not only determine the bits per edge ratio of the diagonal part, but also the compression ratio of the subgraph compressed with BV. However, our results indicate that improvement over BV occurs for most of the values within this range; e.g., for graphs `dblp-2010` and `amazon-2008`, better results were achieved for $k \in [1, 7]$ and $k \in [1, 8]$ respectively.

4 Conclusion and Future Work

In this paper we propose a simple method for exploiting a particular property of social network graphs, namely, locality, in a more effective way than the state-of-the-art method of Boldi et al. [1,2]. Our experiments point out that our method achieves higher compression rates on a broad dataset of social network graphs, while also offering constant retrieval time for the diagonal part of the graph.

We will investigate the issue of optimizing the representation of the diagonal stripe and further decreasing the total compression ratio by using an entropy encoding algorithm, such as Huffman coding, preferably without introducing a significant access time overhead. Moreover, our intuition suggests that a rigorous study of graph reordering methods will lead to the identification of even more attractive labellings for our proposal.

Acknowledgments. We are grateful to Elias Koutsoupias and Alex Delis for discussions on aspects of this work.

References

1. Boldi, P., Vigna, S.: The WebGraph Framework I: Compression Techniques. In: WWW (2004)
2. Boldi, P., Rosa, M., Santini, M., Vigna, S.: Layered Label Propagation: A MultiResolution Coordinate-Free Ordering for Compressing Social Networks. In: WWW (2011)
3. Boldi, P., Santini, M., Vigna, S.: Permuting Web and Social Graphs. Internet Mathematics 6(3), 257–283 (2009)
4. Brisaboa, N.R., Ladra, S., Navarro, G.: k2-Trees for Compact Web Graph Representation. In: SPIRE (2009)
5. Chierichetti, F., Kumar, R., Lattanzi, S., Mitzenmacher, M., Panconesi, A., Raghavan, P.: On compressing social networks. In: KDD (2009)
6. Randall, K.H., Stata, R., Wiener, J.L., Wickremesinghe, R.G.: The Link Database: Fast Access to Graphs of the Web. In: DCC (2002)