On the Effect of Locality in Compressing Social Networks

University of Athens

Panagiotis Liakos[1] - Katia Papakonstantinopoulou[1] - Michael Sioutis[2]

[1] University of Athens, Greece
[2] Université Lille-Nord de France, Lens, CRIL

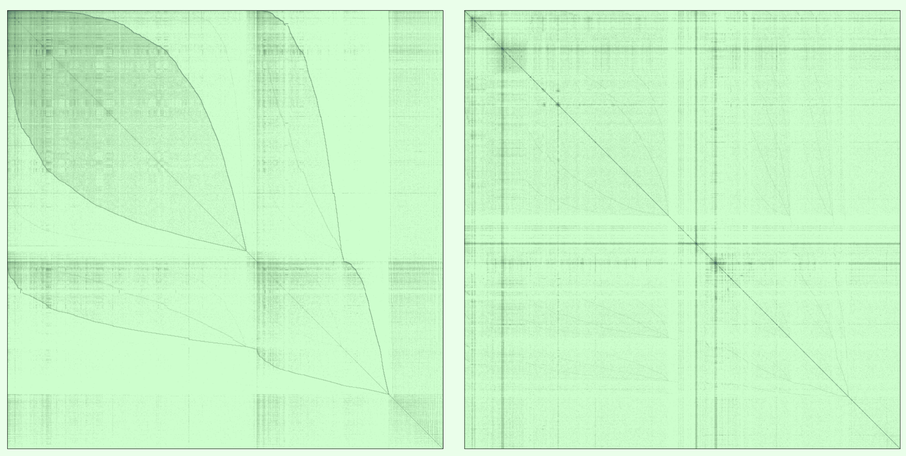
Université Lille-Nord de France

Overview

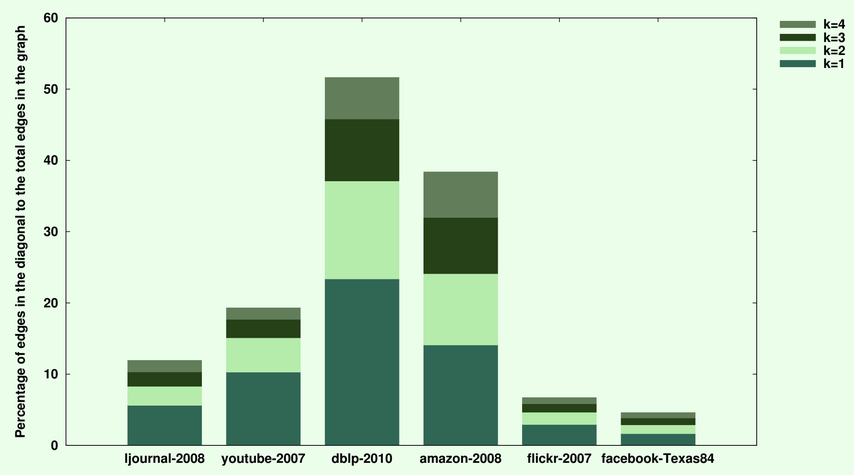
We *compress social network graphs* by exploiting the *locality* property, according to which adjacent nodes have labels that are close to each other, and build upon the state-of-the-art implementation of Boldi et al.

We achieve a *greater compression rate* and show that there is still room for future exploitation. Our approach also *improves overall speed* since it allows accessing a significant amount of edges in constant time.

Our Observations



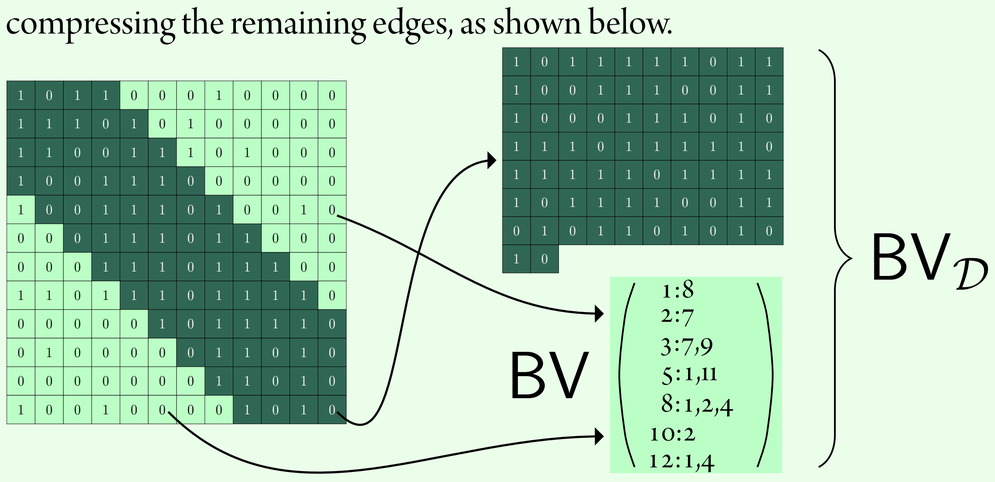
In a social network graph, the *concentration* of edges around the main diagonal of its adjacency matrix can *increase* after applying on it a *reordering algorithm*, e.g., the LLP algorithm of Boldi et al. The adjacency matrix of a graph from the `youtube` social network is illustrated above, before and after reordering its nodes.



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. This trend for $k \in \{1, 2, 3, 4\}$ is illustrated here for a number of well-studied social network graphs.

Our Approach

We call the dense area around the main diagonal of the adjacency matrix of a graph the *diagonal stripe*:
Let $k \in \mathbb{Z}_+$; an edge (i, j) is in the k-diagonal stripe, iff $i - k \leq j \leq i + k$.
An example of a 3-diagonal stripe is illustrated below on the left.
Our contribution: We propose a hybrid method, $\mathsf{BV}_\mathcal{D}$, which uses a bit vector to represent the diagonal stripe and resorts to the method of Boldi et al. (BV) for compressing the remaining edges, as shown below.



Advantages of our approach:

- It results into a more compact representation.
- Our mapping allows the retrieval of the edges of the diagonal stripe in constant time and thus can only decrease the overall query time on the compressed graph's elements, compared with the query time of BV alone.
- The computational complexity is the same as in BV.

Results

We used a dataset of six social network graphs to test our approach.
A comparison of our proposed method $\mathsf{BV}_\mathcal{D}$ for the optimum k with the BV method is outlined in the table below.
Comments:

- Largest improvement (10%) was achieved for `dblp-2010`, which has the densest diagonal stripe in our dataset.
- For the rest of the graphs, $\mathsf{BV}_\mathcal{D}$ managed to surpass BV, even in cases where the percentage of edges in their diagonal stripes is relatively small!

Achieving a good compression ratio with $\mathsf{BV}_\mathcal{D}$ depends heavily on choosing an appropriate k. The most appropriate value can only be known a posteriori.

10% improvement over `dblp-2010`!

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

Future Directions - Contact

We continue with *optimizing the representation* of the diagonal stripe in order to further *decrease the total compression ratio* by using an entropy encoding algorithm, without introducing a significant access time overhead [LPS14, submitted], thus remaining faster than the state-of-the-art method.

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.

Contact info: http://hive.di.uoa.gr/network-analysis