Scalable Large Scale Graph Analytics

Y. Ineichen, C. Bekas, and A. Curioni
IBM Research – Zurich
Computational Sciences
Säumerstrasse 4, 8803 Rüschlikon - Switzerland
{yin,bek,cur}@zurich.ibm.com

In recent years, graph analytics has become one of the most important and ubiquitous tools for a wide variety of research areas and applications. Indeed, modern applications such as ad hoc wireless telecommunication networks, or social networks, have dramatically increased the number of nodes of the involved graphs, which now routinely range in the tens of millions and out-reaching to the billions in notable cases.

We developed novel near linear ($\mathcal{O}(N)$) methods for sparse graphs with N nodes estimating the most important nodes in a graph, the subgraph centralities, and spectrograms, that is the density of eigenvalues of the adjacency matrix of the graph in a certain unit of space.

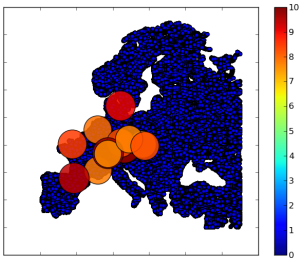

Figure 1: Most central intersections in the European street network.

The method to compute subgraph centralities employs stochastic diagonal estimation [2] and Krylov subspace techniques to drastically reduce the complexity which, using standard methods, is typically $\mathcal{O}(N^3)$. With this technique we can approximate the centralities in a fast, highly scalable and accurate fashion, and thereby open the way for centrality based big data graph analytics that would have been nearly impossible with standard techniques. Subgraph centralities provide a wealth of information in many situations. For example, the subgraph centralities can be used to identify possible bottlenecks in huge networks. Figure 1 visualizes the most central nodes in the European street network[1] (part of the 10th DIMACS challenge [1]) with 51 million nodes. Our efficient parallel implementation only required 800 seconds to compute the centralities on 16 threads.

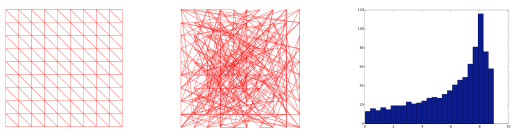

Figure 2: It is almost impossible to determine if the graphs (left and middle column) are the same for a human observer.

In the age of big data it becomes increasingly difficult to compare and visualize graphs. The spectrogram helps to visually interpret and compare data, for example the two leftmost

[1] https://www.cise.ufl.edu/research/sparse/matrices/DIMACS10/europe_osm.html

graphs in Fig. 2. For humans it is impossible to compare the two graphs. In fact, both graphs are exactly the same and one of the graphs can be transformed into the other by a set of simple permutations. The spectrogram shown on the right in Fig. 2 captures the essential characteristic of the graphs immediately and graphically. It transforms complex graphs into a 1-dimensional vector - a simple picture that fosters convenient interpretation.

Spectrograms are powerful in capturing the essential structure of graphs and provide a natural and human readable (low dimensional) representation for comparison. How about comparing graphs that are almost similar (see Figure 3)? Of course, this is a massive dimensionality reduction, however at the same time the shape of the spectrogram yields a tremendous wealth of information.

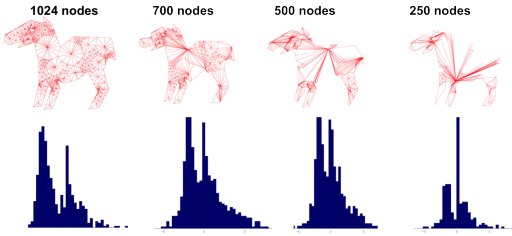

Figure 3: Almost similar graphs and spectrograms

Solving the underlaying eigenvalue problem is getting much harder to master in the era of big data. The cubic complexity of dense methods and the limitation of iterative techniques to look deep into the interior of the spectrum at an acceptable cost, call for a new approach. Our approach starts by estimating $\lambda_{\min}$ and $\lambda_{\max}$ of the adjacency matrix by a few steps of Lanczos, in order to shift and scale the adjacency matrix to have its spectrum in the interval $[-1, 1]$. Next, we divide the range $[-1, 1]$ in the number of requested bins μ, known as inflection points. Subsequently, we estimate the number of eigenvalues below μ, using trace estimation techniques (similar to [3, 2]) of the Fermi-Dirac distribution function, to compute the spectrogram.

In order to tackle arising big data challenges an efficient utilization of available HPC resources is key. Both developed methods exhibit an efficient parallelization on multiple hierarchical levels. For example, computing the spectrogram can be parallelized on three levels: bins and matrix-vector products can be computed independently, and each matrix-vector product can be computed in parallel. The combination of a highly scalable implementation and algorithmic improvements enable us to tackle big data analytics problems that are nearly impossible to solve with standard techniques. A broad spectrum of applications in industrial and societal challenges can profit from fast graph analytics.

Acknowledgments Yves Ineichen and Costas Bekas acknowledge the support from XDATA program of the Defense Advanced Research Projects Agency (DARPA), administered through Air Force Research Laboratory contract FA8750-12-C-0323.

References

[1] D. A. Bader, H. Meyerhenke, P. Sanders, and D. Wagner. *Graph partitioning and graph clustering*, volume 588. American Mathematical Soc., 2013.

[2] C. Bekas, E. Kokiopoulou, and Y. Saad. An estimator for the diagonal of a matrix. *Appl. Numer. Math.*, 57(11-12):1214–1229, Nov. 2007.

[3] M. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.