Compressing Graph Collections Based on Edit Arborescences

Lucas Gnecco[1], Nicolas Boria[1], Florian Yger[1], Sébastien Bougleux[2] and David B. Blumenthal[3]

[1]LAMSADE, Université Paris-Dauphine
[2]GREYC, Université Caen Normandie
[3]Chair of Experimental Bioinformatics, Technical University of Munich

September 2020

Given a collection of attributed graphs $\mathcal{G}$ with N graphs $G_1, ..., G_N$, we propose a unified and compact representation of $\mathcal{G}$ as an *Edit-Arborescence* $\mathcal{A}$. The nodes $u_1, ..., u_N$ of $\mathcal{A}$ correspond to graphs of $\mathcal{G}$ and an arc (u_i, u_j) corresponds to the edit path transforming u_i into u_j using the elementary edit operations. Arcs are penalized by a cost that measures the number of bits needed to save such an edit path. We fix a node u_0 corresponding to the empty graph and, for every graph in the collection, consider an arc (u_0, u_i) that represents the usual encoding of the graph by listing all its nodes, edges and attributes. This model yields a general framework for the reference based compression of graphs, as well as an underlying optimization problem which we denote by Arborescence-Based Compression Problem (ABC).

In this paper we propose a compact format for encoding the edit paths between graphs. The GED and the corresponding edit paths are approximated to weight the edges (u_i, u_j) so that a minimal spanning arborescence problem with root u_0 can be solved to find the optimal *Edit-Arborescence* $\mathcal{A}^*$ for $\mathcal{G}$. The method is then tested by writing $\mathcal{G}$ using the edit paths in $\mathcal{A}^*$ and reconstructing a new collection of graphs $U_1, ..., U_N$ where each decompressed graph U_i is isomorphic to the original G_i of $\mathcal{G}$. The total memory required to store the edit paths and meta data is compared with the theoretical optimal encoding size for attributed graphs and with the traditional file compression formats performance (*.zip*, *.tar.gz*). Results obtained so far show that the ABC method generally outperforms traditional compression methods.