An Algorithm for Determining Isomorphism

Chris Augeri[1*], Barry Mullins[1], Leemon Baird[2], Dursun Bulutoglu[1], and Rusty Baldwin[1]
[1]Air Force Institute of Technology (AFIT), Wright-Patterson Air Force Base, OH
[2]United States Air Force Academy (USAFA), USAFA, CO

We present an algorithm for finding a canonical isomorph of a graph; subsequent comparison of two similarly obtained canonical isomorphs determines isomorphism. Our work is based on the PageRank algorithm, which transforms an adjacency matrix to a positive stochastic matrix, computes the leading eigenvector of the transformed matrix, and sorts vertices (web pages) on this eigenvector. By the Perron-Frobenius theorem, this eigenvector exists and is unique; however, it will not typically find a canonical isomorph.

Our first extension lexicographically sorts on all eigenvectors and applies the induced permutation to the adjacency matrix. We iterate this step by the base-2 logarithm of the number of vertices. The next extension considers other information matrices, such as the inverse, to sort on. We use two isomorphism-preserving transforms to ensure an inverse exists: the first is a seemingly trivial way to improve performance; the second leverages the Gershgorin Circle theorem to yield a diagonally dominant matrix. Finally, we sort lexicographically on the inverse's individual sorted rows and then its native form.

Our numerical implementation uses parallel linear algebra libraries and Cholesky decomposition; using symbolic libraries, we have implemented it with arbitrary precision. This simple, deterministic algorithm executes in polynomial time. We discuss rare cases where it gives incorrect answers, and show that it is correct for all graphs having eight or fewer vertices, and for almost all regular and random graphs that we have tested.

Keywords: Graph Isomorphism, Canonical Isomorph, Inverse, PageRank