Abstract

Problem:
- Want to know how similar two networks are, however, when given two large networks, it is hard to tell vertex correspondence. Two isomorphic networks may look drastically different.
- Enumerating every possible vertex correspondence takes $n!$ combinations, which is too exhaustive as $n$ increases to 1000 or 10000.

Goal:
- Quantify the similarity in structure between two large networks

Solution:
- Compute similarity based on occurrence of trees
- Since directly counting of tree size $k$ takes $O(n^k)$ runtime complexity, we estimate the similarity statistic via color coding algorithm

Applications:
- Online social networks, protein-protein interaction network alignment, and shape matching in computer vision

Methods

Tree Counting:
- Randomly assign color to each node in given graph
- Center graph by assigning weight values to edges
- Partition trees into smaller trees
- Use dynamic programming to count colorful trees (bottom-up counting via tree partitions)

Similarity Score:
- Find all non-isomorphic trees of a given size
- Determine the counts for each of those trees in both graphs
- Take dot product of the vectors containing tree counts and normalize

Sampling:
- Sample a specified number of nodes from each graph, and from that node sampling, sample a certain number of edges
- Calculate similarity scores between pairs of these same-sized samples

Tools:
- C++: Fast Approximate Subgraph Counting and Enumeration (FASCIA) package from Penn State University [1]
- Duke Compute Cluster: OpenMP and job arrays

Results

- Color coding algorithm scales quadratically as network size increases
- Almost complete separation between the 20 correlated and 20 independent Erdos-Renyi graphs over 1000 different random colorings with size of 1000 nodes and edge probability of 0.001 at $K=6$, where $K$ is the number of edges of the tree, which provided the best results to runtime tradeoff

References