Background

**Frequent graph mining**: Discover frequent graph patterns in labeled networks. Application in bio-informatics, chemistry, social network.

**Transactional setting**: A set of many moderate sized networks, can easily be mined in parallel.

**Single graph setting**: Single large sparse graph, more challenging as the input may not fit in memory of a single machine.

Facebook currently has 1.4 billion active monthly users.

Existing approaches [1-3] are sequential/parallel and shared-memory based, cannot mine a very large graph. We are in need of a scalable distributed solution that can process large graph.

Challenges

1. **Large input graph**: The input graph is split into multiple parts with 1 hop overlap. Graph partitioning is hard; existing solutions use heuristics. A balanced partition is desired.

2. **False negatives**: Certain edges involved in the isomorphisms of a pattern can span across partitions, and thus missed.

3. **Local support and pruning**: Need a local support measure that can effectively prune patterns that are globally infrequent.

4. **False positives**: Vertex mappings of a pattern can be large, and are exchanged only when a pattern is estimated to be globally frequent.

Broader Impact

Our distributed mining approach made it possible to discover patterns from massive networks.

The scalability plot shows performance from scaled up Protein interaction graph (www.rcsb.org). 1x graph consists 17.4 million vertices and 68.5 million edges. Therefore, 64x consists more than 1 billion vertices.

References