Abstract

Counting the number of patterns in graph is a fundamental problem in graph mining which has been extensively studied in both theory and practice.

In this proposal , we present three results in this area. The first result is a new cycle-counting algorithm in property-testing model. At present, there is a huge gap between the two lines of work. The current best theoretical result is an $O(m^{k/2})$-time algorithm (to obtain a constant-factor approximation with constant probability), where m is the number of edges in the graph, and t is the actual number of length-k cycles in the graph. But the algorithm performs poorly in practice, as shown in our experiments. On the other hand, those methods known to work well in practice have worse or no theoretical guarantees. Indeed, there are hard instances on which these methods would perform badly. Our algorithm with running time $O(\frac{m^{k/2}}{t})$ achieves the best theoretical result known to date. Meanwhile, our algorithm is also highly practical. The algorithm can also be extended to count all k-node Hamiltonian subgraph with the running time preserved.

The second result is a subgraph counting and sampling algorithm in a model which allows a linear-time preprocessing and it gives an estimation of the number of arbitrary pattern with constant error in running time $O(\frac{m^{\rho^*}}{t})$, where ρ^* is the fractional edge cover of the pattern. Note that this matches the bound of our algorithm in cycle counting.

We finally present an algorithm for join size estimation problem which is a special case of subgraph counting. A major challenge here is that join queries often come with ad hoc selection predicates that are only known at query time, thus there is no way to take the predicates into account when the statistics are collected. Our new algorithm called *two-level sampling* combines the advantages of three previous sampling methods while making further improvements. Besides the estimate itself, our algorithm also returns a confidence interval. We also extend the algorithm to support SUM and AVG aggregation with error guarantee. Both analytical and empirical comparisons show that the new algorithm outperforms all the previous algorithms on a variety of joins.