Distributed / Dynamic k-core Decomposition

Project Plan

Wong, Fu Wing
3035275111

Supervisor
Hubert T.H., Chan

Introduction

A lot of networks can be model as graphs, such as the friendship on a social media website, Internet connection, earthquake frequencies, etc. Those graphs represent the network, and operations can be done on the graph. Analysts can analyze a network through analyzing the corresponding graph(s). In real-world application, analysts may have to analyze a very large graph (with thousands or millions of vertices) and it is not practicable to analyze the graph with only the original information. Graph clustering is a tool used to analyze large graph and Core Clustering utilizes the core value of vertices retrieved from k-core decomposition algorithm which takes shorter time than other approaches [3]. K-core decomposition can also be applied to distinguish computer generated graphs from real-world networks [2].

There already exists a linear-time algorithm for k-core decomposition when the whole graph is known [1]. The algorithm keeps removing the set of vertices with the minimum degree and label them according to the degree when they are removed.

In the decentralized scenario, where every vertex only knows the connectivity of itself and its neighbors, without knowing the connectivity of the whole graph. It takes longer to compute the core value of a vertex because the core value of a vertex is not related to its neighbors only.

In real world application, time complexity is the main concern for graph mining as it may not require the actual core value. This project will focus on approximating core value instead of finding the exact value which is more efficient in terms of time complexity.

When calculating core values in the decentralized scenario. Vertices make use of the connectivity information of its neighbor. In the real world applications such as Internet connection, neighbors may not be honest, and information obtained from them may not be trustful. Using fake information for calculation will results in significant error. Therefore, calculating core value in the real-world yield security concerns and a secure protocol for communicating among vertices is required to ensure small differences between the calculated core value and the actual core value. This project will focus on designing a secure protocol to verify the correctness of the core value to be communicated. Each vertex must provide proofs to enforce the honesty of their core value calculated.

Theoretical Background

K-core decomposition is problem of calculating the core value of every vertices in a graph. A graph G contains vertices and edges, denoted as sets V and E respectively, and $G = (V, E)$. Two vertices u and v are neighbors if they are connected by an edge, i.e. $(u, v) \in E$. The degree of a vertex v is the number of neighbors of v. If a vertex v has a core value of k, then v has at least k neighbors, all having at least k neighbors, all having at least k neighbors, and so on. Formally, core value of a vertex v is the largest k such that there exist a sub-graph containing v and every vertices in the sub-graph have degree at least k.

There exists a $O(|V| + |E|)$ algorithm for k-core decomposition when the whole graph is known [1]. For decentralized scenario, there exists an algorithm that require $O(|V|)$ rounds of communication [1]. Core value can also be calculated by hosts, which controlling communications and calculation in a sub-graph [1].

Calculating approximation ratio is a way to show the performance of an approximation algorithm. The approximation ratio of an approximation algorithm is calculated by dividing the actual value by the calculated value. Overestimation will have a larger approximation ratio and smaller for underestimation. The more the approximation ratio is closer to 1, the more accurate the approximation algorithm achieved, and the better the approximation algorithm.

For secure protocol, a protocol is a specific way of communication among two nodes (vertices), with specific structure, data, and order of communication. The structure will keep consistent for all nodes using the same protocol.

Digital signature is a security measure to authenticate the source of the information. A vertex will compute its core value based on the authenticated core values obtained from its neighbor. This avoids using fake information from dishonest vertices which ensure correctness of the core value calculated.

Methodology

For designing the approximation algorithm for k-core decomposition, the project team will analyze graph properties, such as changes on the core value when removing a vertex. Graph properties will be analyzed through discovering new properties and analyzing properties from other related literatures. The project team will design an approximation algorithm by utilizing the analyzed graph properties.

Time complexity, space complexity, correctness and approximation ratio of the approximation algorithm will be analyzed and calculated. The project team will find and prove the time and space complexity of the algorithm mathematically. Correctness and the approximation ratio of the algorithm will also be proved mathematically if possible.

The algorithm will be implemented in Python, which is a programming language with a lot of user defined libraries, which will help when implementing secure protocol. The implementation will be tested using randomly generated graphs and an average approximation ratio will be obtained through testing.

For designing the secure protocol, the project team will design a protocol based on the designed approximation algorithm. Knowledge for cyber security such as digital signature and private key will be applied to implement authentication. The project team will design a structure for communication among vertices. The protocol will be implemented using Python, using library PyNaCl, which is a networking and cryptography library that provides variety of security and authentication methods such as digital signature.

Simulation will be done on the implemented protocol using randomly generated graphs. Four phases of testing will be done, with 0%, 10%, 25% and 33% dishonest vertices respectively. Dishonest vertices are randomly generated and allocated in the generated graphs. Dishonest vertices will fake the information in any means to fake other vertices to take their fake information as genuine. Average performance of the protocol will be estimated through simulation under different dishonest node ratio.

Project Schedule

Duration	Work Scheduled and Deliverables
1st Sep ~ 30th Sep	Background research
30th Sep	Deliverables of Phase 1 • Project Plan
1st Oct ~ 20th Nov	Investigate graph properties
21st Nov ~ 10th Jan	Design approximation algorithm utilizing found properties
20th Jan	Deliverables of Phase 2 • Approximation algorithm for k-core decomposition • Implementation of approximation algorithm
11th Jan ~ 28th Feb	Design secure protocol for decentralized computation of core value
1st Mar ~ 31st Mar	Implement the protocol in Python Simulation on randomized graphs
14th Apr	Deliverables of Phase 3 • Protocol design • Implementation of protocol

The project will be separated into two focuses, designing an approximation algorithm for k-core decomposition and designing a protocol for secure core value computation. The project will focus on designing approximation algorithm first, then designing the secure protocol.

For designing an approximation algorithm, the project team will investigate graph properties which will be applied when designing the approximation algorithm. The project team will then design an approximation algorithm based on the analyzed properties. The designed algorithm will be analyzed to proof its correctness and calculate its complexity and approximation ratio. In case of difficulties in analyzing its complexity and approximation ratio, the project team will implement the algorithm using Python and do simulations on randomly generated graphs, and the average approximation ratio will be estimated through simulation. The algorithm and the implementation will be the deliverables for phase 2.

For designing a secure protocol, the project team will design a protocol utilizing digital signature and other security measures. Space complexity of the protocol will be analyzed to ensure acceptable memory requirement. The project team will implement the protocol in Python. Average performance of the protocol will be estimated through simulation on randomly generated graphs. The protocol design and the implementation will be the deliverables for phase 3.

Risks and Challenges

When designing the approximation algorithm, the project team will analyze the approximation ratio of the algorithm, which may not be practical or accurate. Simulations have to be done to estimate the average approximation ratio. Number of simulation and scale of the graph have to be reasonable to support the average approximation ratio. Also, computer generated graphs may not reflect real-world scenario, the estimated approximation ratio may not reflect the performance of the algorithm when used in real-world applications.

For designing the secure protocol, the protocol has to handle information correctly in various situations. Random simulation may not contain the situation where dishonest vertices are closely related and cooperate together, and thus may not reflect the performance when implemented in real-world applications.

Summary

This project aims to design an efficient algorithm to approximate core value and design a secure protocol for calculating core value based on authenticated information retrieved. Performance of the algorithm and the protocol will be analyzed and calculated mathematically and experimentally. Graph properties will be analyzed to design an efficient approximation algorithm. The algorithm and the protocol will be implemented and tested on randomly generated graphs and average performance will be measured through testing.

References

[1] Alberto, M., Francesco, D.P., and Daniele, M. (2011). *Distributed k-Core Decomposition*. Retrieved from https://arxiv.org/pdf/1103.5320.pdf.

[2] Ignacio, A.H., Luca, D.A., Alain B., and Alessandro, V. (2005). *k-core decomposition: a tool for visualization of large scale networks*. Retrieved from https://arxiv.org/pdf/cs/0504107.pdf.

[3] Christos, G., Fragkiskos, D.M., Dimitrios, M.T., and Michalis, V. (2014). CoreCluster: *A Degeneracy Based Graph Clustering Framework*. Retrieved from https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/viewFile/8530/8397.