Scaling Up Structural Clustering to Large Probabilistic Graphs Using Lyapunov Central Limit Theorem

by

Joseph Howie
B.Sc., University of Victoria, 2020

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER'S OF SCIENCE

in the Department Computer Science

Scaling Up Structural Clustering to Large Probabilistic Graphs Using Lyapunov Central Limit Theorem

by

Joseph Howie
B.Sc., University of Victoria, 2020

Supervisory Committee

Dr. A. Thomo, Supervisor
(Department of Computer Science)

Dr. V. Srinivasan, Supervisor
(Department of Computer Science)

Supervisory Committee

Dr. A. Thomo, Supervisor
(Department of Computer Science)

Dr. V. Srinivasan, Supervisor
(Department of Computer Science)

ABSTRACT

In this thesis, we focus on structural clustering of probabilistic graphs, which comes with significant computational challenges and has, so far, resisted efficient solutions that are able to scale to large graphs, e.g. state-of-art can only handle graphs with a few million edges. We address the main bottleneck step of probabilistic structural clustering, computing the structural similarity of vertices based on their Jaccard similarity over the set of possible worlds of a given probabilistic graph. State-of-art used Dynamic Programming, a quadratic run-time algorithm, that does not scale to pairs of vertices of high degree. In this thesis we present a novel approach based on Lyapunov Central Limit Theorem. By using a carefully chosen set of random variables we are able to cast the computation of structural similarity to computing a one-tailed area under the Normal Distribution. Our approach has linear run-time as opposed to quadratic, and as such, it scales to much larger inputs. Extensive experiments show that our approach can handle massive graphs at web-scale which state-of-art cannot.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

My Fiancée and Parents, for their ongoing loving support through this journey.

Dr. Alex Thomo and Dr. Venkatesh Srinivasan, for their fantastic mentoring, support, and encouragement.

Vancouver Island Health Authority, for being a wonderful and understanding employer during the writing of this thesis.

DEDICATION

To my wonderful and loving Fiancée.

Chapter 1

Introduction

Probabilistic graphs are graphs in which each edge has a probability of existence. The uncertainty associated with this data structure allows for the modelling of many natural phenomena which have probabilistic interactions. For social networks, the influence users have over one another represents a probability of information passing between users of the network [20, 28]. With online dating networks, probabilistic graphs can model the likelihood that a user will visit another user's profile and whether they will send a message to said user [22, 30]. In the study of protein-protein interactions, experimental procedures determine the connections formed with a measurement uncertainty which is interpreted as the edge probability [25, 17].

For large graphs, a popular data mining operation is clustering, which groups similar vertices in the same cluster and separates dissimilar vertices into different clusters. A widely used approach for clustering deterministic graphs is the Structural Clustering Algorithm for Networks (SCAN) [31]. What distinguishes this algorithm from other clustering approaches is that it allows the clusters to overlap and furthermore introduces vertex classification. Namely, the algorithm defines three types of vertices: core, hubs, and outliers; and uses a metric to determine the type of each vertex in the network. Clusters are formed by grouping core vertices together based on maximal connectivity. After the clusters are formed, vertices that do not belong to any cluster are either hubs or outliers based on whether the vertex has edges connecting to multiple or just one cluster respectively. With these labels for vertices, great use has been made of the SCAN method on problems such as: community detection on population networks, fraud detection in financial networks, and on protein-protein interactions in biological networks [4, 23]. The SCAN method has a strong foundation in the literature, with many works expanding upon the original framework by modifying

the similarity metric, and implementing parallel processes [6, 26, 24, 3, 5, 27, 19]. To illustrate the SCAN method, consider the well known data set Zachary's karate club. In the ground truth cluster of this dataset, there are two distinct clusters, as displayed on Wikipedia. The clusters produced by SCAN depicted in Figure 1.1 are near identical to the ground truth for this dataset. The difference between SCAN and the ground truth clustering are that nodes 10, 29, and 32 were labelled as hubs, and nodes 12 and 25 were labelled outliers. In the ground truth clustering, 10, 25, 29, and 32 are part of the green cluster and 12 is part of the magenta cluster. SCAN labels these nodes as hubs and outliers because of their weak affiliation to the green and magenta clusters.

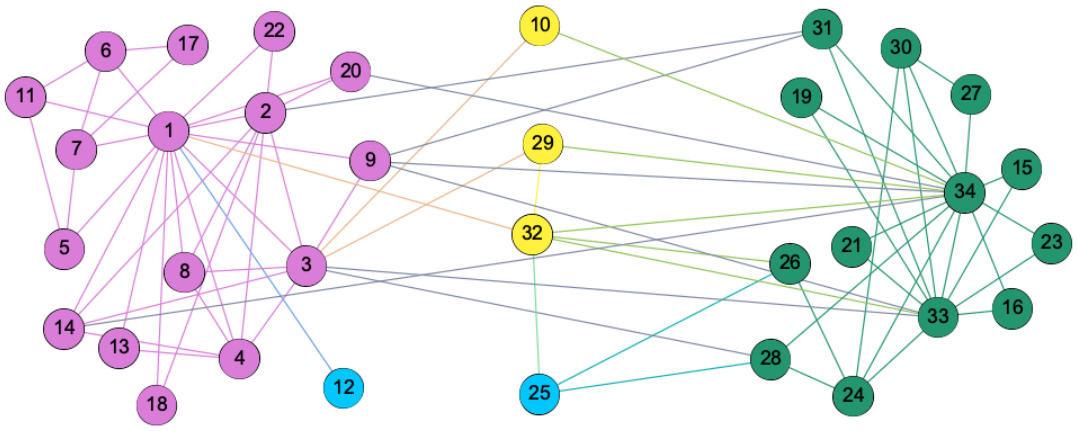

Figure 1.1: The network represented in this diagram is the well known dataset "Zachary's karate club". The nodes in magenta and green are two distinct clusters; the blue nodes are outliers, and the yellow nodes are hubs. The parameters were set to $(\varepsilon, \mu) = (0.2, 5)$. The results are nearly indistinguishable from the ground truth clustering displayed on the dataset's Wikipedia page.

One of the notable extensions to the SCAN method is the translation of the problem to the probabilistic setting. The probabilistic version of SCAN, called USCAN [24], introduces the probability of structural similarity for pairs of vertices which finds the probability of their structural similarity being above a threshold ε over all possible worlds. From there, the process continues in similar fashion to SCAN, with vertices added to the structural neighbourhood of a vertex if the probability of their structural similarity is above a threshold η. However, to iterate over all possible worlds would take an exponential amount of time. To combat this, the authors of USCAN devise a Dynamic Programming (DP) method for computing structural similarity. The DP solution runs in quadratic time, however, this is still not practical for computing structural similarity for pairs of vertices with many neighbours. This is indeed the case for most real world networks, where the maximum degree of vertices is well in the millions. The result is that the DP solution is unable to handle pairs of high degree vertices or large networks with many medium to high degree vertices.

In this thesis, we propose an efficient statistical approximation method for calculating the probability of structural clustering being above a given threshold. Our calculation is built on Lyapunov's version of the Central Limit Theorem (CLT) [32] which works for non-identically distributed but independent random variables. The crux of our method is to express the Jaccard similarity of a pair of vertices in a probabilistic graph as the sum of a special set of random variables which we prove to give the structural similarity of the pair. This needs to be done carefully in order to properly satisfy the highly technical conditions stipulated in the Lyapunov CLT. Once we achieve the expression of structural similarity in terms of a sum of random variables that satisfies Lyapunov CLT, the problem then becomes that of computing a one-tailed area under the Normal Distribution, which is easily done. Our approach runs in linear time with respect to the number of neighbours between a pair of vertices connected by an edge. Since, the Central Limit Theorem is an approximation to the true distribution of the sum of random variables, our approach also yields an approximate solution. However, it is well known that CLT produces a very tight approximation in practice for large numbers. This is also what we observe for our problem. For pairs of vertices with a number of neighbours in the few hundreds, which is where DP starts being impractical, our CLT approach produces approximations that are indistinguishable from numbers produced by DP. We give theoretical bounds on the quality of the approximation using the Berry-Essen Theorem [12, 32].

To reiterate, the complexity of our method reduces to linear time from the quadratic

time achieved by DP, with our method running up to 130 times faster for datasets that DP can handle. Furthermore, our method achieves significantly greater scalability, clustering graphs with up to half a billion edges in less than an hour. Meanwhile USCAN was not able to complete on datasets with more than 20 million edges.

We give in the following a summary of our contributions.

- We derive an efficient approximate method to calculate the probability of structural similarity with the Lyapunov Central Limit Theorem, which gives practically identical results to the exact computation from the Dynamic Programming solution. We give a proof of correctness and bound the quality of the approximation solution.

- We derive the time complexity of our method and validate its time improvement over USCAN through experimentation on real world datasets. We show that our method yields up to a 130x improvement over the exact calculation.

- The reduction in time complexity allows our method to scale up to much larger datasets than the state-of-art USCAN algorithm. Our algorithm finishes in less than an hour on graphs with over half a billion edges.

Chapter 2

Related Works

Since the original publication of the SCAN paper, many improvements and additions have been built on top of SCAN. Some authors [35, 6, 29] insert parallel processing paradigms to improve practical performance of calculating the structural similarity. The paper by Che et. al. [5] develops a min-max pruning method to improve the performance on detecting core vertices. For Seo et. al. [27] they demonstrate how detecting and merging local clusters scales the performance of the clustering framework. In Chang et. al. [3] they prove that the SCAN algorithm is worst case optimal and introduce new scaleable techniques that practically improve the structural clustering procedure. The structural similarity formula is an integral component of the SCAN framework and many authors have explored variations of this ratio to improve performance. Recently, papers have been adopting the Jaccard similarity as the equation for structural similarity [26, 24, 3]. All these methods discussed above are for deterministic graphs, for SCAN to work on probabilistic graphs requires an extension of the basic definitions.

In Qiu et. al. [24] the authors derive new definitions that construct a probabilistic structural clustering algorithm, called USCAN. The key idea of USCAN is the notion of probability of structural similarity, which calculates the probability that a pair of vertices connected by an edge have a structural similarity (Jaccard similarity) above ε. Hence rather than having $\varepsilon-$neighbourhoods, where nodes in the set have structural similarities larger than ε; there are $(\varepsilon, \eta)-$reliable neighbourhoods, where vertices in the set have a probability greater than η that the edge pair has a structural similarity greater than ε over all possible worlds. Then using the reliable neighbourhoods, reliable core vertices are determined and the algorithm from there follows the remainder of the SCAN protocol.

The novelty of USCAN is the definition of the probability of structural similarity, which the authors show can be calculated in polynomial time by dynamically iterating over classes of neighbourhoods for each edge. However, the Dynamic Program that calculates the probability is the bottleneck of the entire program, taking quadratic time with respect to the union of neighbourhoods between the edge. As a result of the time complexity, USCAN can not scale to large graph datasets with over 20 million edges.

Liang et. al. [19] claim to improve the time complexity of USCAN with a different formulation of the calculation for the probability of structural similarity. The proposed algorithm called ProbSCAN, displayed an equation for the bottleneck process that calculates the probability of structural similarity in linear time with respect to the union of neighbourhoods, rather than the quadratic time in USCAN. The paper does not give a proof, and unfortunately, as we show in Appendix A, their approach is incorrect. Hence, we regard USCAN as the current state-of-art algorithm for structural clustering of probabilistic graphs. Unfortunately, the Dynamic Programming algorithm for calculating the probability of structural similarity does not scale to large graphs due to its quadratic time complexity.

There are also other clustering frameworks that apply to probabilistic graphs. Some methods extend the framework of K-Means clustering to probabilistic graphs by maximizing the average connection probability in each of the k clusters [2, 15]. With Halim et. al. [14] the approach for clustering probabilistic graphs analyzes the surrounding neighbourhood of vertices for an edge and calculates a weighted average to decide whether the edge in question passes a static threshold cut. The authors of [9] utilize a variation of Jaccard similarity that uses random walks to identify similarity, then feeds the probabilities into an encoder and deep learning network with a Gaussian embedding to discover the resulting clusters. Other methods for clustering non-graph data include a density-based algorithm (DBSCAN) which identifies dense regions in the data [33, 34, 13]. These density-based algorithms are similar to SCAN, as they identify cores, border, and noise nodes in an effort to classify the different data points based on their neighborhood structures. There are also algorithms that use sampling techniques to find clusters in probabilistic graphs [16, 21]. Our approach is uniquely distinct from these works in terms of problem definition and techniques used. We focus on the family of structural clustering which allows the clusters to overlap and classifies the vertices into three types of nodes, namely, cores, hubs, and outliers, and uses a metric to determine the type of each vertex in the network. From the

techniques point of view, we utilize the Lyapunov CLT which has not been used before for the problem of clustering probabilistic graphs. Lyapunov CLT has been used in [8, 12, 11, 10] for problems unrelated to clustering, such as core and truss decomposition, non-linear system control, and further applications relevant to social science and economics. The modelling of those problems in terms of Lyapunov CLT does not bear any similarity to our approach.

Chapter 3

Background

Definition 1 (Probabilistic Graph)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$ be an un-directed probabilistic graph s.t. $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and $p : \mathcal{E} \to (0, 1]$.*

Definition 2 (Possible Worlds)**.** *Unlike deterministic graphs, probabilistic graphs represent possible worlds, which all have different probabilities of occurring. A graph $G = (\mathcal{V}, E)$, where $E \subseteq \mathcal{E}$, is a possible world of $\mathcal{G}$, where the probability of occurring from $\mathcal{G}$ is given as:*

$$P[G|\mathcal{G}] = \prod_{e \in E} p(e) \prod_{e \in \mathcal{E} \setminus E} (1 - p(e)) \tag{3.1}$$

and we say $G \sqsubseteq \mathcal{G}$, meaning G is a possible world of $\mathcal{G}$. Hence, for a probabilistic graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, there are $2^{|\mathcal{E}|}$ possible worlds of $\mathcal{G}$, where each edge $e \in \mathcal{E}$ has probability $p(e)$ of existing in a possible world G.

3.1 Definitions

Our method uses the same framework as pSCAN and USCAN [24, 3] which in turn are based on SCAN [31]. Specifically, we present the following definitions.

Definition 3 (Structural Neighbourhood [31])**.** *Given a deterministic graph $G = (\mathcal{V}, E)$, the structural neighbourhood, N_u, of a vertex $u \in \mathcal{V}$, is a closed neighbourhood, meaning $N_u = \{v \in \mathcal{V} \mid (u, v) \in E\} \cup \{u\}$. That is, the structural neighbourhood of u, contains u by definition.*

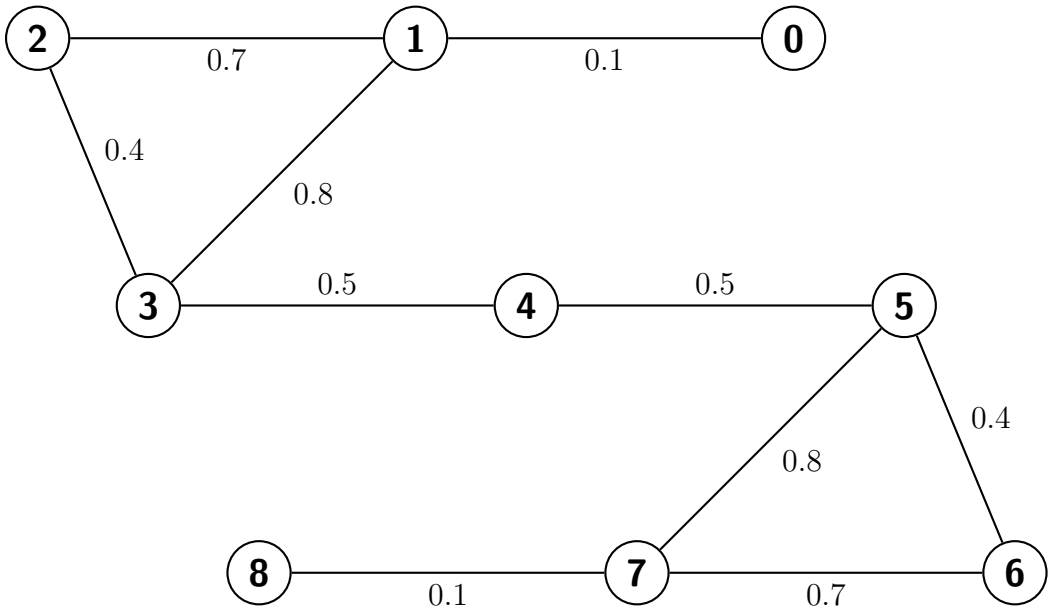

Figure 3.1: Probabilistic graph example $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$. Let $\overline{G} = (\mathcal{V}, \mathcal{E})$ be the maximal possible world of $\mathcal{G}$, meaning $\overline{G}$ is the possible world where all $e \in \mathcal{E}$ are present.

Figure 3.1 is an example of a probabilistic graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, with nine vertices and ten edges. For this graph, $\mathcal{G}$, there are $2^{|\mathcal{E}|} = 2^{10} = 1024$ distinct possible worlds. Let $\overline{G} = (\mathcal{V}, \mathcal{E})$ be the possible world graph where all edges in $\mathcal{E}$ are present in $\overline{G}$. We call this possible world the maximal possible world of $\mathcal{G}$. Moreover, we denote the structural neighbourhoods of the maximal possible world as $\overline{N_u}$, $\forall u \in \mathcal{V}$.

Example 1. *Consider vertices 1 and 3 in the deterministic graph $\overline{G} = (\mathcal{V}, \mathcal{E})$ from Figure 3.1. The structural neighbourhoods of vertices 1 and 3 are $\overline{N_1} = \{0, 1, 2, 3\}$ and $\overline{N_3} = \{1, 2, 3, 4\}$ respectively.*

Definition 4 (Structural Similarity [24])**.** *Given a deterministic graph $G = (V, E)$, the structural similarity between vertices u and v, $\sigma(u, v)$, is defined as the number of common structural neighbours between u and v, divided by the number of structural neighbours in either u or v, that is*

$$\sigma(u, v) = \frac{|N_u \cap N_v|}{|N_u \cup N_v|} \tag{3.2}$$

where equation 3.2 is the Jaccard similarity, which is an effective measure for structural clustering in networks [26, 24, 3].

Example 2. *Consider the edge $(1, 3)$ in the deterministic graph $\overline{G}$ from Figure 3.1. The structural neighbourhoods for this edge are given in example 1. The sizes of the intersection and union are then $|\overline{N_1} \cap \overline{N_3}| = |\{1, 2, 3\}| = 3$ and $|\overline{N_1} \cup \overline{N_3}| = |\{0, 1, 2, 3, 4\}| = 5$, hence $\sigma(1, 3) = \frac{3}{5}$.*

Definition 5 ($\varepsilon-$Structural Similarity [31])**.** *Given a deterministic graph $G = (V, E)$, an edge $(u, v) \in E$, and threshold ε, u is $\varepsilon-$structural similar to v if $\sigma(u, v) \geq \varepsilon$.*

Example 3. *The $\varepsilon-$structural similarity of the edge $(1, 3)$ is $\sigma(1, 3) = \frac{3}{5}$ in $\overline{G}$ from Figure 3.1. Hence, if $\varepsilon = \frac{1}{2}$, then vertex 1 is $\varepsilon-$structural similar to 3 and vice versa.*

Thus far, all the definitions have to do with deterministic graphs, and largely come from SCAN [31]. Next we introduce key ideas that hail from USCAN [24] which are designed to elevate the SCAN model to probabilistic networks.

Definition 6 (Probability of Structural Similarity [24])**.** *Given a similarity threshold $\varepsilon \in (0, 1]$, the probability that $\sigma(e) \geq \varepsilon$ is the sum of the probabilities over all possible*

worlds $G \sqsubseteq \mathcal{G}$, such that the structural similarity of $e = (u, v)$ is no less than ε in G. That is,

$$P[e, \varepsilon] = \sum_{G \sqsubseteq \mathcal{G}} P[G|\mathcal{G}] \cdot \Theta(\sigma(e) \geq \varepsilon) \tag{3.3}$$

where $\Theta(\sigma(e) \geq \varepsilon)$ is an indicator function that equals 1 when $\sigma(e) \geq \varepsilon$, and 0 otherwise.

Example 4. *Consider the edge $(1, 3)$ in the probabilistic graph $\mathcal{G}$ from Figure 3.1. There are a total of 1024 possible worlds of $\mathcal{G}$, each of which occurs with probability derived from Equation 3.1 based on the included edges. Suppose that $\varepsilon = \frac{1}{2}$, then only the possible worlds where $\sigma(1, 3) \geq \frac{1}{2}$ contribute to the sum. Using equation 3.3, $P\left[(1, 3), \frac{1}{2}\right] = 0.7784$.*

We can now define the notion of reliable neighbourhoods and reliable core vertices using Definition 6.

Definition 7 (Reliable Structural Similarity [24])**.** *Given an edge $e = (u, v)$ and a threshold η, u is reliable structural similar to v if $P[e, \varepsilon] \geq \eta$.*

Example 5. *Consider $\mathcal{G}$ in Figure 3.1, the probability of structural similarity for edge $(1, 3)$ is $P\left[(1, 3), \frac{1}{2}\right] = 0.7784$. Then if $\eta = \frac{2}{3}$, vertices 1 and 3 are reliable structurally similar to each other since $P\left[(1, 3), \frac{1}{2}\right] \geq \frac{2}{3}$.*

Definition 8 ($(\varepsilon, \eta)-$Reliable Neighbourhood [24])**.** *Given a similarity threshold $\varepsilon \in (0, 1]$, and a probability threshold $\eta \in (0, 1]$, the $(\varepsilon, \eta)-$reliable neighbourhood of u is the subset of vertices in $\overline{N_u}$ such that $P[e, \varepsilon] \geq \eta$, meaning the set is given by $N_u(\varepsilon, \eta) = \{v \in \overline{N_u} \mid P[e, \varepsilon] \geq \eta\}$.*

Example 6. *When $\eta = \frac{2}{3}$ and $\varepsilon = \frac{1}{2}$, then the $(\varepsilon, \eta)-$reliable neighbourhoods in Figure 3.1 are: $N_0\left(\frac{1}{2}, \frac{2}{3}\right) = \{0\}$, $N_1\left(\frac{1}{2}, \frac{2}{3}\right) = \{1, 2, 3\}$, $N_2\left(\frac{1}{2}, \frac{2}{3}\right) = \{1, 2\}$, $N_3\left(\frac{1}{2}, \frac{2}{3}\right) = \{1, 3\}$, $N_4\left(\frac{1}{2}, \frac{2}{3}\right) = \{4\}$, $N_5\left(\frac{1}{2}, \frac{2}{3}\right) = \{5, 7\}$, $N_6\left(\frac{1}{2}, \frac{2}{3}\right) = \{6, 7\}$, $N_7\left(\frac{1}{2}, \frac{2}{3}\right) = \{5, 6, 7\}$, and $N_8\left(\frac{1}{2}, \frac{2}{3}\right) = \{8\}$.*

Notice that for all $(\varepsilon, \eta)-$reliable neighbourhoods every vertex u is contained in its own $(\varepsilon, \eta)-$reliable neighbourhoods $N_u(\varepsilon, \eta)$. Recall each vertex has a minimum structural neighbourhood size of one by the definition, and every vertex is in every possible world. Consider that each node is connected to itself via *self loop*, then $P[(u, u), \varepsilon] = 1$, $\forall \varepsilon$. Therefore, all vertices are in their own $(\varepsilon, \eta)-$reliable neighbourhood, by definition.

Definition 9 ($(\varepsilon, \eta, \mu)-$Reliable Core Vertex [24])**.** *Given a similarity threshold $\varepsilon \in (0, 1]$, a probability threshold $\eta \in (0, 1]$, and an integer threshold $\mu \geq 2$, a vertex u is a $(\varepsilon, \eta, \mu)-$reliable core vertex if $|N_u(\varepsilon, \eta)| \geq \mu$.*

Example 7. *When $\mu = 3$, and with the $\left(\frac{1}{2}, \frac{2}{3}\right)-$reliable neighbourhoods from the previous example, only vertices 1 and 7 are reliable core vertices; because they are the only nodes with reliable neighbourhoods that contain three or more elements.*

Definition 10 (Reliable Structure-reachable [24])**.** *Given parameters $\varepsilon \in (0, 1]$, $\eta \in (0, 1]$, and $\mu \geq 2$, vertex v is reliable structure-reachable from vertex u if there is a sequence of vertices $v_1, \ldots, v_l \in V$ with $l \geq 2$, such that:*

- *$v_1 = u$ and $v_l = v$;*
- *$v_1, v_2, \ldots, v_{l-1}$ are reliable core vertices;*
- *$v_{i+1} \in N_{v_i}(\varepsilon, \eta)$ for each $i \in [1, l-1]$*

For v to be reliable structure-reachable from u means there is a path of reliable core vertices from u that reaches v. Notice from the definition, v does not need to be a reliable core vertex. The only requirement of v is that it belongs to the reliable neighbourhood set of the last reliable core vertex in the path of reliable core vertices starting from u.

Example 8. *Consider the probabilistic graph in Figure 3.1. Let the parameters $(\varepsilon, \eta, \mu) = \left(\frac{1}{2}, \frac{2}{3}, 3\right)$. Since the only core vertices are 1 and 7, which are disconnected for one another; then all reliable structure-reachable path are merely a single edge from the cores to each of their $(\varepsilon, \eta)-$reliable neighbours. Thus, both vertices 2 and 3 are reliable structure-reachable from 1; and both vertices 5 and 6 are reliable structure-reachable from 7.*

From the definitions above, USCAN [24] formulated the problem of structural clustering on probabilistic graphs as follows.

Definition 11 (The Probabilistic Graph Clustering Problem [24])**.** *Given a probabilistic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$ and parameters $\varepsilon \in (0, 1]$, $\eta \in (0, 1]$, and $\mu \geq 2$, the problem of probabilistic graph clustering is to compute the set $\mathbb{C}$ of reliable clusters in $\mathcal{G}$. Each reliable cluster, $C \in \mathbb{C}$, must contain at minimum two vertices and satisfy:*

- ***Maximality****: for each reliable core vertex $u \in C$, all vertices that are reliable structure-reachable from u must be in C.*
- ***Connectivity****: for any two vertices $v_1, v_2 \in C$, there exists $u \in C$ such that both v_1, v_2 are reliable structure-reachable from u.*

Example 9. *Consider the probabilistic graph, $\mathcal{G}$, in Figure 3.1 with $(\varepsilon, \eta, \mu) = \left(\frac{1}{2}, \frac{2}{3}, 3\right)$. By maximality, and connectivity, the nodes 1,2,3 and 5,6,7 form two distinct clusters.*

Notice with these definitions it is possible for a non-core vertex to belong to multiple clusters at once. Suppose the example graph in Figure 3.1 had an additional non-core vertex 9, and this vertex was connected to the graph in such a way that 9 is in both the $(\varepsilon, \eta)-$reliable neighbourhoods of 1 and 7. Then vertex 9 would be reliable structure-reachable from both 1 and 7, and therefore would be apart of both the clusters formed in Example 9. Hence the cluster sets produced are not partitions since overlaps are permitted.

Definition 12 (Hubs and Outliers [24])**.** *Given the set $\mathbb{C}$ of reliable clusters in a probabilistic graph $\mathcal{G}$, a vertex u that is not in any reliable cluster in $\mathbb{C}$ is a hub vertex if it connects two or more reliable clusters, and it is an outlier vertex otherwise.*

It is possible that for a given probabilistic graph, $\mathcal{G}$, that no hubs or outliers are found after identifying the set of clusters $\mathbb{C}$ with specified values for the parameters η, ε, and μ.

Example 10. *From Example 9 we have two clusters $C_1 = \{1, 2, 3\}$ and $C_2 = \{5, 6, 7\}$. Since vertex 4 is not in any cluster, but is attached by an edge to clusters C_1 and C_2 via vertices 3 and 5 respectively; therefore, vertex 4 is a hub. Additionally, vertices 0, and 8 are also not in any cluster. Unlike 4, nodes 0 and 8 only connect to one cluster each via edges to 1 and 7 respectively. Therefore, vertices 0 and 8 are outliers.*

DP algorithm To overcome the $O(2^{|\mathcal{E}|})$ complexity for computing the probability of structural similarity, the authors of USCAN derive a clever approach of computing the probability in equation 3.3. After all of the optimization observations are applied, the DP algorithm runs in $O(|\overline{N_u} \cup \overline{N_v}|^2)$ time [24].

3.2 Framework

The framework for producing clusters, hubs, and outliers is inherited from pSCAN [3] and subsequently USCAN [24]. The distinction between our method and the state-of-art resides in the function ComputePr(u, v, ε). In USCAN, ComputePr(u, v, ε) is the DP calculation to determine the probability of structural similarity. For our method, if an edge has a neighbourhood union size that meets a preset threshold parameter, then Lyapunov CLT is used to find the value of $P[(u, v), \varepsilon]$; the time complexity of using Lyapunov CLT is linear in the neighbourhood union size.

Algorithm 1 Clustering Framework

```
1:  procedure Framework(G = (V, E, p))
2:      Initialize G_c = (V, ∅)
3:      ∀u ∈ V, initialize u as a non-core vertex
4:      for each u ∈ V do
5:          if IsReliableCore(u) then  Label u as a core vertex
6:          for each v ∈ N_u(ε, η) do
7:              if IsReliableCore(v) then Add (u, v) to G_c
8:      C_c ← the set of connected components in G_c
9:      C ← {C_c ∪_{u∈C_c} N_u(ε, η) | C_c ∈ C_c}
10:     return C
11: procedure IsReliableCore(u)
12:     N_u(ε, η) ← ∅
13:     for each v ∈ N_u \ {u} do
14:         ComputePr(u, v, ε)
15:         if P[(u, v), ε] ≥ η then Add v to N_u(ε, η)
16:     if |N_u(ε, η)| ≥ μ then return True
17:     else return False
```

Algorithm 1 starts by initializing an edgeless graph G_c with all the vertices in $\mathcal{G}$, line 2. Each vertex becomes marked as a non-core vertex in line 3. Then the algorithm checks whether each vertex is a reliable core vertex, lines 4-5. For each reliable core vertex u found, any reliable neighbours of that vertex that are also reliable core vertices v, have their corresponding edge (u, v) added to G_c, lines 6-7. The graph G_c now exclusively contains edges that connect reliable core vertices together. Thus, the connected components of G_c begin to form the clusters in $\mathbb{C}$, line 8. However, by the definition of reliable structure-reachable, the last vertex in the path need not be a reliable core. Hence, each vertex, u, in each cluster must incorporate their $(\varepsilon, \eta)-$reliable neighbourhood into their cluster as well, line 9.

For Algorithm 1, the proof of correctness is given in [3]. Lines 2-8 take $O(m)$ time, if ComputePr(u, v, ε) is constant. However, ComputePr(u, v, ε) under the DP formulation takes $O(|\overline{N_u} \cup \overline{N_v}|^2)$. From the analysis in Qiu et. al. [24], the entire clustering process takes $O(d_{max}^2 \times \alpha \times m)$, where α is the arboricity of the graph which comes from the original proof in [3], and m is the number of edges in $\mathcal{G}$ [7].

3.3 Challenges

The method proposed in this thesis aims to reduce the time complexity of the bottleneck process for clustering probabilistic graphs. In the USCAN algorithm, the process that takes the most time is the DP algorithm that calculates $P[(u, v), \varepsilon]$. The DP algorithm takes $O(|\overline{N_u} \cup \overline{N_v}|^2)$ time for a single edge $(u, v) \in \mathcal{E}$. In our proposed algorithm, our Lyapunov CLT approach computes $P[(u, v), \varepsilon]$ in $O(|\overline{N_u} \cup \overline{N_v}|)$ time.

Chapter 4

Proposed Algorithm

We now describe our proposed algorithm, NUSCAN, for computing $P[(u,v),\varepsilon]$. In Section 4.1, we show the core technique of NUSCAN, which makes use of Lyapunov Central Limit Theorem for computing $P[(u,v),\varepsilon]$. Then, we decribe the main steps of NUSCAN in Section 4.2. Finally, in Section 4.3, we derive bounds on the quality of the solution for NUSCAN.

4.1 Structural Similarity using Lyapunov CLT

Theorem 1 (Lyapunov CLT)**.** *Let $\xi_1, \xi_2, \ldots, \xi_n$ be a sequence of independent, but non-identically distributed random variables, each with finite expected value μ_k and variance σ_k^2. Let*

$$s_n^2 = \sum_{k=1}^{n} \sigma_k^2 \tag{4.1}$$

Lyapunov CLT states if

$$\lim_{n\to\infty} \frac{1}{s_n^{2+\delta}} \sum_{k=1}^{n} E[|\xi_k - \mu_k|^{2+\delta}] = 0 \tag{4.2}$$

for some $\delta > 0$, then $\frac{1}{s_n}\sum_{k=1}^{n}(\xi_k - \mu_k)$ converges in distribution to a standard normal random variable [12, 32].

Remark. Showing that Lyapunov CLT can be applied to calculate the probability of structural similarity is technical and requires care. Suppose that $J_{u,v}$ is the random variable that represents the value of $\sigma(u,v)$ over all possible worlds. Then the

calculation of $P[(u,v),\varepsilon]$ becomes equivalent to the expression $P[J_{u,v} \geq \varepsilon] \times p(u,v)$. We show that the random variable $J_{u,v}$ can be expressed as sum of independent, non-identically distributed random variables. Then we determine the choice of δ required to satisfy the limit condition of Lyapunov CLT. In what follows, we put forth a series of definitions and lemmas needed to derive an expression for the probability of structural similarity, $P[(u,v),\varepsilon]$, allowing the use of Lyapunov CLT.

To start, we define for each edge in the probabilistic graph a Bernoulli Random Variable that indicates whether a given edge is present in any possible world in accordance with its edge probability.

Definition 13 (Edge Random Variable)**.** *Given a probabilistic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, let $X_{u,v}$ be a Bernoulli Random Variable that determines whether an edge $(u,v) \in \mathcal{E}$ is present in an arbitrary possible world G, meaning*

$$X_{u,v} = \begin{cases} 1, & p(u,v) \\ 0, & 1 - p(u,v) \end{cases} \tag{4.3}$$

where $p(u,v)$ is the probability that edge (u,v) is present in any possible world $G \sqsubseteq \mathcal{G}$. We call $X_{u,v}$ an Edge Random Variable (ERV).

The sequence of all ERV is by definition a sequence of independent and non-identically distributed random variables since each ERV may have a different value of $p(u,v)$.

Example 11. *Consider the probabilistic graph in Figure 3.1, and specifically the edge $(1,3)$. The probability that $(1,3)$ is in any arbitrary possible world G is $p(1,3) = 0.8$. Then $X_{1,3}$ has the value 1 with probability 0.8 and is 0 otherwise.*

Next, we construct a special sequence of vertices from the combined neighbourhood sets between an edge (u,v). We will use this sequence in order to derive the distribution of $J_{u,v}$.

Definition 14 (Neighbourhood Edge Sequence)**.** *Given a probabilistic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, $\forall (u,v) \in \mathcal{E}$ let $N^*_{uv} = (\overline{N_u} \cap \overline{N_v}) \setminus \{u,v\}$ be the set of common neighbours excluding the vertices $\{u,v\}$ and let $\widetilde{N_{uv}} = (\overline{N_u} \cup \overline{N_v}) \setminus \{u,v\}$ be the set of all neighbours between u and v while excluding $\{u,v\}$. Let Y_{uv} be an ordered sequence of the elements in $\widetilde{N_{uv}}$ such that $y_{2i} = y_{2i+1}$ $\forall i \in [0, q-1]$, where $q = q_{u,v} = |N^*_{uv}|$ and*

$\forall i \in [0, 2q-1],\ y_i \in N^*_{uv}$; *and* $\forall j \in [2q, r-1],\ y_j \in \widetilde{N_{uv}} \setminus N^*_{uv}$ *where* $r = |\widetilde{N_{uv}}| + |N^*_{uv}|$. *Therefore,*

$$Y_{uv} : y_0, y_1, \ldots, y_{2q-1}, y_{2q}, \ldots, y_{r-1} \tag{4.4}$$

Without loss of generality, the elements in $\overline{N_u} \setminus \overline{N_v}$ *appear in the sequence before the elements that are in* $\overline{N_v} \setminus \overline{N_u}$.

For each edge (u, v) in a probabilistic graph, there is an associated Neighbourhood Edge Sequence Y_{uv} with three distinct sections. The first section contains the elements that are in the maximal neighbourhoods of both u, v (excluding u, and v themselves). The elements in the first section are duplicated to signify membership to both maximal neighbourhoods. The second section of the sequence contains elements exclusively belonging to the maximal neighbourhood of u (this also excludes u, and v). The third section of Y_{uv} holds elements only in the maximal neighbourhood of v (again excluding u, and v). In the second and third sections, the elements only appear once as opposed to the first section where elements are repeated. The reason is to symbolized ownership of the vertex to only one maximal neighbourhood set; contrast to the first section where the represented vertex belonged to both maximal neighbourhood sets. It is possible that any of the three sections of Y_{uv} do not contribute any elements. The three sections of Y_{uv} discussed above derive from three sets N^*_{uv}, $\overline{N_u} \setminus \overline{N_v}$, and $\overline{N_v} \setminus \overline{N_u}$ respective to the outlined order above, which for some $(u, v) \in \mathcal{E}$ may be empty. Therefore, for some (u, v) if $N^*_{uv} = \overline{N_u} \setminus \overline{N_v} = \overline{N_v} \setminus \overline{N_u} = \emptyset$, then Y_{uv} is an empty sequence.

Example 12. *Consider the probabilistic graph in Figure 3.1, and the edge* $(1, 3)$. *In* $\overline{G}$, *the edge* $(1, 3)$ *has maximal structural neighbourhoods* $\overline{N_1} \setminus \{1, 3\} = \{0, 2\}$ *and* $\overline{N_3} \setminus \{1, 3\} = \{2, 4\}$. *Then* $\widetilde{N_{13}} = \{0, 2, 4\}$ *and* $N^*_{13} = \{2\}$, *and thus* Y_{13} *is* $2, 2, 0, 4$.

From the sequence Y_{uv}, we define a homomorphic sequence of Edge Random Variables for the edges represented in the original sequence.

Definition 15 (Correspondence Sequence)**.** *Given* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, $\forall (u, v) \in \mathcal{E}$ *with Neighbourhood Edge Sequence* Y_{uv}, *let* χ_{uv} *be a sequence of ERV in one-to-one correspondence to* Y_{uv} *under the following definition,*

$$\chi_{uv} : X_{y_0,u}, X_{y_1,v}, \ldots, X_{y_{2q-2},u}, X_{y_{2q-1},v}, X_{y_{2q},z}, \ldots, X_{y_{r-1},z} \tag{4.5}$$

where $X_{y_i,z}$ *is the ERV for the edge* (y_i, z), *and* z *is either* u *or* v *as defined by* Y_{uv}.

Example 13. *Suppose we have the sequence* $Y_{13} : 2, 2, 0, 4$. *Then the Correspondence Sequence is* $\chi_{13} : X_{2,1}, X_{2,3}, X_{0,1}, X_{4,3}$.

Unlike Y_{uv} where elements were integers and not necessarily unique, each element of χ_{uv} is a unique random variable that represents the same corresponding edge in Y_{uv}. Now for each edge (u, v), we have a sequence of ERV that represents edges in both maximal neighbourhoods of u and v. We exploit the ordering of this sequence to derive two random variables that constitute the numerator and denominator of $J_{u,v}$.

Lemma 1. *Given a probabilistic graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, $(u, v) \in \mathcal{E}$, *and sequence* χ_{uv}, *the random variable that represents* $|N_u \cap N_v|$ *over all possible worlds is defined as,*

$$\mathcal{M}_{u,v} = 2 + \sum_{i=0}^{q-1} X_{y_{2i},u} X_{y_{2i+1},v} \tag{4.6}$$

Proof. The 2 is for the presence of u, v, and the sum contributes elements possibly in the intersection only when both ERV are equal to 1. □

Example 14. *In Example 13, the Correspondence Sequence for edge* (u, v) *was* $\chi_{13} : X_{2,1}, X_{2,3}, X_{0,1}, X_{4,3}$; *then* $\mathcal{M}_{1,3} = 2 + X_{2,1}X_{2,3}$.

Lemma 2. *Given a probabilistic graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, $(u, v) \in \mathcal{E}$, *and sequence* χ_{uv}, *the random variable that represents* $|N_u \cup N_v|$ *over all possible worlds is defined as,*

$$\mathcal{N}_{u,v} = 2 + \sum_{i=0}^{q-1} \max\left(X_{y_{2i},u}, X_{y_{2i+1},v}\right) + \sum_{\substack{j=2q \\ z \in \{u,v\}}}^{r-1} X_{y_j,z} \tag{4.7}$$

Proof. The 2 is once again for the presence of u, and v. The first sum counts intersecting elements if at least one of the ERV is equal to 1. The second sum counts elements outside the intersection when their ERV are equal to 1. □

Example 15. *In Example 13, the Correspondence Sequence was* $\chi_{13} : X_{2,1}, X_{2,3}, X_{0,1}, X_{4,3}$; *then* $\mathcal{N}_{1,3} = 2 + \max\left(X_{2,1}, X_{2,3}\right) + X_{0,1} + X_{4,3}$.

For edge (u, v), the random variables $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$ represent respectively, the size of the intersection and union of structural neighbourhoods over all possible worlds. The two random variables $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$ assume that the edge u, and v exist. Hence any further derived random variables inherit this assumption of existence. Using the random variables $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$, we derive the probabilistic Jaccard similarity $J_{u,v}$.

Corollary 1. *Given a probabilistic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, $(u, v) \in \mathcal{E}$, sequence χ_{uv}, and random variables $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$, the Jaccard similarity over all possible worlds is,*

$$J_{u,v} = \frac{\mathcal{M}_{u,v}}{\mathcal{N}_{u,v}} \tag{4.8}$$

Proof. Lemmas 1 and 2 proved that $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$ are the random variable representation of the intersection and union of the structural neighbourhoods of u, and v over all possible worlds. Therefore it follows the ratio of $\mathcal{M}_{u,v}$ to $\mathcal{N}_{u,v}$ is exactly the random variable representation of $\sigma(u, v)$ over all possible worlds. □

Example 16. *From Examples 14 and 15, $\mathcal{M}_{1,3} = 2 + X_{2,1}X_{2,3}$ and $\mathcal{N}_{1,3} = 2 + \max(X_{2,1}, X_{2,3}) + X_{0,1} + X_{4,3}$. Therefore for edge $(1, 3)$, the probabilistic Jaccard similarity is,*

$$J_{1,3} = \frac{\mathcal{M}_{1,3}}{\mathcal{N}_{1,3}} = \frac{2 + X_{2,1}X_{2,3}}{2 + \max(X_{2,1}, X_{2,3}) + X_{0,1} + X_{4,3}}$$

We now have a random variable representation of the structural similarity measure $\sigma(u, v)$ over all possible worlds, called the probabilistic Jaccard similarity $J_{u,v}$. Next we determine the probability that $J_{u,v} \geq \varepsilon$, where $\varepsilon \in (0, 1]$.

$$P[J_{u,v} \geq \varepsilon] = P\left[\frac{\mathcal{M}_{u,v}}{\mathcal{N}_{u,v}} \geq \varepsilon\right] = P[\mathcal{M}_{u,v} - \varepsilon\mathcal{N}_{u,v} \geq 0] \tag{4.9}$$

In order to approximate $P[J_{u,v} \geq \varepsilon]$, we wish to employ the Lyapunov CLT. Before we proceed, the random variables must be independent. Since $\mathcal{M}_{u,v}$, $\mathcal{N}_{u,v}$ contain some overlapping random variables in their definitions, they are not independent. So we substitute in the formulas for $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$ to decouple the ERV in the sum over the first $2q - 1$ terms.

$$
\begin{aligned}
&P[\mathcal{M}_{u,v} - \varepsilon\mathcal{N}_{u,v} \geq 0] = \\
&= P\Big[2 + \sum_{i=0}^{q-1} X_{y_{2i},u}X_{y_{2i+1},v} - 2\varepsilon - \varepsilon\sum_{i=0}^{q-1}\max\left(X_{y_{2i},u}, X_{y_{2i+1},v}\right) - \varepsilon \sum_{\substack{j=2q \\ z\in\{u,v\}}}^{r-1} X_{y_j,z} \geq 0\Big] \\
&= P\Big[2(1-\varepsilon) + \sum_{i=0}^{q-1}\Big\{X_{y_{2i},u}X_{y_{2i+1},v} - \varepsilon\max\left(X_{y_{2i},u}, X_{y_{2i+1},v}\right)\Big\} - \varepsilon \sum_{\substack{j=2q \\ z\in\{u,v\}}}^{r-1} X_{y_j,z} \geq 0\Big]
\end{aligned}
\tag{4.10}
$$

The term inside the first summand depends on two ERV $X_{y_{2i},u}$, and $X_{y_{2i+1},v}$, which combined have four possible outcomes. We derive a new random variable that encapsulates all possible states of the expression inside the first sum of equation 4.10.

Proposition 1. *For $i \in [0, q-1]$, let $Z(u, v, y_{2i})$ be a random variable such that $Z(u, v, y_{2i}) = X_{y_{2i},u}X_{y_{2i+1},v} - \varepsilon\max\left(X_{y_{2i},u}, X_{y_{2i+1},v}\right)$ then the possible states of $Z(u, v, y_{2i})$ are*

$$
Z(u, v, y_{2i}) = \begin{cases} 0, & (1 - p(y_{2i}, u))(1 - p(y_{2i}, v)) \\ -\varepsilon, & p(y_{2i}, v)(1 - p(y_{2i}, u)) + p(y_{2i}, u)(1 - p(y_{2i}, v)) \\ 1 - \varepsilon, & p(y_{2i}, u)p(y_{2i}, v) \end{cases} \tag{4.11}
$$

we call $Z(u, v, y_{2i})$ the Intersect Random Variable.

Notice, $Z(u, v, y_{2i})$ are independent random variables since each one is dependent on distinct pairs of edge random variables. We can substitute in the Intersect Random Variable into $P[J_{u,v} \geq \varepsilon]$.

$$
P[J_{u,v} \geq \varepsilon] = P\left[\sum_{i=0}^{q-1} Z(u, v, y_{2i}) + \sum_{\substack{i=2q \\ z\in\{u,v\}}}^{r-1} (-\varepsilon)X_{y_i,z} \geq 2(\varepsilon - 1)\right] \tag{4.12}
$$

Proposition 2. *Let $W(z, y_i)$ be a random variable such that $W(z, y_i) = (-\varepsilon)X_{y_i,z}$, then the possible states of $W(z, y_i)$ are*

$$W(z, y_i) = \begin{cases} -\varepsilon, & p(y_i, z) \\ 0, & 1 - p(y_i, z) \end{cases} \tag{4.13}$$

The sets of random variables $Z(u, v, y_{2i})$ and $W(z, y_i)$ are independent but non-identically distributed random variables, as required for the Lyapunov CLT. Let Z be the sum of $Z(u, v, y_{2i})$ Intersect Random Variables; and let W be the sum of $W(z, y_i)$ random variables.

$$Z = \sum_{i=0}^{q-1} Z(u, v, y_{2i}) \quad \text{and} \quad W = \sum_{\substack{i=2q \\ z\in\{u,v\}}}^{r-1} W(z, y_i) \tag{4.14}$$

With Z and W, the probability expression $P[J_{u,v} \geq \varepsilon]$ becomes,

$$P[Z + W \geq 2(\varepsilon - 1)] = P[V \geq 2(\varepsilon - 1)] \tag{4.15}$$

where $V = Z + W$. Therefore we now have a probability expression of independent but non-identically distributed random variables, which satisfies the first condition required for Lyapunov CLT.

Theorem 2. *For the Lyapunov CLT, let*
$Z(u, v, y_0), Z(u, v, y_2), \ldots, Z(u, v, y_{2q-2}), W(z, y_{2q}), \ldots, W(z, y_{r-1})$
be a sequence of independent but non-identically distributed random variables, each with finite expected value $\mu_{Z(u,v,y_{2i})}$, $\mu_{W(z,y_i)}$ and variance $\sigma^2_{Z(u,v,y_{2i})}$, $\sigma^2_{W(z,y_i)}$. Let

$$s_n^2 = \sum_{i=0}^{q-1} \sigma^2_{Z(u,v,y_{2i})} + \sum_{i=2q}^{r-1} \sigma^2_{W(z,y_i)} = \sum_{k=1}^{n} \sigma^2_{V_k} \tag{4.16}$$

and when $\delta = 1$, the limit

$$\lim_{n\to\infty} \frac{1}{s_n^3} \sum_{k=1}^{n} E[|V_k - \mu_{V_k}|^3] = 0 \tag{4.17}$$

where $n = r - q$ and V_k is either $Z(u, v, y_{2i})$ or $W(z, y_i)$. Then the random variable V converges to a standard normal random variable. Therefore $P[J_{u,v} \geq \varepsilon]$

approximates a one-tailed area under the Normal Distribution with mean

$$\mu_V = \sum_{i=0}^{q-1} \mu_{Z(u,v,y_{2i})} + \sum_{i=2q}^{r-1} \mu_{W(z,y_i)} = \sum_{k=1}^{n} \mu_{V_k} \tag{4.18}$$

and variance s_n^2.

To prove the theorem above, we demonstrate that the limit in equation 4.17 converges to zero.

Proof. The means and variances are calculated from the moment generating functions $M_{Z(u,v,y_{2i})}(t)$, and $M_{W(z,y_i)}(t)$. For simplicity let $p_{i1} = p(y_{2i}, u)$, $p_{i2} = p(y_{2i+1}, v)$, $\alpha = p_{i2}(1-p_{i1}) + p_{i1}(1-p_{i2})$, $\beta = (1-p_{i1})(1-p_{i2})$, and $\gamma = p_{i1}p_{i2}$.

$$M_{Z(u,v,y_{2i})}(t) = \alpha e^{-\varepsilon t} + \beta + \gamma e^{(1-\varepsilon)t}$$
$$M_{W(z,y_i)}(t) = p(z,y_i)e^{-\varepsilon t} + (1 - p(z,y_i))$$

By taking the first and second derivatives of $M_{Z(u,v,y_{2i})}(t)$, and $M_{W(z,y_i)}(t)$; using the well known theorem $M_X^{(j)}(0) = \mathrm{E}[X^j]$, the means and variances are,

$$\mu_{Z(u,v,y_{2i})} = \gamma(1-\varepsilon) - \alpha\varepsilon \tag{4.19}$$
$$\mu_{W(z,y_i)} = -\varepsilon p(z,y_i) \tag{4.20}$$
$$\sigma^2_{Z(u,v,y_{2i})} = \alpha\varepsilon^2(1-\alpha) + \gamma(1-\varepsilon)^2(1-\gamma) + 2\alpha\gamma\varepsilon(1-\varepsilon) \tag{4.21}$$
$$\sigma^2_{W(z,y_i)} = \varepsilon^2 p(z,y_i)(1-p(z,y_i)) \tag{4.22}$$

Let $Y_{2i} = |Z(u,v,y_{2i}) - \mu_{Z(u,v,y_{2i})}|$ and $Q_i = |W(z,y_i) - \mu_{W(z,y_i)}|$. Since V_k either represents $Z(u,v,y_{2i})$ or $W(z,y_i)$ then sufficient conditions for equation 4.17 to hold are

$$\mathrm{E}[Y_{2i}^3] \leq \sigma^2_{Z(u,v,y_{2i})} \tag{4.23}$$
$$\mathrm{E}[Q_i^3] \leq \sigma^2_{W(z,y_i)} \tag{4.24}$$

because these conditions will ensure the limit approach zero. To show the inequalities above, we require the expected value functions for Y_{2i}^3 and Q_i^3. We derive the functions

$\mathrm{E}[Y_{2i}^3]$ and $\mathrm{E}[Q_i^3]$ from first principles.

$$\mathrm{E}[Y_i^3] = |-\gamma(1-\varepsilon)+\alpha\varepsilon|^3\beta + |-\varepsilon-\gamma(1-\varepsilon)+\alpha\varepsilon|^3\alpha + |1-\varepsilon-\gamma(1-\varepsilon)+\alpha\varepsilon|^3\gamma \tag{4.25}$$

$$\mathrm{E}[Q_i^3] = |-\varepsilon p(z,y_i)|^3(1-p(z,y_i)) + |-\varepsilon+\varepsilon p(z,y_i)|^3 p(z,y_i) \tag{4.26}$$

Next, the inequality condition from equation 4.24 is derived:

$$\begin{aligned}
\mathrm{E}[Q_i^3] &= |-\varepsilon p(z,y_i)|^3(1-p(z,y_i)) + |-\varepsilon+\varepsilon p(z,y_i)|^3 p(z,y_i) \\
&= (\varepsilon p(z,y_i))^3(1-p(z,y_i)) + \varepsilon^3 p(z,y_i)|p(z,y_i)-1|^3 \\
&= (\varepsilon p(z,y_i))^3(1-p(z,y_i)) + \varepsilon^3 p(z,y_i)(1-p(z,y_i))^3 \\
&= \varepsilon^3 p(z,y_i)(1-p(z,y_i))(p(z,y_i)^2 + (1-p(z,y_i))^2) \\
&\leq \sigma^2_{W(z,y_i)}
\end{aligned} \tag{4.27}$$

For the inequality condition in equation 4.23, consider the alternate (and equivalent) form:

$$f(p_{i1}, p_{i2}, \varepsilon) = \sigma^2_{Z(u,v,y_{2i})} - \mathrm{E}[Y_{2i}^3] \geq 0 \tag{4.28}$$

The condition as stated in equation 4.28, may now be viewed as a minimization problem. We use a numerical solver that implements Sequential Least Squared Programming (SLSQP) to minimize a function of several variables with any combination of bounds [18]. For our problem, we have a scalar function $f(p_{i1}, p_{i2}, \varepsilon)$ with three variables that is bounded by the regions $p_{i1} \in (0,1], p_{i2} \in (0,1], \varepsilon \in (0,1]$. [1] With SLSQP, we are able to find the local minimums of the function in equation 4.28. By this process we observe that $f(p_{i1}, p_{i2}, \varepsilon)$ remains non-negative on the regions $p_{i1} \in (0,1], p_{i2} \in (0,1], \varepsilon \in (0,1]$ (see Appendix B for more details). Thus, $f(p_{i1}, p_{i2}, \varepsilon) \geq 0$ on the unit cube in $(p_{i1}, p_{i2}, \varepsilon)$ space.

Since both conditions from equations 4.23 and 4.24 are satisfied, we insert these inequalities into the limit equation 4.17. Then by L'Hopitals Rule, the limit condition

[1] In our numerical evaluation, we refrain from going lower than 10^{-30} for the three parameters because of numerical instabilities that can occur at extremely small values.

in equation 4.17 is satisfied, when $\delta = 1$.

$$\begin{aligned}
&\lim_{n\to\infty} \frac{1}{s_n^3} \sum_{k=1}^{n} \mathrm{E}[|V_k - \mu_{V_k}|^3] = \\
&\leq \lim_{n\to\infty} \frac{1}{s_n^3} \left[\sum_{i=0}^{q-1} \mathrm{E}[Y_{2i}^3] + \sum_{i=2q}^{r-1} \mathrm{E}[Q_i^3] \right] \\
&\leq \lim_{n\to\infty} \frac{1}{s_n^3} \left[\sum_{i=0}^{q-1} \sigma^2_{Z(u,v,y_{2i})} + \sum_{i=2q}^{r-1} \sigma^2_{W(z,y_i)} \right] \\
&= \lim_{n\to\infty} \frac{s_n^2}{s_n^3} = \ 0
\end{aligned} \tag{4.29}$$

Hence, by Lyapunov CLT, the set of variables in the sequence χ_{uv} (equation 4.5) are normally distributed with mean μ_V and variance s_n^2. □

The probability expression in equation 4.15 can be manipulated such that the Normal Distribution applies.

$$\begin{aligned}
&P[V \geq 2(\varepsilon - 1)] = Pr[V - \mu_V \geq 2(\varepsilon - 1) - \mu_V] \\
&= P\left[\sum_{k=1}^{n} V_k - \mu_{V_k} \geq 2(\varepsilon - 1) - \sum_{k=1}^{n} \mu_{V_k} \right] \\
&= P\left[\frac{1}{s_n} \sum_{k=1}^{n} V_i - \mu_{V_k} \geq \frac{1}{s_n} \left(2(\varepsilon - 1) - \sum_{k=1}^{n} \mu_{V_k} \right) \right]
\end{aligned} \tag{4.30}$$

Therefore, the probability of structural similarity is approximately the above Normal Distribution times the edge probability. That is,

$$P[(u, v), \varepsilon] \approx P[V \geq 2(\varepsilon - 1)] \times p(u, v) \tag{4.31}$$

In the following section, we take the theory developed from this section and design an algorithm which implements the calculation of the probability of structural similarity as defined in equation 4.31.

4.2 NUSCAN

We call our algorithm NUSCAN where "N" is to emphasize the use of the Normal Distribution as per Lyapunov CLT. More specifically, if $|\widetilde{N_{uv}}| \geq t$, then we use Normal Distribution to compute $P[(u,v),\varepsilon]$, for some large $t \in \mathbb{N}$. In practice we set $t = 100$ for all graphs (see the end of Section 5.3 for details). Based on equations 4.30 and 4.31, we propose the following algorithm for calculating $P[(u,v),\varepsilon]$ using Lyapunov CLT.

Algorithm 2 Calculation of $P[(u,v),\varepsilon]$

1: **procedure** ComputePr(u,v,ε)
2: **if** $p(u,v) < \eta$ **then return** 0
3: **else if** $|\widetilde{N_{uv}}| < t$ **then**
4: Use USCAN DP protocol
5: **else**
6: Arrange all $w \in \widetilde{N_{uv}}$ as sequence Y_{uv} (equation 4.4)
7: Split Y_{uv} into W and Z (equation 4.14)
8: $\mu_V \leftarrow 0,\ s_n^2 \leftarrow 0$ (equations 4.18 and 4.16)
9: **for** y_i in W **do**
10: $\mu_V \leftarrow \mu_V + \mu_{W(z,y_i)}$
11: $s_n^2 \leftarrow s_n^2 + \sigma^2_{W(z,y_i)}$
12: **for** y_{2i} in Z **do**
13: $\mu_V \leftarrow \mu_V + \mu_{Z(u,v,y_{2i})}$
14: $s_n^2 \leftarrow s_n^2 + \sigma^2_{Z(u,v,y_{2i})}$
15: Let $F_n \leftarrow Norm(\mu_V, s_n)$
16: **return** $P\left[F_n \geq \frac{2(\varepsilon-1)-\mu_V}{s_n}\right] \times p(u,v)$ (equation 4.31)

The first step on line 2 is a pruning condition inherited from USCAN. Line 3 checks whether there are enough common neighbours for the application of the NUSCAN approximation, which is done in constant time. The next part on lines 6-7 prepares the neighbours into two sets Z and W which contain the random variables $Z(u,v,y_{2i})$ and $W(z,y_i)$ respectively, taking only $O(|\overline{N_u} \cup \overline{N_v}|)$. Lines 9-14 calculate and sum the means and variances of each random variable in the sequence as described in section 4.1, which finishes both loops in $O(|\overline{N_u} \cup \overline{N_v}|)$. In line 15, a Normal Distribution is constructed with mean μ_V and standard deviation s_n, done in constant time. Finally

on line 16, the Normal Distribution is used to return the probability that approximates $Pr[(u,v),\varepsilon]$, also in constant time. Then for an edge (u,v), all neighbours in the union are iterated over. Therefore in the worst case, the run time of Algorithm 2 is $O(|\overline{N_u} \cup \overline{N_v}|)$.

Run Time Complexity In NUSCAN, if an edge (u,v) has $|\widetilde{N_{uv}}| \geq t$ then the approximation method occurs, otherwise the DP algorithm takes place. Since the DP algorithm has a worst case time of $O(|\overline{N_u} \cup \overline{N_v}|^2)$ [24], all edges that run through the DP protocol take $O(t^2)$, however because t is a constant the complexity becomes $O(1)$. Therefore, in the worst case Algorithm 2 takes $O(|\overline{N_u} \cup \overline{N_v}|)$ from the NUSCAN formulation. Since $|\overline{N_u} \cup \overline{N_v}|$ is bounded above by $2 \times d_{max}$, then $\forall\ e \in \mathcal{E},\ \exists\ e\ s.t.$ NUSCAN ComputePr(u,v,ε) takes $O(d_{max})$. Therefore, the total run time for NUSCAN to complete on the entire graph $\mathcal{G}$ is $O(d_{max} \times \alpha \times m)$, where α is the arboricity of the graph and $m = |\mathcal{E}|$ [7].

4.3 Approximation Bound

In this section we bound the error of Normal Distribution approximation in equation 4.30 by using the Berry-Essen Theorem [12, 32].

Theorem 3 (Berry-Essen Theorem)**.** *Given a sequence $\Gamma_1, \ldots, \Gamma_n$ of non-identically distributed and independent random variables with $E[\Gamma_i] = 0$ and $E[\Gamma_i^2] = \lambda_i^2$ and $E[|\Gamma_i^3|] = \rho_i < \infty$, $\exists\ C_0 = 0.56$ s.t. the following is satisfied:*

$$\sup_{x \in \mathbb{R}} |F_n(x) - \Phi(x)| \leq C_0 \psi_0 \tag{4.32}$$

where $F_n(x)$ is the Cumulative Distribution Function (CDF) for

$$S_n = \frac{\Gamma_1 + \cdots + \Gamma_n}{\sqrt{\lambda_1^2 + \cdots + \lambda_n^2}} \tag{4.33}$$

and $\Phi(x)$ is the CDF for the Normal Distribution; and

$$\psi_0 = \psi_0(\vec{\lambda}, \vec{\rho}) = \left(\sum_{i=1}^{n} \lambda_i^2 \right)^{-\frac{3}{2}} \sum_{i=1}^{n} \rho_i \tag{4.34}$$

Note, the value of C_0 is determined to be 0.56 from previous works [12]. We give the following corollary that depicts how to obtain an upper bound on the maximal error of the approximation of V to a Normal Distribution.

Corollary 2. *For each edge $(u, v) \in \mathcal{E}$ in $\mathcal{G}$ with random variables $V_1 - \mu_{V_1}, \ldots, V_n - \mu_{V_n}$, the error on the approximation of the right hand side of equation 4.30 to the Normal Distribution is given by:*

$$\sup_{x \in \mathbb{R}} |F_r(x) - \Phi(x)| \leq \frac{0.56}{\sqrt{\sum_{i=0}^{q-1} \sigma^2_{Z(u,v,y_{2i})} + \sum_{i=2q}^{r-1} \sigma^2_{W(z,y_i)}}} \tag{4.35}$$

Proof. Let

$$\sum_{k=1}^{n} \Gamma_k = \sum_{k=1}^{n} V_k - \mu_{V_k}$$

From the derivation in Section 3.1 we also know that $E[|Y_k^3|] < \infty$ and $E[|Q_k^3|] < \infty$. So

$$S_n = \frac{\sum_{k=1}^{n} V_k - \mu_{V_k}}{\sqrt{\sum_{i=0}^{q-1} \sigma^2_{Z(u,v,y_{2i})} + \sum_{i=2q}^{r-1} \sigma^2_{W(z,y_i)}}} \tag{4.36}$$

and

$$\begin{aligned}
\psi_0(\vec{\sigma}, \vec{\rho}) &= \left(\sum_{k=1}^{n} \sigma^2_{V_k} \right)^{-\frac{3}{2}} \sum_{i=1}^{n} \rho_i \\
&\leq \left(\sum_{i=0}^{q-1} \sigma^2_{Z(u,v,y_{2i})} + \sum_{i=2q}^{r-1} \sigma^2_{W(z,y_i)} \right)^{-\frac{3}{2}} \\
&\quad \cdot \left(\sum_{i=0}^{q-1} \sigma^2_{Z(u,v,y_{2i})} + \sum_{i=2q}^{r-1} \sigma^2_{W(z,y_i)} \right) \\
&= \left(\sum_{i=0}^{q-1} \sigma^2_{Z(u,v,y_{2i})} + \sum_{i=2q}^{r-1} \sigma^2_{W(z,y_i)} \right)^{-\frac{1}{2}}
\end{aligned} \tag{4.37}$$

Therefore, the bound on the approximation of the Normal Distribution is given by the inequality 4.35 □

Chapter 5

Experiments

In this chapter we demonstrate the efficiency, scalability, accuracy, and effectiveness of our proposed algorithm NUSCAN, compared to the state-of-art algorithm USCAN [24]. All algorithms are implemented in `C++` and compiled with `g++` using the `-O3` optimization flag. The experiments are executed on a commodity machine with Intel Xeon E5620, 2.395GHz CPU, and 64Gb RAM, running Ubuntu 18.04. The implementation is available at anonymous.4open.science/r/nuscan-C682.

5.1 Datasets and Experimental Framework

datasets	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	d_{max}	d_{ave}
enron	69,017	254,449	1,634	7.37
cnr-2000	325,557	2,738,969	18,236	16.83
uk-2014-tpd	1,766,010	15,283,718	63,731	17.31
eu-2005	862,664	16,138,468	68,963	37.42
dewiki-2013	1,510,148	33,093,029	118,246	43.83
eswiki-2013	970,327	21,184,931	145,310	43.67
uk-2002	18,483,185	261,787,258	194,955	28.33
indochina-2004	7,414,757	150,984,819	256,425	40.73
arabic-2005	22,743,880	553,903,073	575,628	48.71

Table 5.1: Datasets are retrieved from Laboratory of Web Algorithmics (https://law.di.unimi.it/datasets.php).

Preprocessing The datasets we use in our experimentation have their statistics given in Table 5.1, and are ordered by the maximum degree. For each of the datasets, the edge probabilities are added randomly from a power law distribution with a mean of two, in the range $[0, 1]$. For the algorithms to run properly, all self-loops and isolated nodes are removed from the original datasets. Additionally, directed graphs are converted to undirected graphs by adding symmetrical edges whenever they are missing. The statistics in Table 5.1 reflect the datasets after these modifications. Each dataset runs on 55 different parameter points in the phase space (η, ε, μ) in order to analyse how the variation in parameter values effect the efficiency, scalability, accuracy, and effectiveness of the algorithms.

5.2 Comparison to USCAN

datasets	sample points	RMSE (%)	time improv. (t_{imp})
enron	2000	1.81%	16x
cnr-2000	1740	0.27%	21x
uk-2014-tpd	1446	0.34%	67x
eu-2005	1252	0.34%	26x
dewiki-2013	2164	0.38%	137x
eswiki-2013	2158	0.39%	29x

Table 5.2: RMSE and runtime improvement. RMSE is quite small (less than 2%) and runtime improvement is up to 137 times. Sample edges are chosen from those that pass η pruning and threshold $t = 100$. We have set $(\eta, \varepsilon, \mu) = (0.3, 0.5, 2)$.

We attempted to run a comprehensive suite of 55 parameter combinations of η, ε, μ on USCAN and NUSCAN to adequately analyze the time improvement from our approximation method. However, only a handful of points could finish for USCAN on the six smallest datasets within 48 hours of allowed running time. In order to still provide some sensible comparision between USCAN and NUSCAN, we focus on the portion that calculates $P[e, \varepsilon]$ for a random sample of several thousand edges from each dataset. More specifically, edges that are sampled must have $p(u, v) \geq \eta$ and $|\widetilde{N_{uv}}| \geq t$ in order to compare the two different methods for calculating $P[e, \varepsilon]$. Recall that calculating $P[e, \varepsilon]$ is the bottleneck in structural clustering of probabilistic graphs. After obtaining the sample of edges for each dataset, we run USCAN's DP

and our NUSCAN's Normal approximation to compute $P[e, \varepsilon]$ for each edge, and measure the average elapsed time as well as the error of approximation as compared to the DP result.

Root mean squared error To measure the error in the value of $P[e, \varepsilon]$ between both methods, we use the root mean squared error ($RMSE$). That is, let S be the set of sampled edges e, let $P_n(e) = P[e, \varepsilon]$ calculated with Algorithm 2 and let $P_d(e) = P[e, \varepsilon]$ calculated with the DP algorithm. Then,

$$RMSE = \sqrt{\frac{1}{|S|} \sum_{e \in S} \left(P_n(e) - P_d(e)\right)^2} \tag{5.1}$$

The $RMSE$ provides an empirical quantification of the deviation between these two methods for calculating $P[e, \varepsilon]$.

Time improvement To measure the scalability of the proposed algorithm NUSCAN, we determine the time to execute the bottleneck calculation of $P[e, \varepsilon]$ and compare the time with USCAN. That is, let S be the set of sampled edges e, let $T_n(e)$ be the time for edge e to complete under Algorithm 2, and let $T_d(e)$ be the time for e to complete under the DP algorithm. Then the average time improvement (t_{imp}) from USCAN to NUSCAN is defined as

$$t_{imp} = \frac{1}{|S|} \sum_{e \in S} \left(\frac{T_d(e)}{T_n(e)}\right) \tag{5.2}$$

where the summand is the time improvement fraction between Algorithm 2 and the DP algorithm on edge e.

The data in Table 5.2 are derived from a sampling of edges in each datasets that passed both the η pruning and the t threshold, for parameter values $(\eta, \varepsilon, \mu) = (0.3, 0.5, 2)$. Hence the edges compared ran through the DP calculation on USCAN and Algorithm 2 on NUSCAN. The comparison reveals that the RMSE percentage is remarkably small, with the error on $P[e, \varepsilon]$ from Algorithm 2 being between 0.3% to 1.8% depending on the dataset. Additionally, the same points had the time for each method recorded, and compared using equation 5.2, which yields the average time improvement over the sampled points. Algorithm 2 improves the running time of the bottleneck process by a factor of 16x to 137x depending on the dataset. NUSCAN produces a significant improvement in running time while maintaining a highly

accurate approximation for the calculation of $P[e, \varepsilon]$.

5.3 Efficiency Evaluation

In this section, we study the running time of our proposed algorithm NUSCAN over the space of parameters η, ε, μ. We set threshold $t = 100$ for each dataset. This means that we trigger the structural similarity computation using Normal Distribution only if the size of the union of neighbours for a pair of vertices is at least 100, otherwise, DP is used. Parameters η, ε, μ are varied over 55 different points in the phase space to generate a holistic sampling, and draw insights into how the parameters effect the running time of the datasets.

Comparing the variation in η across the different datasets in Figure 5.1, the running time generally increases with the maximum degree of the dataset. As η increases, each dataset running time curve drops drastically. Figure 5.1 reveals that some datasets plateau off earlier than others because of the random edge probability assignment coupled with the differences in structure. For instance, with the dataset *cnr-2000*, the time drops from ten minutes when $\eta = 0.2$, all the way to one second when $\eta = 0.8$. A larger dataset such as *eswiki-2013* starts off with a time at close to 30 minutes when $\eta = 0.2$, and it goes down to 25 seconds when $\eta = 0.8$. In general, all the datasets level off as η increases. Since the edge probabilities are drawn from a power law distribution, η pruning happens frequently as the value of η increases. Overall, NUSCAN completes on the largest dataset, *arabic-2005*, in less than an hour for the majority of threshold parameter points (η, ε, μ) tested. In contrast, USCAN was not able to complete in a reasonable time on any of the large datasets we tested with more than 20 million edges.

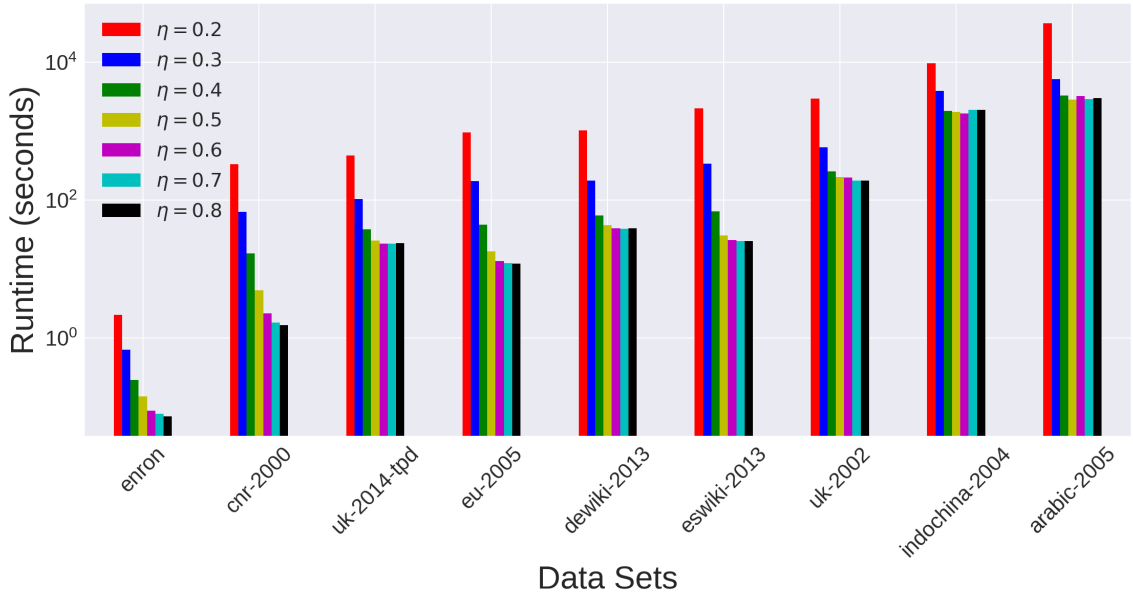

Figure 5.1: Running time for varying η across datasets. We set $(\eta, \varepsilon, \mu) = (\eta, 0.5, 2)$, and $t = 100$. As η increases, fewer probabilities are calculated (due to η pruning) and this reduces running time. Since the edge probabilities follow a power-law distribution, this effect plateaus as η approaches the top of its range.

Phase space Each of the input parameters η, ε, μ have their own range of allowed values. Both η and ε are in the range $(0, 1]$, whereas $\mu \geq 2$. We can break the 55 explored points into three groups based on the fixed values $(0.2, 0.5, 2)$, $(0.5, 0.2, 2)$ and $(0.5, 0.5, 5)$. For each of the 55 runs, two of the three parameters (η, ε, μ) are held constant and the third is varied over a range of points. The range of values chosen for η and ε are $[0.1, 0.2, 0.3, 0.4, 0.5,\ 0.6, 0.7, 0.8]$, and μ has range $[2, 3, 4, 5, 6, 7]$. For all of the nine datasets, our algorithm completes on all points well within 48 hours; with one exception of *arabic-2005* for the three points $(0.1, \varepsilon, \mu)$ where the algorithm did not finish within the time limit.

Vanishing cluster set Over the explored 55 points of the phase space, the number of clusters diminished as the parameters (η, ε, μ) reach the top of their ranges. The vanishing of clusters is consistent with USCAN, and thus not unique to NUSCAN. Specifically, when μ surpasses five, the number of clusters, independent of dataset, becomes zero. Since μ is the parameter responsible for determining if nodes form a cluster, the absence of clusters is bound to occur at some finite value of μ regardless of η and ε. However, clusters also vanish when η and ε become large because these parameters influence reliable structural similarity and the probability of structural similarity respectively. Consequently, when all these parameters are high, the odds of enough edges passing all the threshold requirements approaches zero. Therefore the lower half of these parameter ranges are more desirable for generating larger cluster sets.

After analyzing the running time results from each of the 55 points in the phase space, we found that only the parameter η sensibly effects the time. Reflecting on the clustering algorithm, because μ is a threshold on the size of the reliable neighbourhood set to determine which vertices are reliable core vertices, then each vertex will be checked regardless of the value of μ. Similarly, ε does not effect the run time, since Algorithm 2 runs in the same time regardless of ε. However, η will effect the run time because of a pruning condition that was developed in the USCAN algorithm. Since two nodes are reliable structural similar only if $P[e, \varepsilon] \geq \eta$, then if the probability $p(e) < \eta$ that implies $P[e, \varepsilon] < \eta$, by definition of $P[e, \varepsilon]$. Hence for NUSCAN, out of the three parameters (η, ε, μ), only η significantly effects the running time.

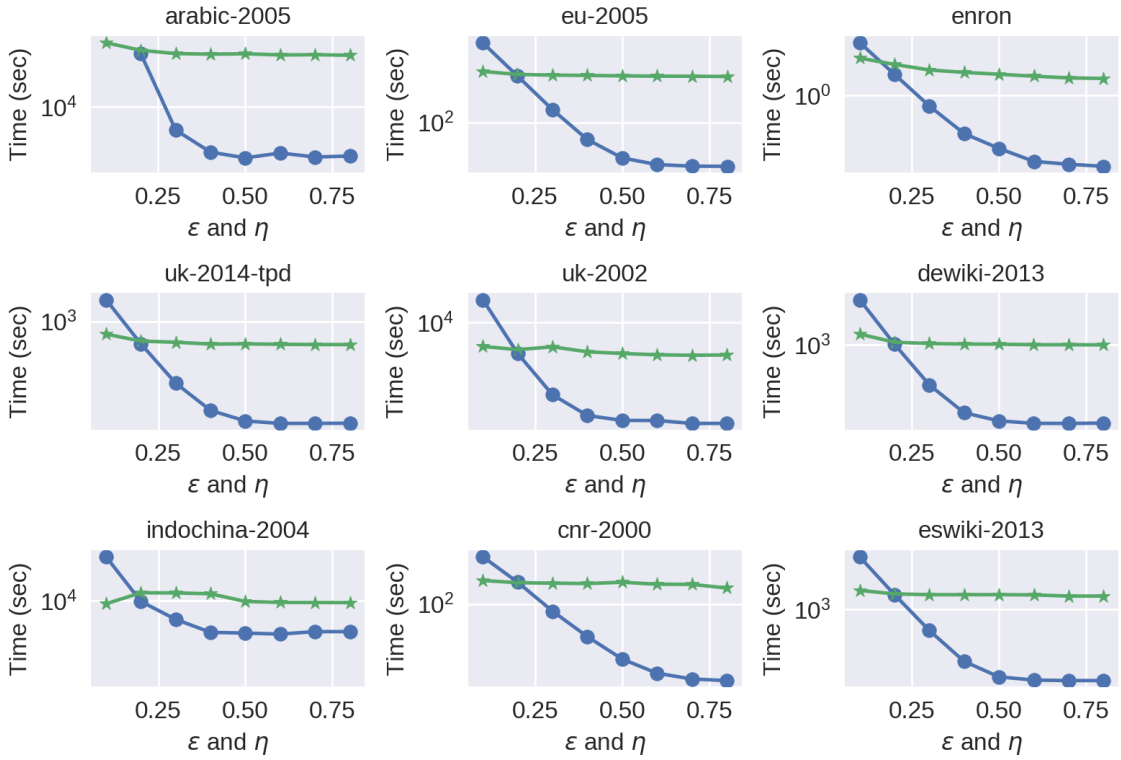

Figure 5.2: Running time for varying η and ε with $\mu = 2$ and $t = 100$. Since η and ε share the same range of values, we present both running time curves on one plot. Blue shows the variation of η, with $\varepsilon = 0.5$; green shows the variation of ε, with $\eta = 0.2$. We see that running time is mainly influenced by η.

Figure 5.2 displays the effect η has on the run time. The η varying curve drops super exponentially in time over the range of chosen points, while the ε curve is a flat line. Moreover, the ε line intersects the η curve at 0.2, which is the value η is set to in Figure 5.2. In each of the nine plots in Figure 5.2, the ε line intersects the η curve right at the $\eta = 0.2$ position. Since the value of η is the parameter that dictates the running time of the process, then it is expected that the ε curve is a straight line constant in time that intersects at $\eta = 0.2$ on the η curve.

With the threshold t determining whether Algorithm 2 is used in NUSCAN, clearly t has an impact on the running time. In the theoretical framework, the Normal Distribution becomes a tighter approximation as the number of random variables approaches infinity. However, it is well known that CLT can be safely applied when the number of random variables is as low as 50. We tested the NUSCAN algorithm on a range of increasing t values $[60, 70, 80, 90, 100, 500, 1000]$. The results showed that from 60 to 100 there was no noticeable difference in running time or clustering quality. Where as, the values $500, 1000$ had significantly longer running times, but no great benefit in quality. From these results, the value $t = 100$ was chosen as a sensible standard for experimentation.

5.4 Effectiveness Testing

The only ground truth datasets known are small and already used by the authors of USCAN [24]. As we already mentioned, our clustering results are indistinguishable from those of USCAN, so there is no point repeating the same analysis as [24] for the ground truth datasets. Additionally, since our results are near identical to USCAN, we do not reproduce the comparison to other clustering algorithms done in the USCAN paper. However, what we would like to do here is to show the effectiveness of structural clustering in terms of quality for large datasets on which USCAN cannot scale, but our algorithm NUSCAN can. We start with testing a clustering metric called *Average Expected Density* (AED) defined as:

$$AED = \frac{1}{|\mathbb{C}|} \sum_{C_i \in \mathbb{C}} \sum_{e \in C_i} \frac{2p_e}{|V_i| \times (|V_i| - 1)} \tag{5.3}$$

which measures the strength of connection in each cluster averaged over all clusters, where V_i is the set of vertices in C_i.

In Biswas et. al. [1], they outline three metrics to measure quality of clusters when no ground truth presents itself for comparison. The authors define three metrics Average Isolability (Q_{AVI}), Average Unifiability (Q_{AVU}), and Average Isolability and Unifiability (Q_{ANUI}). Isolability determines the strength of connection within each cluster—similarly to AED—where as Unifiability measures the strength of connection between two distinct clusters. Then Q_{ANUI} is a ratio of the average Isolability and Unifiability. For a single cluster, Isolability (I) is defined as:

$$I(C_i) = \frac{\sum\limits_{u \in C_i v} p(u,v)}{\sum\limits_{u \in C_i v} p(u,v) + \sum\limits_{u \in C_i, v \notin C_i} p(u,v)} \tag{5.4}$$

and for a pair of clusters, Unifiability (U) is defined as:

$$U(C_i, C_j) = \frac{\sum\limits_{u \in C_i, v \in C_j} p(u,v)}{\sum\limits_{u \in C_i, v \notin C_i} p(u,v) + \sum\limits_{u \notin C_j, v \in C_J} p(u,v) - \sum\limits_{u \in C_i, v \in C_j} p(u,v)} \tag{5.5}$$

where C_i, and C_j are different clusters in $\mathbb{C}$. Then the averages of these measures Q_{AVI} and Q_{AVU} are defined as the arithmetic mean over all cluster sets. Then Q_{ANUI} is given from the two above equations as:

$$Q_{ANUI} = \frac{Q_{AVI}}{1 + Q_{AVI} \times Q_{AVU}} \tag{5.6}$$

Since Q_{ANUI} is a function of both Q_{AVI} and Q_{AVU}, we display Q_{ANUI} as the objective measure.

The goal is to demonstrate that NUSCAN returns a cluster set that is as good as the USCAN cluster set under these two metrics AED and Q_{ANUI}. For the six smallest datasets we are able to measure the quality of the cluster sets that NUSCAN produces. In *dewiki-2013* and *eswiki-2013*, for the ε variation curve, after $\varepsilon = 0.6$ the number of clusters found becomes zero. Hence for these two datasets, the two metrics are indeterminate for the points $(0.5, 0.7, 2)$ and $(0.5, 0.8, 2)$.

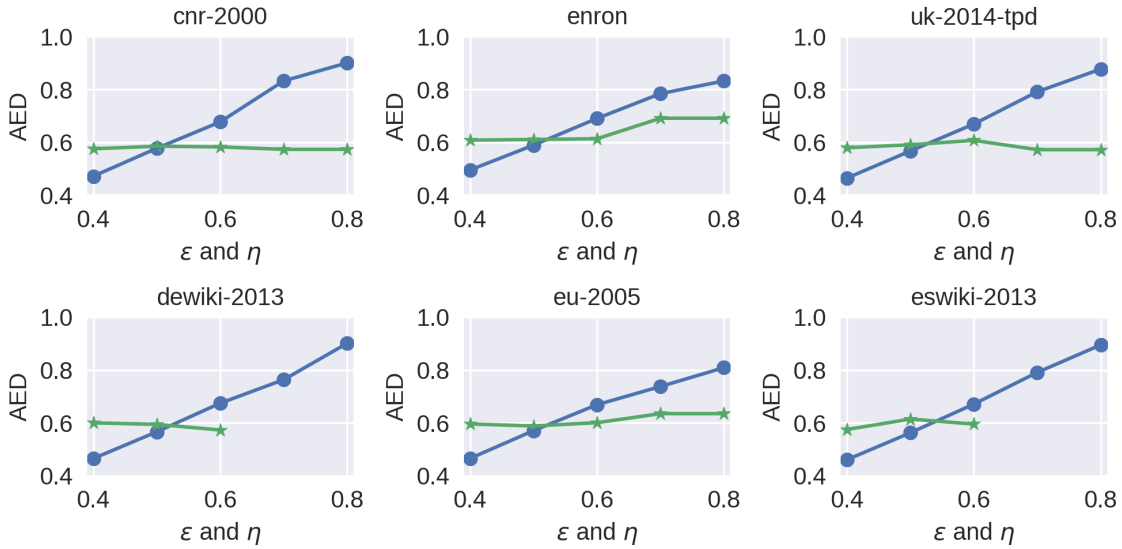

Figure 5.3: AED for NUSCAN when varying η and ε ($\mu = 2, t = 100$). Since η and ε share the same range of values, we show both AED curves on one plot. Blue shows the variation of η, with $\varepsilon = 0.2$ and $\mu = 2$; green shows the variation of ε, with $\eta = 0.5$ and $\mu = 2$. Again, varying ε does not influence AED much, whereas varying η has a more pronounced effect. As η increases over its ranges, AED linearly increases towards 1. Absence of some points in the ε line is due to lack of clusters at the high end of the parameter range.

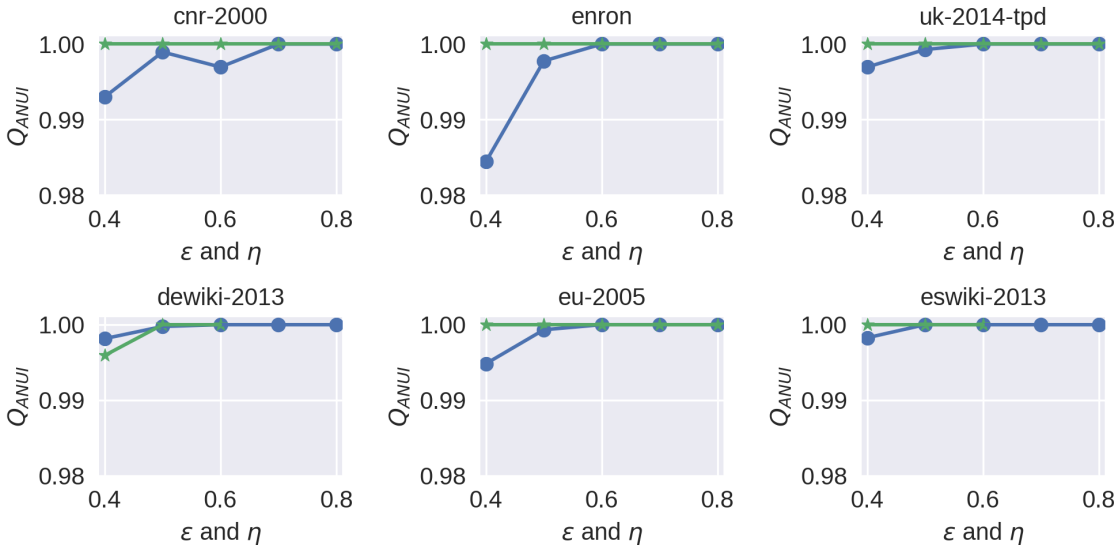

Figure 5.4: Q_{ANUI} for NUSCAN when varying η and ε ($\mu = 2, t = 100$) similar to Fig. 5.3.

Once again we see that η is the parameter responsible for the shape of these curves. For both metrics, ε forms a flat line that intersects the η curve at 0.5, which is the value η is held constant at for the ε line. The AED plots in Figure 5.3 demonstrate that as η increases from 0.4 to 0.8, the AED value increases from about 0.4 to 0.8-0.9 depending on the specific dataset. For instance, *enron* and *eu-2005* only make it to just over 0.8 at $\eta = 0.8$; meanwhile the four other datasets exceed 0.9 at $\eta = 0.8$. All of the η curves possess a positive linear slope. Since η is the threshold parameter that is responsible for whether two nodes are reliable structural similar, as η becomes large, exponentially less edges pass the threshold cut. Moreover, the edges that are reliably structurally similar at high η have very large edge probabilities, which leads to a high value for AED. As for the Q_{ANUI} metric plots in Figure 5.4, all the datasets quickly approach 1 in the η varied curve. Next we compare the three smallest datasets that were able to complete calculation of these metrics under the USCAN algorithm.

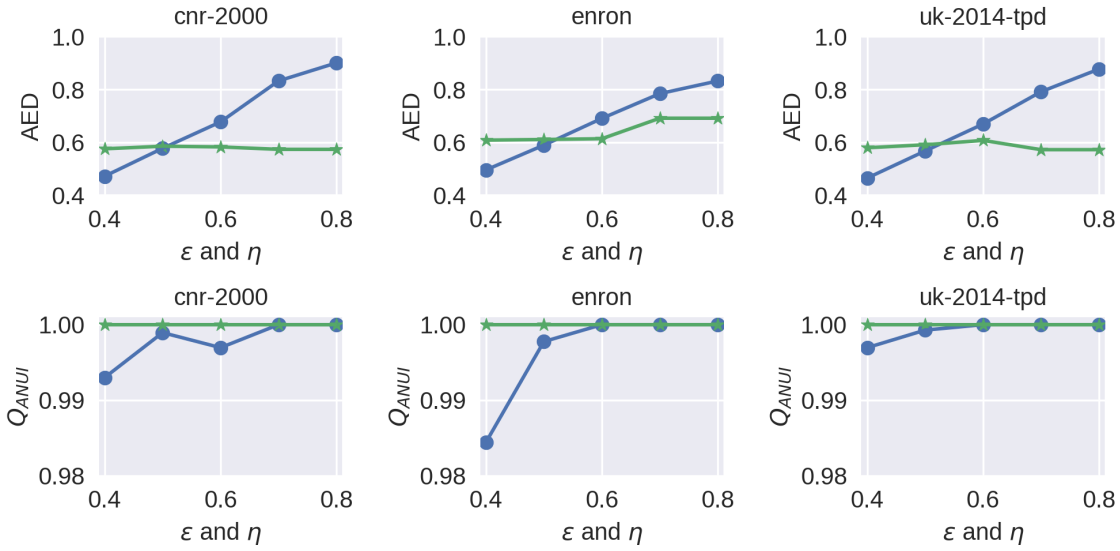

Figure 5.5: AED (first row) and Q_{ANUI} (second row) for USCAN when varying η and ε ($\mu = 2, t = 100$) similarly to Figs. 5.3 and 5.4. Observe that AED charts are indistinguishable from the AED charts in Fig. 5.3 (first row of Fig. 5.3). Likewise, observe that Q_{ANUI} charts are also indistinguishable from the Q_{ANUI} charts in Fig. 5.4 (first row of Fig. 5.4). So, our approximation method, NUSCAN, produces results virtually same as USCAN in practice.

Figure 5.5 shows both the *AED* and Q_{ANUI} plots for the three datasets *enron, cnr-2000, uk-2014-tpd*, as these were the only datasets that finished inside our 48 hour time constraint on USCAN. The plots of these three datasets for *AED*, and Q_{ANUI} show precisely the same curves as NUSCAN. The perfect alignment of these metric plots indicate that NUSCAN produces a near identical clustering to USCAN. Moreover, we have shown that NUSCAN does as good as USCAN on these quality metrics, meaning that with the approximation algorithm we do not sacrifice quality for the observed time improvement. Therefore, NUSCAN yields quality cluster sets that are comparable to its predecessor USCAN.

Chapter 6

Conclusions

Probabilistic graph are an important data structure that are finding a broader use in research and industry. These data structures are capable of modelling a vast amount of information, from social networks all the way to protein interactions. With the abundance of data being produced, these probabilistic graphs can become massive in size, where some networks contain more than a billion connections.

Clustering probabilistic networks is a complex and time consuming endeavor with low to moderate scalability. State-of-art algorithm tackles the problem of probabilistic graph clustering by deploying a DP algorithm which has a run time complexity of $O(d_{max}^2 \times \alpha \times m)$. Unfortunately the state-of-art algorithm cannot meet the demands of scaling to large data sets. In this thesis, we set forth a sophisticated approach to overcome the complexity hurdle that prevents the state-of-art method from scaling up to larger probabilistic graphs.

The proposed algorithm NUSCAN offers an approximation solution to the probabilistic clustering problem. The bottleneck of the state-of-art algorithm comes down to computing the structural similarity between each edge in the network over all possible worlds. The method outlined in this thesis aims to reduce the complexity of the bottleneck process such that the entire clustering framework takes only $O(d_{max} \times \alpha \times m)$ time in the worst case. We achieve the reduction of complexity by constructing a clever sequence of random variables that represent the edges making up the neighbourhood sets of a given edge. The specially constructed set of random variables is then shown to conform to a variant of the Central Limit Theorem, called Lyapunov CLT, which in turn is used to approximate the bottleneck calculation of structural similarity over all possible worlds.

The proposed approximation for the calculation of the structural similarity over all possible worlds enables the clustering of probabilistic graphs to large scale datasets on the order of a billion edges. Our algorithm NUSCAN was able to obtain highly accurate results on datasets with up to a half a billion edges in less than an hour, where the state-of-art algorithm USCAN was not able to complete on graphs with more than 20 million edges. When compared to the DP based algorithm USCAN, our approximation algorithm produces near identical clusters with up to two orders of magnitude time improvement on some real world datasets.

Future works can explore practical time improvements to NUSCAN by utilizing techniques such as parallel processing. Additionally, our approach works for undirected probabilistic graphs, and their are opportunities to expand this framework for scaling up the clustering of probabilistic directed graphs. Moreover, the NUSCAN approach has the potential to efficiently cluster weighted probabilistic graphs with tractable modifications to the NUSCAN method.

Appendix A

ProbSCAN

In the literature, work [19] presents ProbSCAN, a method claimed to offer a significant time improvement for an exact calculation of the probability of structural similarity. However, they do not give a proof of correctness for the method outlined in their paper. We show here that, unfortunately, ProbSCAN produces incorrect results.[1]

A.1 Counterexample

In order to show that the ProbSCAN algorithm is incorrect, we use the following counter example to demonstrate that even with a simple graph, the two methods fail to come to the same answer. First we will use the USCAN method to derive the probability of structural similarity on a single edge of the graph. Then we will repeat the calculation with the formula provided by the ProbSCAN paper. What we will find is that the two probabilities are not equal. Therefore ProbSCAN does not improve the computation time of USCAN since the anwser produced is incorrect. Consider the probabilistic graph $\mathcal{G}$ in Figure A.1. We wish to determine the probability of structural similarity for the edge (0,1) with $\varepsilon = 0.2$.

[1] We informed the authors of the error through personal communication.

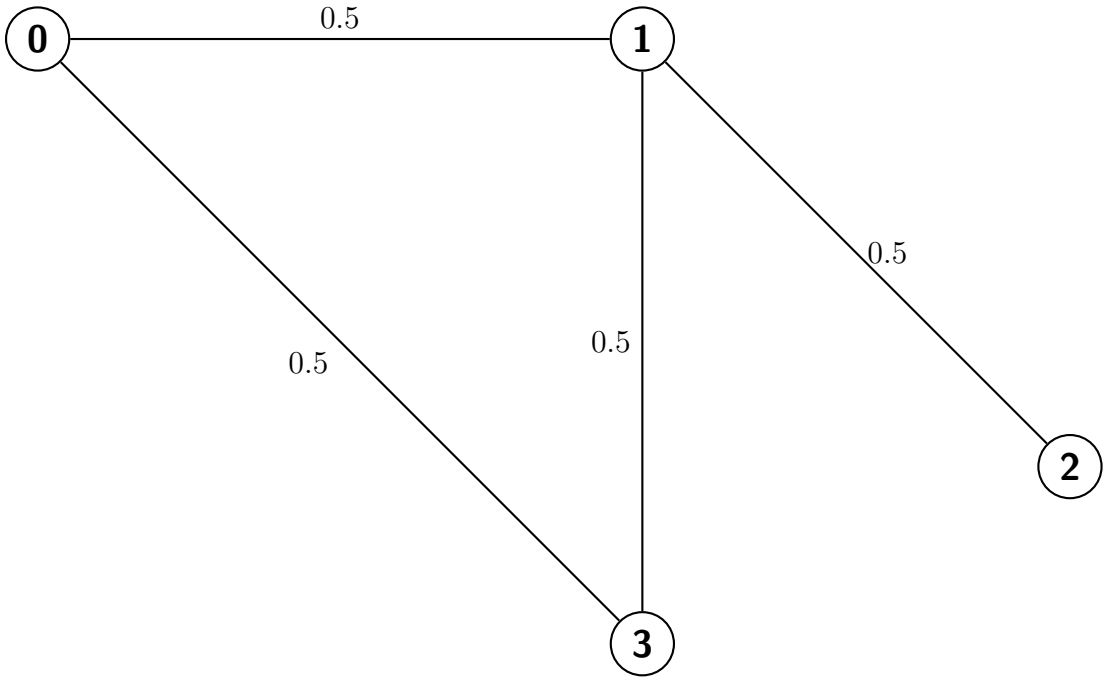

Figure A.1: Probabilistic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, $\mathcal{V} = \{0, 1, 2, 3\}$ and $\mathcal{E} = \{(0, 1), (0, 3), (1, 2), (1, 3)\}$ with $p_{u,v} = 0.5\ \forall (u, v) \in \mathcal{E}$.

USCAN In USCAN the calculation of $P[e, \varepsilon]$ relies on the DP equation:

$$\begin{aligned} X(h, m, n) = {} & p_{wu} p_{wv} X(h, m-1, n-1) \\ & + (p_{wu}(1 - p_{wv}) + p_{wv}(1 - p_{wu}))X(h, m, n-1) \\ & + (1 - p_{wu})(1 - p_{wv})X(h, m, n) \end{aligned}$$

Consider the edge $(0, 1)$ (where $\varepsilon = 0.2$): $|\overline{N_u} \cup \overline{N_v}| = 4$ and $|\overline{N_u} \cap \overline{N_v}| = 3$.
When $h = 1, 2$ let $w = 3, 2$ respectively. Running over the algorithm, start with $X(0, 2, 2) = 1$, for $w = 3$:

$$\begin{aligned} X(1, 2, 2) &= X(0, 2, 2) \cdot 0.5^2 = \frac{1}{4} \\ X(1, 2, 3) &= X(0, 2, 2) \cdot 0.5 = \frac{1}{2} \\ X(1, 3, 3) &= X(0, 2, 2) \cdot 0.5^2 = \frac{1}{4} \end{aligned}$$

then for $w = 2$:

$$
\begin{aligned}
X(2,2,2) &= X(1,2,2) \cdot 0.5^2 = \frac{1}{16} \\
X(2,2,3) &= X(1,2,2) \cdot 0.5 + X(1,2,3) \cdot 0.5^2 = \frac{1}{4} \\
X(2,3,3) &= X(1,3,3) \cdot 0.5^2 + X(1,2,2) \cdot 0.5^2 = \frac{1}{8} \\
X(2,2,4) &= X(1,2,3) \cdot 0.5 = \frac{1}{4} \\
X(2,3,4) &= X(1,3,3) \cdot 0.5 = \frac{1}{8}
\end{aligned}
$$

summing up all the terms where $\frac{m}{n} \geq \varepsilon$, we get:

$$
\begin{aligned}
Pr[(0,1),0.2] &= 0.5 \left(\frac{1}{16} + \frac{1}{4} + \frac{1}{8} + \frac{1}{4} + \frac{1}{8} \right) \\
&= 0.40625
\end{aligned}
$$

ProbSCAN In ProbSCAN $P[e, \varepsilon]$ is determined by:

$$
P[e,\varepsilon] = \sum_{i=0}^{k_{(u,v)}} P[sup(u,v) = i] \times \sum_{j=i+1}^{k_u} P[d_u = j] \times \sum_{k=i+1}^{\min(k_v, i-j+\lceil \frac{i+2}{\varepsilon} \rceil)} P[d_v = k] \quad \text{(A.1)}
$$

Consider the edge $(0,1)$ (where $\varepsilon = 0.2$): The coefficients for the sum indices are: $k_{0,1} = 1, k_0 = 2, k_1 = 3$. Then the sum expands to,

$$
\begin{aligned}
P[(0,1),0.2] = P[sup(0,1) = 0] &\cdot \big(P[d_0 = 1] + P[d_0 = 2]\big) \\
&\cdot (P[d_1 = 1] + P[d_1 = 2] + P[d_1 = 3]) \\
&+ P[sup(0,1) = 1] \cdot (P[d_0 = 2]) \cdot (P[d_1 = 2] + P[d_1 = 3])
\end{aligned}
$$

From the graph we can read off the probabilities of nodes 0 and 1 degrees equaling j and k:

$$P[d_0 = 1] = 0.5^2 \cdot 2 = \frac{1}{2}$$
$$P[d_0 = 2] = 0.5^2 = \frac{1}{4}$$
$$P[d_1 = 1] = 0.5^3 \cdot 3 = \frac{3}{8}$$
$$P[d_1 = 2] = 0.5^3 \cdot 3 = \frac{3}{8}$$
$$P[d_1 = 3] = 0.5^3 = \frac{1}{8}$$

Since all the edges have a probability of $p = \frac{1}{2}$, this implies that all of the 16 possible worlds occur with the same probability $P[G|\mathcal{G}] = \frac{1}{16}$. Hence the probabilities of the $sup(u, v) = 0, 1$ are just the count of the worlds where $(0, 1) \in E$ and they have 0 or 1 neighbours in common, respectively. There are six worlds where (0,1) are connected and share no neighbours; and there are two worlds where (0,1) are connected and share a single neighbour. So we have,

$$P[sup(0, 1) = 0] = \frac{3}{8}$$
$$P[sup(0, 1) = 1] = \frac{1}{8}$$

Now substituting these values into the expansion of equation A.1, we get

$$P[(0, 1), 0.2] = 0.1796875$$

Therefore, ProbSCAN does not give the same output as USCAN.

Appendix B

Sequential Least Squared Programming (SLSQP)

In order to show that condition 4.23 is satisfied for the proof of 2 in Section 4, we rely on numerical methods. Specifically we used an implementation of Sequential Least Squared Programming (SLSQP), a procedure introduced to solve nonlinear programs (NLP). Our problem of minimizing $f(p_{i1}, p_{i2}, \varepsilon)$ with a set of boundary constraints is an example of a NLP where the optimizing function is $f(p_{i1}, p_{i2}, \varepsilon)$ and the constraints are the boundary intervals. The SLSQP method was first developed by Kraft et. al. [18], and has a detailed outline of the algorithm in Section 2.2.4 of the original paper. The method uses Lagrange multipliers—a powerful tool designed to locate the extreme values of a constrained function—to iteratively update the current location and direction of the minimum for the next iteration step. Similar to Newton's method of root finding, the SLSQP algorithm takes a starting point in the region, and uses the Lagrange function to update the position to a point closer to a minimum of the function in the region. The program continually updates the direction of each step it takes based on the Lagrange multipliers.

The python library scipy has an implementation of the SLSQP algorithm that we used to locate all the minima inside the constrained region. The method `scipy.optimize.minimize()` takes as arguments: the function to minimize, initial starting point, the constraint bounds, maximum number of iterations, and the method of minimization—which was chosen to be SLSQP. The program converges to a point that is a global minimum inside the constrained region.

Since these types of algorithms that locate extreme point may be influenced by the starting position, we ran the algorithm with 8000 distinct starting points equally distributed throughout the constrained space. After experimentation, we found all the minima in the constrained region. Every minima found, occurred on a boundary edge of the rectangular constrained volume. The requirement for the function $f(p_{i1}, p_{i2}, \varepsilon)$ is that it remains non-negative in the bounded space. The numerical method SLSQP was able locate the minima of the function $f(p_{i1}, p_{i2}, \varepsilon)$ in the bounded rectangular region; furthermore all the minima resulted in a non-negative value for the function $f(p_{i1}, p_{i2}, \varepsilon)$. So we conclude that condition 4.23 holds on the bounded volume, and are thus able to show that the limit in equation 4.29 converges to zero.

Bibliography

[1] Anupam Biswas and Bhaskar Biswas. Defining quality metrics for graph clustering evaluation. *Expert Systems with Applications*, 71:1–17, 2017.

[2] Matteo Ceccarello, Carlo Fantozzi, Andrea Pietracaprina, Geppino Pucci, and Fabio Vandin. Clustering uncertain graphs. *Proceedings of the VLDB Endowment*, 11(4):472–484, 2017.

[3] Lijun Chang, Wei Li, Lu Qin, Wenjie Zhang, and Shiyu Yang. pscan: Fast and exact structural graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, 29(2):387–401, 2017.

[4] Sudarshan S Chawathe. Clustering blockchain data. In *Clustering Methods for Big Data Analytics*, pages 43–72. Springer, 2019.

[5] Yulin Che, Shixuan Sun, and Qiong Luo. Parallelizing pruning-based graph structural clustering. In *Proceedings of the 47th International Conference on Parallel Processing*, pages 1–10, 2018.

[6] Jia-Jun Chen, Ji-Meng Chen, Jie Liu, and Va-Lou Huang. Pscan: A parallel structural clustering algorithm for networks. In *2013 International Conference on Machine Learning and Cybernetics*, volume 2, pages 839–844. IEEE, 2013.

[7] Norishige Chiba and Takao Nishizeki. Arboricity and subgraph listing algorithms. *SIAM Journal on computing*, 14(1):210–223, 1985.

[8] Alfredo Cuzzocrea, Edoardo Fadda, and Alessandro Baldo. Lyapunov central limit theorem: Theoretical properties and applications in big-data-populated smart city settings. In *2021 5th International Conference on Cloud and Big Data Computing (ICCBDC)*, pages 34–38, 2021.

[9] Malihe Danesh, Morteza Dorrigiv, and Farzin Yaghmaee. Dgcu: A new deep directed method based on gaussian embedding for clustering uncertain graphs. *Computers and Electrical Engineering*, 101:108066, 2022.

[10] Can Ding, Jing Zhang, Yingjie Zhang, Zhe Zhang, and Xiaoyao Li. Neural network-based adp cotrol for nonliear systems with prescribed performance constriant. In *2021 33rd Chinese Control and Decision Conference (CCDC)*, pages 6342–6346. IEEE, 2021.

[11] Fatemeh Esfahani, Mahsa Daneshmand, Venkatesh Srinivasan, Alex Thomo, and Kui Wu. Scalable probabilistic truss decomposition using central limit theorem and h-index. *Distributed and Parallel Databases*, 40(2):299–333, 2022.

[12] Fatemeh Esfahani, Venkatesh Srinivasan, Alex Thomo, and Kui Wu. Efficient computation of probabilistic core decomposition at web-scale. In *EDBT*, pages 325–336, 2019.

[13] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[14] Zahid Halim, Muhammad Waqas, Abdul Rauf Baig, and Ahmar Rashid. Efficient clustering of large uncertain graphs using neighborhood information. *International Journal of Approximate Reasoning*, 90:274–291, 2017.

[15] Kai Han, Fei Gui, Xiaokui Xiao, Jing Tang, Yuntian He, Zongmai Cao, and He Huang. Efficient and effective algorithms for clustering uncertain graphs. *Proceedings of the VLDB Endowment*, 12(6):667–680, 2019.

[16] Syed Fawad Hussain, Ifra Arif Butt, Muhammad Hanif, and Sajid Anwar. Clustering uncertain graphs using ant colony optimization (aco). *Neural Computing and Applications*, pages 1–18, 2022.

[17] Yun Joong Kim, Kiyong Kim, Heonwoo Lee, Junbeom Jeon, Jinwoo Lee, and Jeehee Yoon. The protein-protein interaction network of hereditary parkinsonism genes is a hierarchical scale-free network. *Yonsei medical journal*, 63(8):724, 2022.

[18] Dieter Kraft. A software package for sequential quadratic programming. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.

[19] Yongjiang Liang, Tingting Hu, and Peixiang Zhao. Efficient structural clustering in large uncertain graphs. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1966–1969. IEEE, 2020.

[20] Ian McCulloh, Joshua Lospinoso, and KM Carley. Social network probability mechanics. In *Proceedings of the World Scientific Engineering Academy and Society 12 th International Conference on Applied Mathematics*, 2007.

[21] Naoto Ohsaka, Tomohiro Sonobe, Sumio Fujita, and Ken-ichi Kawarabayashi. Coarsening massive influence networks for scalable diffusion analysis. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 635–650, 2017.

[22] David Ong and Jue Wang. Income attraction: An online dating field experiment. *Journal of Economic Behavior & Organization*, 111:13–22, 2015.

[23] Symeon Papadopoulos, Ioannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media. *Data Min. Knowl. Discov.*, 24:515–554, 05 2012.

[24] Yu-Xuan Qiu, Rong-Hua Li, Jianxin Li, Shaojie Qiao, Guoren Wang, Jeffrey Xu Yu, and Rui Mao. Efficient structural clustering on probabilistic graphs. *IEEE Transactions on Knowledge and Data Engineering*, 31(10):1954–1968, 2018.

[25] Panyu Ren, Xiaodi Yang, Tianpeng Wang, Yunpeng Hou, and Ziding Zhang. Proteome-wide prediction and analysis of the cryptosporidium parvum protein-protein interaction network through integrative methods. *Computational and Structural Biotechnology Journal*, 2022.

[26] Boyu Ruan, Junhao Gan, Hao Wu, and Anthony Wirth. Dynamic structural clustering on graphs. In *Proceedings of the 2021 International Conference on Management of Data*, pages 1491–1503, 2021.

[27] Jung Hyuk Seo and Myoung Ho Kim. pm-scan: an i/o efficient structural clustering algorithm for large-scale graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2295–2298, 2017.

[28] Brian Skyrms and Robin Pemantle. A dynamic model of social network formation. In *Adaptive networks*, pages 231–251. Springer, 2009.

[29] Thomas Ryan Stovall, Sinan Kockara, and Recep Avci. Gpuscan: Gpu-based parallel structural clustering algorithm for networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(12):3381–3393, 2014.

[30] Peng Xia, Kun Tu, Bruno Ribeiro, Hua Jiang, Xiaodong Wang, Cindy Chen, Benyuan Liu, and Don Towsley. Characterization of user online dating behavior and preference on a large online dating site. In *Social Network Analysis-Community Detection and Evolution*, pages 193–217. Springer, 2014.

[31] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas AJ Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 824–833, 2007.

[32] Samuel Zahl. Bounds for the central limit theorem error. *SIAM Journal on Applied Mathematics*, 14(6):1225–1245, 1966.

[33] Xianchao Zhang, Han Liu, and Xiaotong Zhang. Novel density-based and hierarchical density-based clustering algorithms for uncertain data. *Neural networks*, 93:240–255, 2017.

[34] Xianchao Zhang, Han Liu, Xiaotong Zhang, and Xinyue Liu. Novel density-based clustering algorithms for uncertain data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[35] Weizhong Zhao, Venkataswamy Martha, and Xiaowei Xu. Pscan: a parallel structural clustering algorithm for big networks in mapreduce. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pages 862–869. IEEE, 2013.