STUDY OF COMMUNITY AWARE NETWORK EXPANSION

Student Name: Sarika

IIIT-D-MTech-CS-DE-18-MT16051
June, 2018

Indraprastha Institute of Information Technology
New Delhi

Thesis Advisors
Dr. Vikram Goyal
Dr. Tanmoy Chakraborty

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science,
in Data Engineering Category

©2018 IIIT-D-MTech-CS-DE-18-MT16051
All rights reserved

Keywords: Community detection, Network expansion

Certificate

This is to certify that the thesis titled **"Study of Community Aware Network Expansion"** submitted by **Sarika** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her under our guidance and supervision at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Vikram Goyal

Indraprastha Institute of Information Technology, New Delhi

Dr. Tanmoy Chakraborty

Indraprastha Institute of Information Technology, New Delhi

Abstract

Community detection has gained immense popularity in recent years. Real world networks consist of millions of nodes and edges. Groups of nodes exhibit interesting characteristics, whose knowledge can be of great help in various fields. Moreover, networks in the real world are ever evolving and it is impossible to have complete information about a network at any given time. The aim of this thesis is to study the different techniques for the exploration of the incomplete network available. The motive is to explore the incomplete network such that nodes which are in the community of the known nodes are brought into the network. We build a machine learning model to predict which node should be explored. We study four methods for selecting the next node to be explored namely, BFS, Random, Machine Learning and Maximum global clustering coefficient. For identifying the communities of the nodes of the incomplete network we study three algorithms namely, community detection by hopcount, community detection by maximizing modularity, and community detection by maximizing permanence. We observe that Machine learning classifier is the best approach to maximize the recall by exploring least number of nodes, whereas global clustering coefficient is the best approach for maintaining high precision. BFS gives better f1-score as compared to other approaches for higher budget. The community detection algorithms are performing equally well. Though with only a slight margin, HopCount method of community detection gives better results for recall and permanence maximization gives better results for precision.

Acknowledgments

I would like to express my deepest gratitude to my advisor Dr. Vikram Goyal and Dr. Tanmoy Chakraborty for their guidance and support. I would like to thank them for their mentorship at every stage of this thesis work.The door to their office was always open whenever I had a doubt. I would not have been able to complete my thesis work this smoothly in IIIT Delhi without their handholding and consistent support.

I would like to thank my supportive family and friends who encouraged me and kept me motivated throughout the thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Community detection can be defined as the process of identifying groups of nodes in a network, which are closely related to each other. These groups of nodes are called communities. Generally, nodes in a community can be characterized as having high intra-community edges and few inter-community edges. In recent years, there has been an increased interest in the field of community detection. This can be explained by the fact that communities help uncover various structural properties of the network. They provide an insight into the various properties exhibited by the nodes of the network. Many algorithms have been designed to identify communities in networks. These algorithms can be classified into two types: static and dynamic. Static community detection algorithms are applied on networks in which the interaction between the nodes does not change with time. Dynamic community detection algorithms are applied to evolving networks, in which nodes and edges may get added or removed from the network over time. Also, dynamic community detection algorithms make use of the already identified communities and do not start community detection from scratch. This helps save time and space. Real world networks are dynamic in nature. Moreover, at any given instance we only have limited knowledge about the network. This is due to various reasons such as high cost of discovering the entire network, limited budget, privacy constraints, etc . We are required to explore this limited network to gain more knowledge about the underlying network. This strategy is termed as network expansion or network discovery. In this strategy we explore the available nodes to get more detail about the underlying network. There are various aims for exploring the limited part of a network. For example, to introduce maximum number of nodes into the incomplete network within a given budget [4], or if all the known nodes belong to the same community and complete network is known, then finding the other nodes in the community of the known nodes called seed set expansion, [11]. Most of the works done in the past are for the above two purposes. However, our aim is to explore the incomplete network, containing nodes from different communities, in such a way that nodes from the known node communities are introduced into the network. One of its applications could be identifying authors in the community of some particular authors in a co-authorship network, discovering proteins which fall under the same category (community)

as some given proteins, uncovering the network of local goons of an area, etc. Suppose, police wants to uncover the network of local goons. They are aware of one local goon from each gang of an area. Now the aim is to identify other members of their gang. In addition to this, they would also be required to assign them to one of the gangs for better identification. This implies that there are two parts to this problem; one is the identification of goons and second is assigning them to a gang.

In this thesis, we try to identify the best method of exploring the given seed networks so as to find the nodes in their community. This is a two step procedure. In the first step, we decide the node to explore and in the second step we apply community detection algorithm. We provide a comparative study of the combination of methods used in the two steps. For the first step, we choose the node in (i) BFS order, (ii) random order, (iii) high global clustering value estimated using MaxReach algorithm, and (iv) by training a classifier to predict whether a node is good enough to be explored (ML approach). For the second step we have used dynamic community detection algorithms, as new nodes and edges get entered, the network size would increase very fast. Using static algorithms may take a lot of time(as they start community detection from scratch) or get trapped in a local optimum or same reaction to small changes in local area of network [2].

1.1 Problem Statement

Given a seed network G', a set of seed nodes T and a budget B, which is the best method to discover the community structure around the seed nodes within the given budget by exploring the nodes and assigning them to a community.

1.2 Some terminology

- **Seed nodes or Target nodes:** These are the initial nodes which are known to exist and around which we have to identify the communities.
- **Induced Subgraph:** An induced subgraph on a set of nodes is a graph with all edges that exist between any pair of those nodes.
- **Seed Network, G':** It is the induced subgraph on the seed nodes.
- **Explored Nodes:** Nodes whose all neighbors have been identified.
- **Discovered Nodes:** Nodes which are known to exist but all their neighbors have not been identified.
- **Complete network, G:** The underlying network from which seed nodes were selected. We have no knowledge about this underlying network.

- **Central nodes:** The nodes in the community whose all neighbors are within the community are known as central nodes
- **Boundary nodes:** Nodes which have neighbors in different communities are known as boundary nodes.
- **Disjoint community structure:** If nodes do not belong to multiple communities then this community structure is called disjoint community structure. Figure 1 shows two communities; it can be seen that there are more edges between nodes of the same community than inter-community edges. Also it can be seen that no node belongs to both the communities, so this is disjoint community structure.
- **Overlapping community structure:** When a node may be present in more than one community, such a community structure is known as overlapping community structure. Figure 2 shows overlapping community structure.

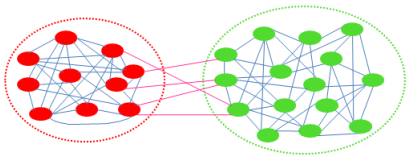

Figure 1.1: Disjoint Communities

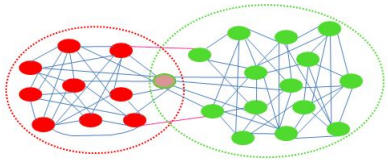

Figure 1.2: Overlapping communities

1.3 Outline

The rest of the thesis is organized as follows: Section 2 talks about the various works in the field of community detection. In section 3 we discuss the approaches used for selecting the next node to explore and community detection. In section 4, we discuss the datasets used and results. In section 5, we conclude the thesis and propose future work.

Chapter 2

Related Work

We divide related work into three sections. First section will describe the metrics that are prevalent for identifying communities; second section talks about the algorithms on incomplete networks and third section will reflect on the application of machine learning on networks.

2.1 Metrics community detection

Community is an ill-defined function [1]. There are various ways in which a community can be defined. There exist many different definitions of communities based on different metrics used to identify them [3]. The metrics used for community detection can be broadly classified into four categories [12]: (i) metrics related to internal connectivity, (ii) metrics pertaining to external connectivity, (iii) metrics using both internal and external connectivity and (iv) metric based on network model. The most widely used metric for identifying communities is modularity. It belongs to the fourth category of metrics i.e., metrics based on network model. Modularity was proposed by [7] and takes values between -1 to 1. If the value is close to -1, then it signifies that nodes in this community are sparsely connected and are worse than random distribution of edges between them. If the value is close to 1, then it means that the community is a densely connected community and that there is less probability that this by chance that they belong to the same community. Modularity suffers from the problem of resolution limit,i.e., it is difficult to find small communities using modularity. Other variations of modularity have been proposed to overcome these issues. A node-centric metric, Permanence, was proposed by [8]. It estimates how likely the node is to stay in its present community. It captures the interaction of the nodes with nodes within its community and nodes in other communities. It value is in the range [-1,1]. A value closer to -1, means the node is highly unlikely to stay in the present community and the interactions with other community is stronger than this community. A value closer to 1, signifies that the interactions of the node within this community are far stronger than the interactions with nodes outside the community.

2.2 Algorithms proposed for incomplete networks

Most of the community detection algorithms proposed are based on community detection on complete networks. Algorithms on incomplete network involve inferring the missing nodes and edges of the network [13], community detection on incomplete network, identifying community members of seed nodes, introducing nodes into the incomplete network, etc. [5] presents techniques to estimate the global network statistics of the underlying network using the information from the incomplete network. The authors of [4], estimate the true degree, clustering coefficient of a node and number of edges of a node that are still unknown. They make use of these statistics to bring many nodes into the incomplete network. Most of the approaches proposed for seed set expansion [11], start with seed nodes belonging to the same community and aim at identify the remaining members of their community. Other approaches aim at discovering the overlapping community structure of large network by first choosing some good seed nodes from every community and then expanding them. In [16] the authors showed that network discovery and community detection can be performed simultaneously. However, there approach suffered from information leak.

2.3 Machine learning

Machine learning has been applied for estimating centrality measures of complete graphs [6], social circle detection in ego networks using learning [14] etc. The authors of [6] have shown that low centrality measures like degree,betweenness, closeness and eccentricity centrality measures can be used to calculate high centrality measures like, eigenvector,information centrality. In [14],the authors aim at identifying overlapping communities in ego networks. They use an unsupervised method to assign each node to a circle(community). However, applying machine learning techniques in order to determine the next node to explore in network expansion has still not be performed.

Chapter 3

Algorithms for Community Aware Network Expansion

Community Aware Network Expansion involves two steps: network expansion and community detection. The algorithms are hence divided into two categories: those that are used for network expansion and those for community detection. Flowchart in figure 3.1 shows the tasks that are performed.

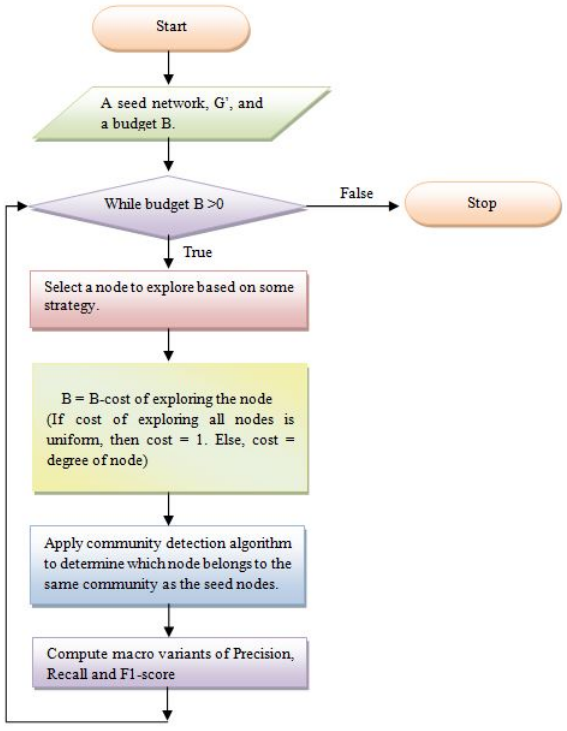

Figure 3.1: Flow chart

We have as input a seed network G and a budget B. The budget may correspond to the number of

nodes that can be explored or the number of interactions between nodes that can be observed. We then apply network expansion techniques followed by identifying communities. In the following sections we describe the algorithms.

3.1 Algorithms for Network expansion

3.1.1 BFS

In this approach, we explore the nodes on a first come basis. The nodes that were brought into the network first are explored first. It means that nodes which are one hop away from the seed nodes would be explored first then nodes that are two hop away would be explored and so on. We explore the nodes at one level before moving to the next level nodes.

3.1.2 Random

In this approach, we select a node at random to be explored from the list of discovered nodes. This is the most naive approach for expanding the network.

3.1.3 High global clustering coefficient

In this approach we select the node which has the highest global clustering coefficient. Since, we have incomplete network at hand, we need to estimate the global clustering coefficient of each known node. For estimating the clustering coefficient we use the method proposed in [4]. It first predicts the degree distribution of the underlying network, followed by estimating the clustering coefficient corresponding to a true degree value. A node's true degree is predicted and the clustering coefficient corresponding to that true degree is assigned to the node.

3.1.4 Machine learning model of classification

The motive behind choosing machine learning model as a node selection method is that we wish to explore how node and graph attributes affect the goodness of a node .In this approach, we have used a trained classifier, which predicts whether a node is good enough to be selected for exploration. For every discovered node, we compute the various features and feed them into the classifier. If the output of the classifier is 1, then the node can be selected for exploration. However, if the classifier predicts zero for all the discovered nodes, then we predict the probability with which a node is assigned to the zero class. Node with the least probability of belonging to zero class is selected for exploration.

3.1.5.1 Building a Classification Model

We have trained a machine learning model for predicting a good node to explore from a list of possible nodes. We predict whether a exploring a node will give us a good f1-score depending on the node and graph attributes. Below we explain how machine learning model was trained.

(i) Generating graphs for training

Generally, real world networks follow power-law degree distribution. We created 500 scale free graphs having power-law degree distribution. We determine their community structure by running fast incremental community detection algorithm from [10]. For each graph we created 3 seed networks of sizes 100,250 and 500. Now the obtained seed networks are explored and training sample created.

(ii) Features

Figure 3.2 shows the features that were able to best classify the nodes into the two classes. A brief discussion of these features follow:
1. Assortativity of a graph signifies the tendency of nodes to attach to other nodes of the same characteristic. Computing assortativity based on degree is the most widely used metric. Degree assortativity is one of the features used in our ML model. We have used a function provided by networkx which uses pearson correlation coefficient for estimating the assortativity of graph.

2. Closeness of a node can be described as reciprocal of the sum of the shortest path between the node and other nodes in the network. It signifies how easily other nodes are reachable from this node. Higher the value of closeness, closer it is to all nodes of the network.

3. Degree of a node has also been taken as one of the features in our machine learning model. It is the number of nodes that are directly connected to the node.

4. Eccentricity of a node is the largest distance between the node to any other node using the shortest path between them.

5. Diameter of a graph is the maximum distance between any two nodes using the shortest path between them.

6. Radius of a graph can be defined as the minimum of the eccentricities of all the nodes.

7. PageRank of a node signifies how many important nodes of the network are related to the node. It helps in knowing which node will yield nodes that might be good to explore.

8. Clustering coefficient of a node tells the probability with which it forms clusters with other nodes.

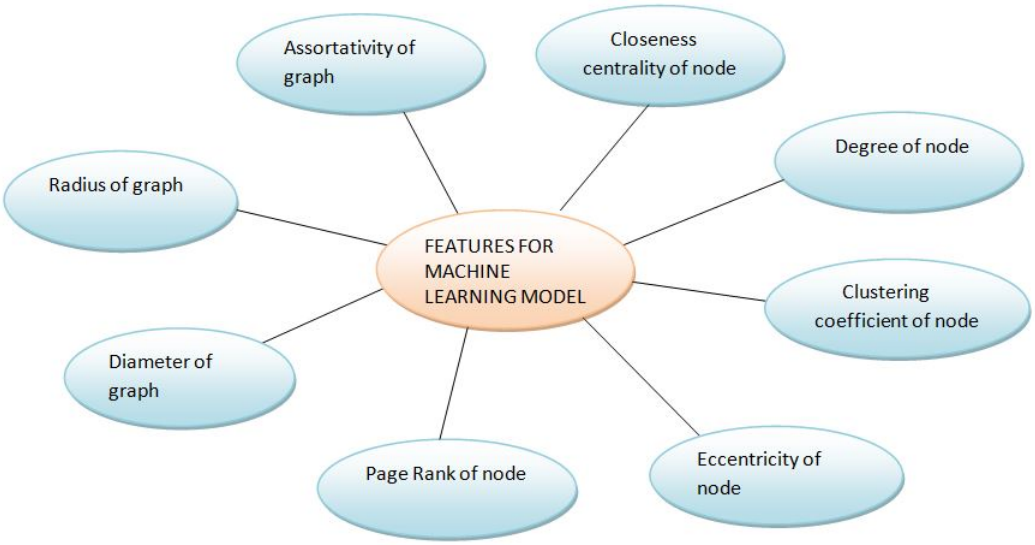

Figure 3.2: Features

(iii) Training Data.

The first step was to assign a class to each of the data points. We assign a class 1 to data points whose f1-score is greater than a threshold, and class 0 otherwise. We vary the threshold of F1-score to decide which one gives us the best results. Some of the feature values were large, whereas some were really small, therefore we applied scaling on the features. We used min-max scaler to perform the task of scaling. Also, since the the number of class zero data points was far more than class one datapoints, we performed undersampling. Undersampling was preferred over oversampling because the number of data points was huge. Moreover, to improve the training process we shuffled the data so that continuous zero class and continuous class one data points do not degrade the learning process. We perform grid search to find the best parameters for the different classifiers.

(iv) Classification Models.

We train models based on the features described above. We have applied SVM, Random Forest and MultiLayerPerceptron. We use gridsearch to find optimal values of the hyper-parameters C,alpha,hidden layers and activation function. It was observed that multilayer perceptron performed better than the other models. Moreover, RandomForest was the slowest as number of trees used for making the decision were 50(obtained using grid search).

3.2 Algorithms for community detection

3.2.1 HopCount

We assign the node to the community of the nearest seed node. The nearness is defined by the number of edges between the seed node and the node. In this approach, whenever a new node gets introduced into the graph, it is assigned the community of its neighbor and its hopcount is one more than its neighbor. If a new edge gets added, then if it an intra-community edge, we perform the bfs on this community starting from the seed node of this community to obtain the updated hopcounts. Else, if the new edge is inter-community edge then bfs is applied on both communities, starting from their respective seed nodes. A node is assigned to the community of the seed node from which its hopcount is least. In case hopcount for a node is same for more than one seed node, then it would be assigned to both the communities.

3.2.2 Maximizing modularity

We use the approach proposed in [2] for community detection by maximizing modularity. [2] proposed a new approach for dynamic community detection which optimizes modularity function. They highlight the drawbacks of using static community detection algorithms in ever-evolving networks. In their algorithm, they handle the situations of node insertion, new edge addition, node removal and edge removal. However, since our method requires only node insertion and edge insertion we describe these two only. When a new node gets added, a new community is formed for it. Then it is decided whether this new node should be merged with the community of its neighbors or form a separate community. They first determine whether there are any neighbors of the new node that might break away from their community and form a new community with this new node. If no such nodes exist, then the new node decides which community it would join. Two forces have been used for this purpose. One force corresponds to the pull that a node observes from its community and second force corresponds to the maximum pull observed by the node from the other communities. These forces are inspired from [15]. In case of edge addition, if u and v are the endpoints of the edge, then they compute the difference in modularity score when u moves to v's community and v moves to u's community. If both the differences are less than zero, then both nodes stay in their respective community. Else, moving whichever node gives larger difference changes its community, followed by reassignment of its neighbors to whichever community they prefer.

3.2.3 Maximizing permanence

This is the dynamic community detection algorithm in which the node to the community which increases the permanence of the node as well as the permanence of the community. This algorithm was proposed in [9]. In this algorithm, they handle the situations of node insertion, edge insertion, node removal and node deletion. In our approach, we need only node insertion and edge deletion,

so we will be highlighting these two functions of [9]. When a new node gets added, a new community is formed for it. Then it is decided whether this new node should be merged with the community of its neighbors. This decision task is performed by repetitively calling the edge insertion module. In case of edge insertion, if u and v are the endpoints of the edge, and both u and v are in the same community then no reassignment of community takes place. However, if u and v belong to different communities, then they compute the difference in permanence of the community by (i)moving u to v's community and iteratively moving u's neighbors to v's community if their permanence values increase and (ii) moving v to u's community and iteratively moving v's neighbors to u's community if their permanence values increase. From the above movements whichever movement increases the permanence of the communities, is performed.

3.3 Number of neighbors retrieved at a time

At times some APIs impose a constraint on the number of edges of a node that can be retrieved at once. In this section we highlight two approaches based on the number of neighbors of a node that can be retrieved at any given instance [4]. One is all-neighbor probing and second is k-neighbor probing.

All neighbor probing: In this approach, exploration of a node leads to all its edges being introduced into the network.
k-neighbor probing: In this approach, only k edges of a node are introduced into the network at a time. The edges once retrieved will not be returned again.

We use All-neighbor probing for analysis of techniques for network expansion and community detection. We also experimentally determine the affects of varying k by keeping network expansion and community detection technique constant.

3.4 Example demonstrating the above techniques

Figure 3.3 shows the example graph. Figure 3.4 shows the communities in the example graph. Nodes having same color belong to the same community. Figure 3.5 shows the seed network. Figure 3.6 is obtained by using a small part of the budget to bring in all neighbors of the seed nodes. Figure 3.7 shows the node that gets selected using bfs approach. Figure 3.8 shows the node that gets selected at random. Figure 3.9 shows the node that gets selected using highest global clustering coefficient approach. Figure 3.10 shows the node that gets selected using machine learning approach. Figure 3.11 shows the node selection using highest global degree approach. The figures are on the next page. In the shown example since the nodes are disconnected components, therefore all the community detection algorithms we output the same communities at the shown intermediate states.

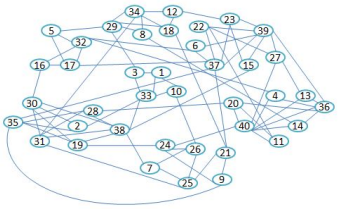

Figure 3.3: Example graph

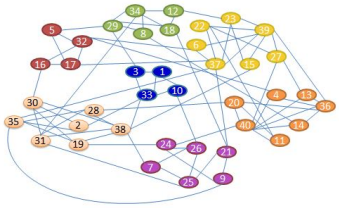

Figure 3.4: Communities of example graph

Figure 3.5: Seed network

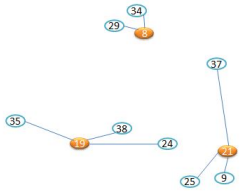

Figure 3.6: Network after exploring seed nodes

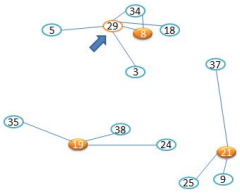

Figure 3.7: Node selection using bfs

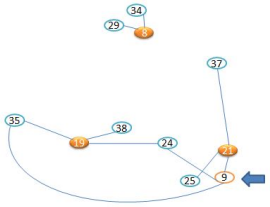

Figure 3.8: Node selection using random approach

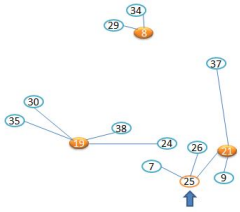

Figure 3.9: Node selection using global CC

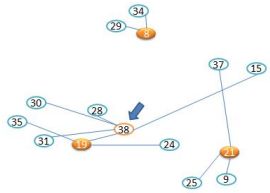

Figure 3.10: Node selection using ML approach

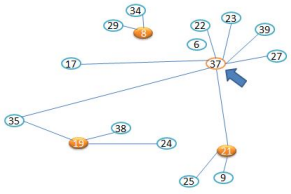

Figure 3.11: Node selection using global degree

Chapter 4

Experiments

4.1 Dataset used

Table 4.1: Datasets.

Dataset	No. of nodes	No. of Edges	No. of Communities	Average Degree
DBLP	317080	1049866	13477	6.62
Amazon	334863	925872	75149	5.53
LFR	30000	95717	1086	6.41

DBLP is a co-authorship graph. The nodes in this graph represents authors, and an edge connecting two nodes represents their collaboration on a paper. This dataset was taken from [17]. It consists of 317080 nodes and 1049866 edges. The number of communities in the ground truth are 13477. The average degree of nodes in DBLP graph is 6.62 and average community size is 53.4. Amazon dataset is a co-purchasing graph. The nodes in this graph represents products on the website, and an edge connecting two nodes represents that the two products are brought together. It consists of 334863 nodes and 925872 edges. The number of communities in the ground truth are 75,149. The average degree of nodes in Amazon graph is 5.53. Amazon dataset was taken from [17]. The details have been tabulated in table 4.1. Figure 4.1 and 4.2 shows the degree distribution of DBLP and Amazon dataset respectively. LFR dataset is an artificial dataset created using Lancichinetti—Fortunato—Radicchi (LFR) benchmark. It creates networks that resemble real world networks. We created this dataset with mu equal 0.2. mu controls the number of intra-community and inter-community edges.

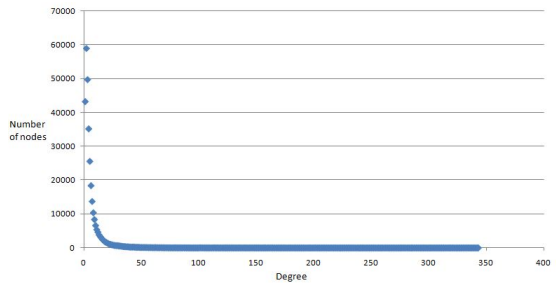

Figure 4.1: Degree Distribution of DBLP.

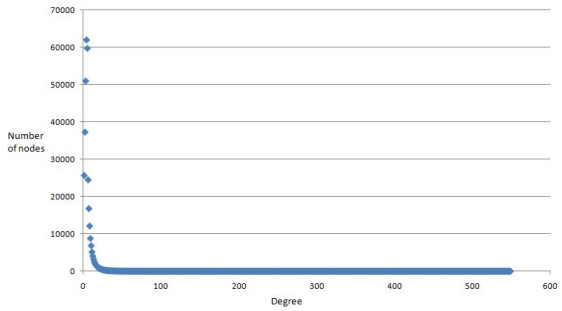

Figure 4.2: Degree Distribution of Amazon.

4.2 Metrics used for evaluation

To check the correctness of the nodes explored and the communities thereby detected, we have used precision, recall and f1-score. We have used macro-variants of the above metrics. This is because say in one sample graph, if there are X nodes and one seed node has a large number of nodes in its community which are identified and its ground truth also has many nodes in its community; then even though communities around other nodes are less identified, the micro variants will produce good results;thereby showing bias towards such nodes. Macro-variants give equal weightage to each of the nodes. Below, we list the formulas used for computing the metrics for one sample graph.

4.2.1 Precision

$$MacroPrecision = \frac{1}{Numberofseednodes} * \sum_{n \epsilon seednodes} \frac{TruePositive(n)}{NodesIdentifiedincommunityof(n)} \tag{4.1}$$

4.2.2 Recall

$$MacroRecall = \frac{1}{Numberofseednodes} * \sum_{n \epsilon seednodes} \frac{TruePositive(n)}{Nodesingroundtruthcommunityof(n)} \tag{4.2}$$

4.2.3 F1-score

$$F1score = \frac{2 * MacroPrecision * MacroRecall}{MacroPrecision + MacroRecall} \tag{4.3}$$

4.3 Selection of seed nodes

The selection of seed nodes was done such that they belonged to pairwise disjoint communities and belong to only one community. For one seed network, we selected 100 seed nodes for dblp and amazon dataset, while 40 nodes were selected for the artificial dataset. The seed networks

selected had central nodes as well as boundary nodes. The experimental results shown for each dataset have been averaged over 16 seed networks.

4.4 Comparison of network expansion techniques

In this section we show the results of varying the network expansion techniques while keeping the same community detection technique.

4.4.1 Detection technique used: Hopcount

In this subsection, the results are for different network exploration technique with community detection technique as hopcount.

DBLP dataset

Figure 4.3 to 4.8 show precision, recall and f1-score for dblp dataset. In figures 4.3 to 4.5, the cost of exploring a node is equal to 1. In figure 4.6 to 4.8 the cost of exploring a node is equal to its degree.

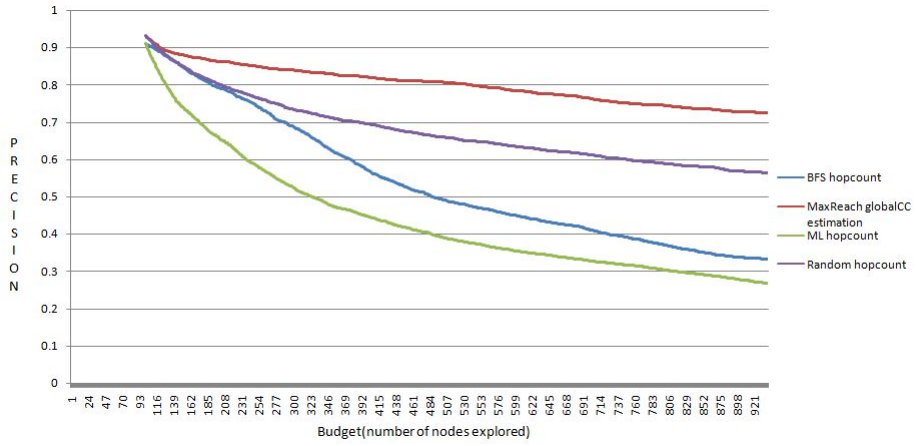

Figure 4.3: Precision for different network expansion techniques where cost of exploring a node equal to one

We observe that selecting node on the basis of highest estimated clustering coefficient gives good precision and low recall even after several nodes have been explored. This is due to the fact that the degree of nodes with high clustering coefficients is usually low. So, few new nodes are getting added into the network. BFS gives a good recall and f1score. This is because all the possible neighborhood is explored. However, due to this reason the precision value gets effected. Machine learning approach is initially able to identify nodes that have higher connections within the seed node community. Random approach selects a combination of nodes with high degree, high clustering coefficient and nodes near to the seed nodes. Its evaluation metrics are close to

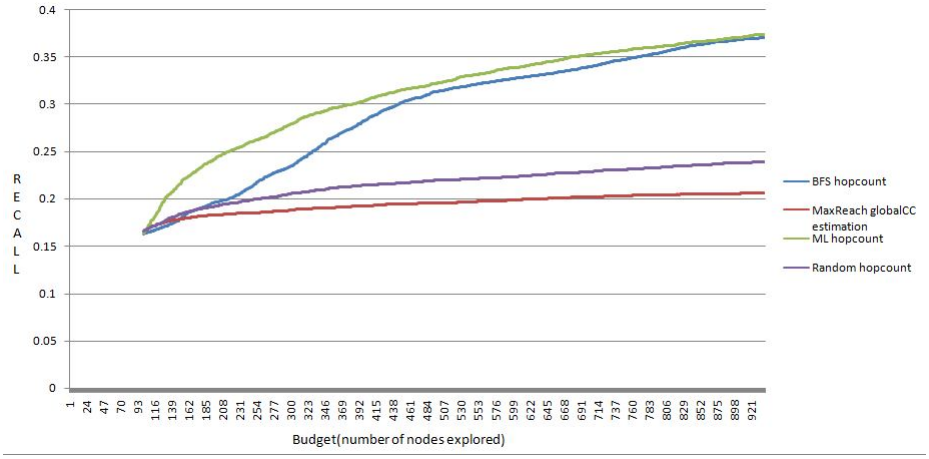

Figure 4.4: Recall for different network expansion techniques where cost of exploring a node equal to one

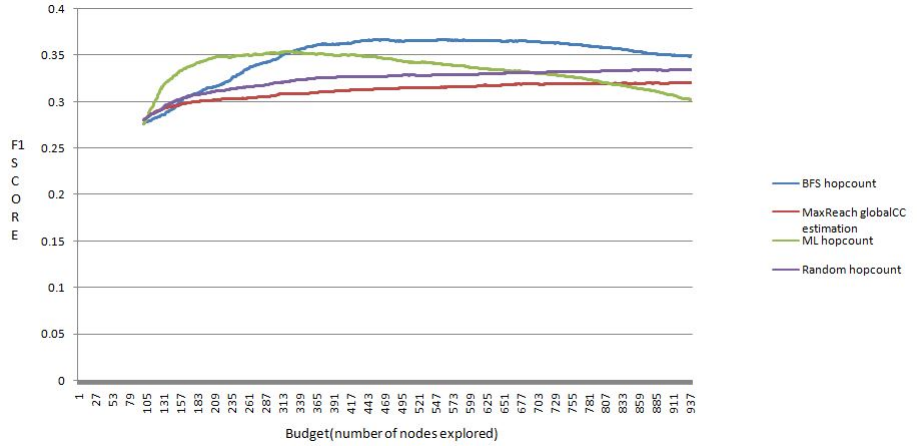

Figure 4.5: F1score for different network expansion techniques where cost of exploring a node equal to one

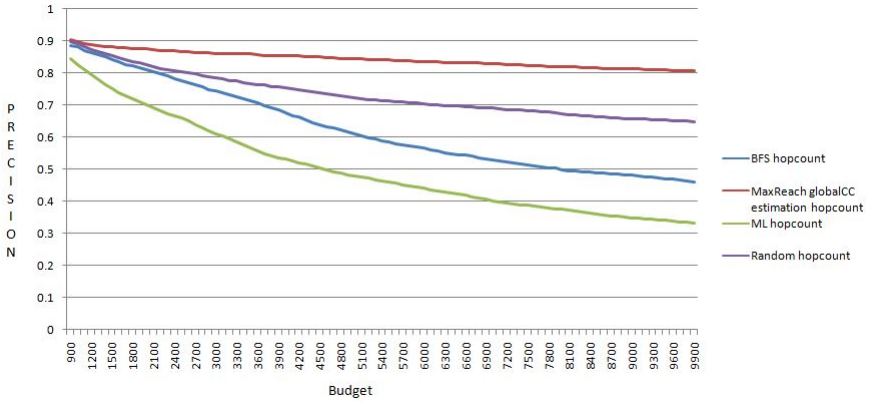

Figure 4.6: Precision for dblp dataset with cost of exploring node equal to its degree

average of the other approaches metrics. Similar results are obtained for cases where cost of exploring a node is 1 and cost of exploring the node is equal to its degree.

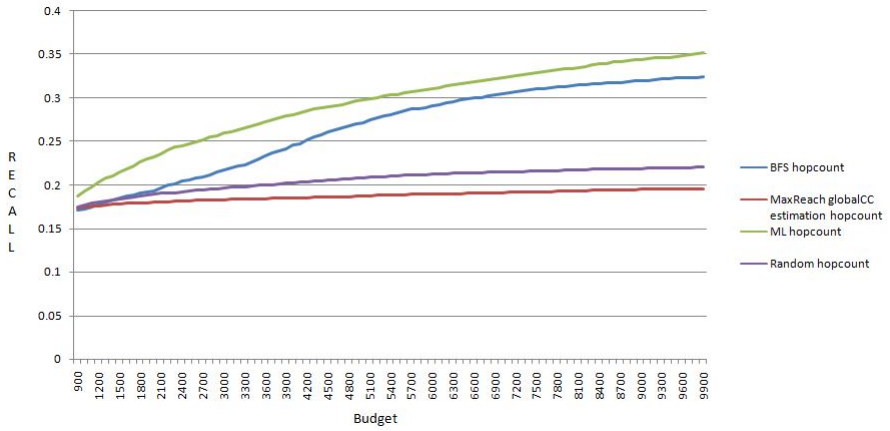

Figure 4.7: Recall for dblp dataset with cost of exploring node equal to its degree

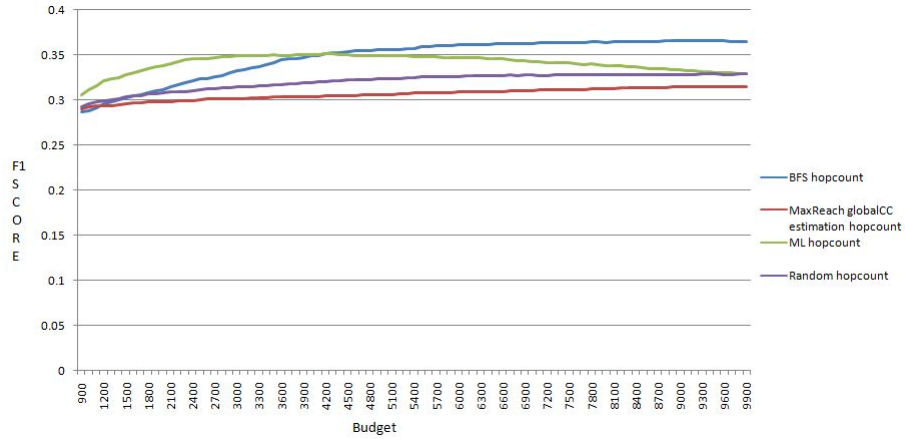

Figure 4.8: F1score for dblp dataset with cost of exploring node equal to its degree

Amazon dataset

Figure 4.9 to 4.14 show precision, recall and f1-score for amazon dataset. In figures 4.9 to 4.11 the cost of exploring a node is equal to 1. In figure 4.12 to 4.14 the cost of exploring a node is equal to its degree.

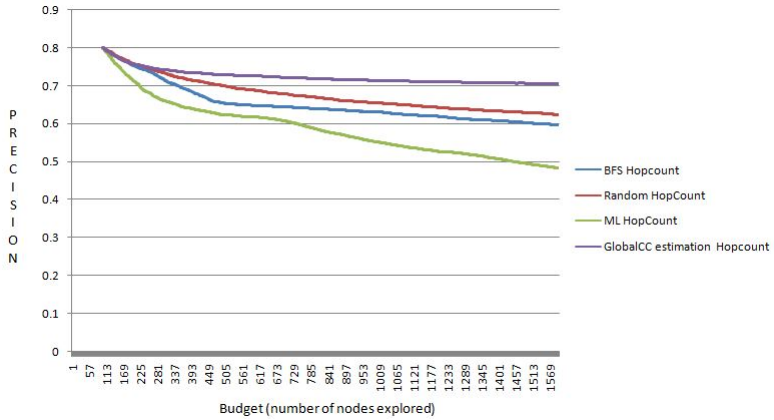

Figure 4.9: Precision for different network expansion techniques Amazon dataset

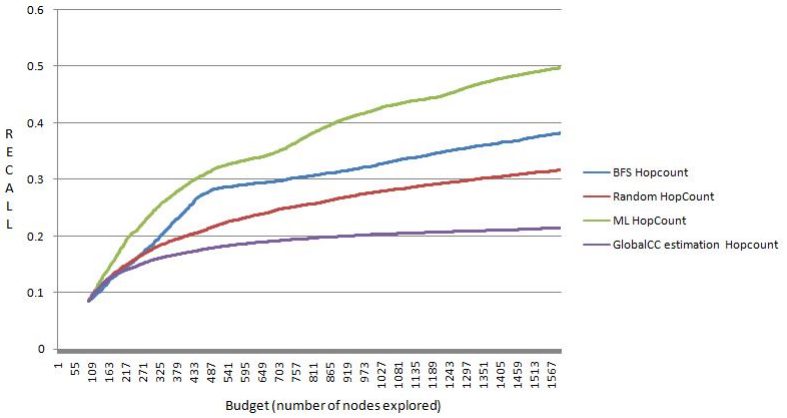

Figure 4.10: Recall for different network expansion techniques Amazon dataset

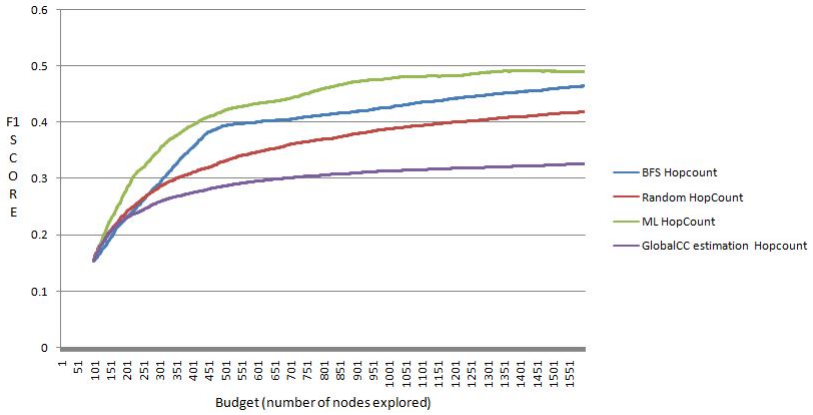

Figure 4.11: F1score for different network expansion techniques Amazon dataset

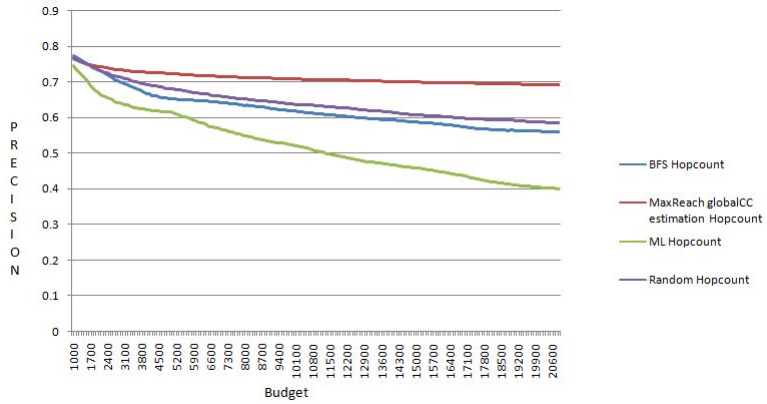

Figure 4.12: Precision for amazon dataset with cost of exploring node equal to its degree

The f1score is maximum for BFS approach for larger values of budget. Though it can be seen that machine learning approach gives better recall than BFS but overtime its F1-score starts to decrease. Also it can be observed that nodes having high estimated clustering coefficient give good precision and low recall even after several nodes have been explored. This is because few nodes are getting added to the network, as the degree of nodes with high clustering coefficients is

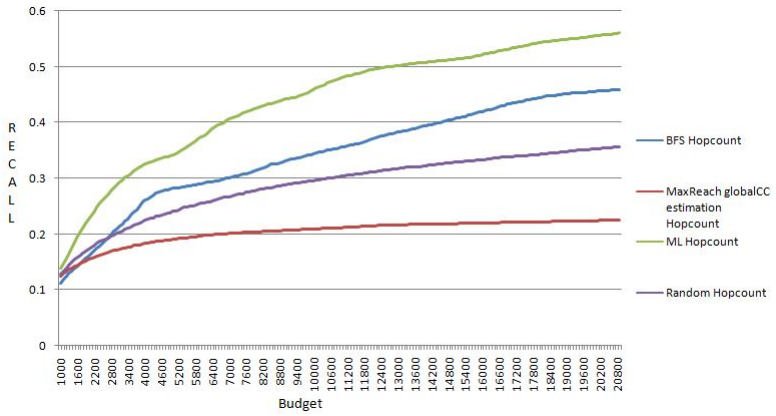

Figure 4.13: Recall for amazon dataset with cost of exploring node equal to its degree

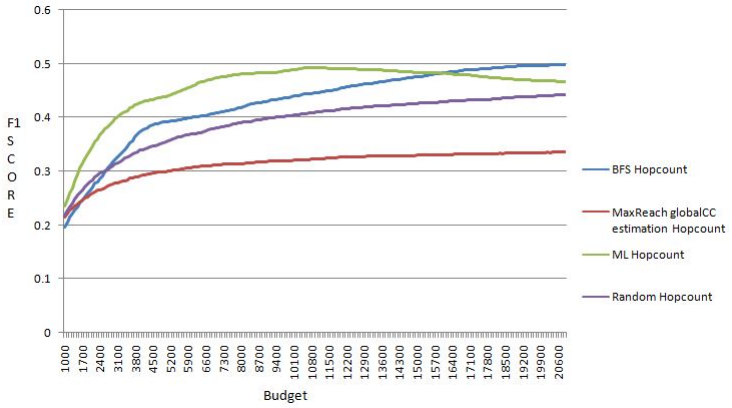

Figure 4.14: F1score for amazon dataset with cost of exploring node equal to its degree

usually low. Random approach selects a combination of nodes with high degree, high clustering coefficient and nodes near to the seed nodes. Its evaluation metrics are close to average of the other approaches. In the Amazon dataset, the seed nodes are away from higher degree nodes whereas in dblp dataset, the higher degree node is in close vicinity of the seed nodes. So the number of nodes that are brought into the network are less. This results in slow increase in the ratio of number of nodes discovered to the number of nodes in the network and machine learning is therefore able to better predict the nodes for exploration for a higher budget value also.

LFR Artificial dataset

Figure 4.15 to 4.17 show precision, recall and f1-score for amazon dataset. In figures 4.15 to 4.17 the cost of exploring a node is equal to 1. It can be observed that BFS performs better than ML approach in artificial dataset. This is because the number of seed nodes compared to the total number of nodes is far more than the ratio from the above datasets. As we explore the seed nodes before applying any approach, the number of nodes in the incomplete network is large in comparison to the other datasets. Though, initially for few node explorations ML performs better. So, it can be said that lower the value of known nodes to the total number of nodes in

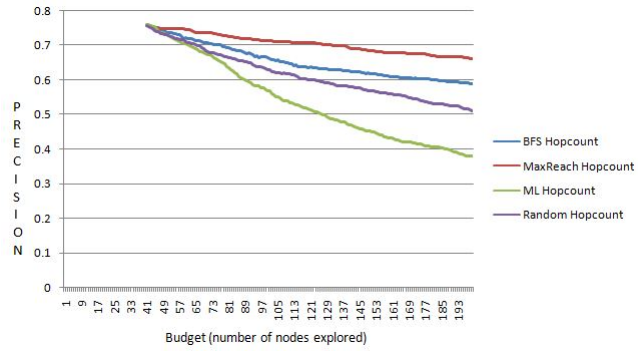

Figure 4.15: Precision for different network expansion techniques Artificial dataset

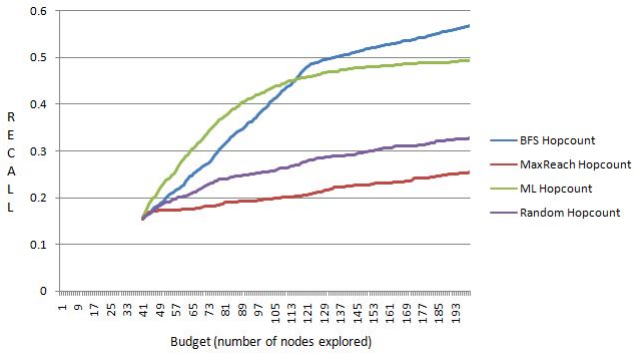

Figure 4.16: Recall for different network expansion techniques Artificial dataset

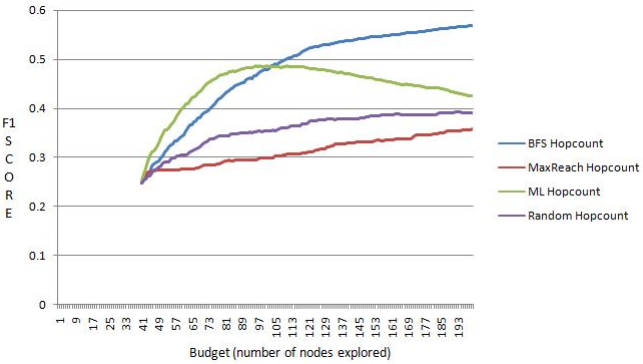

Figure 4.17: F1score for different network expansion techniques Artificial dataset

the network, better will be the performance of machine learning algorithm.

4.4.2 Detection technique used: Modularity

In this subsection, the results are for different network exploration technique with community detection technique as maximizing modularity.

DBLP

Figure 4.18 to 4.20 show precision, recall and f1-score for dblp dataset. In these figures the cost of exploring a node is equal to 1.

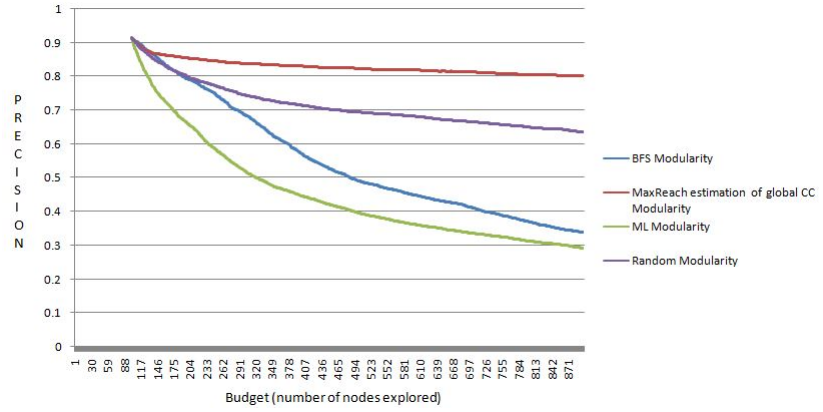

Figure 4.18: Precision for different network expansion techniques

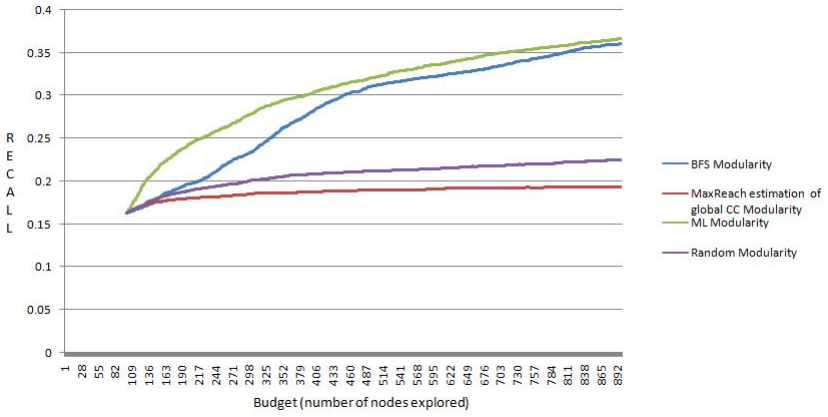

Figure 4.19: Recall for different network expansion techniques

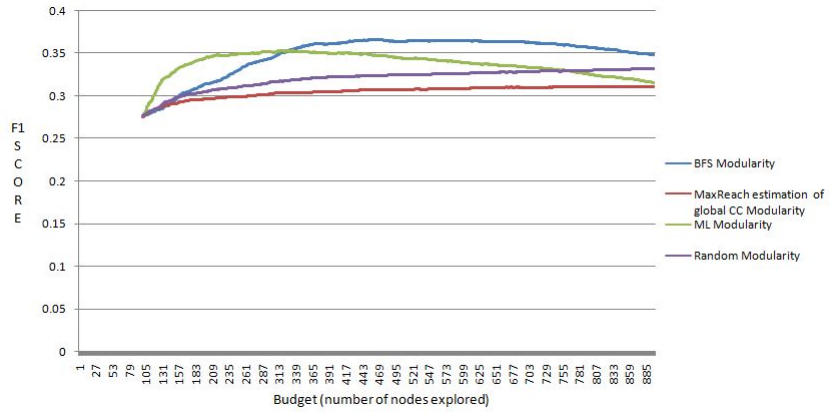

Figure 4.20: F1score for different network expansion techniques

Similar results were obtained when the cost of exploring the node was equal to its degree. The

trends for modularity as community detection are same as hopcount approach. Here also we observe that machine learning gives better results initially.

Amazon dataset

Figure 4.21 to 4.23 show precision, recall and f1-score for amazon dataset. In these figures the cost of exploring a node is equal to 1. Similar results were obtained when the cost of exploring the node was equal to its degree.

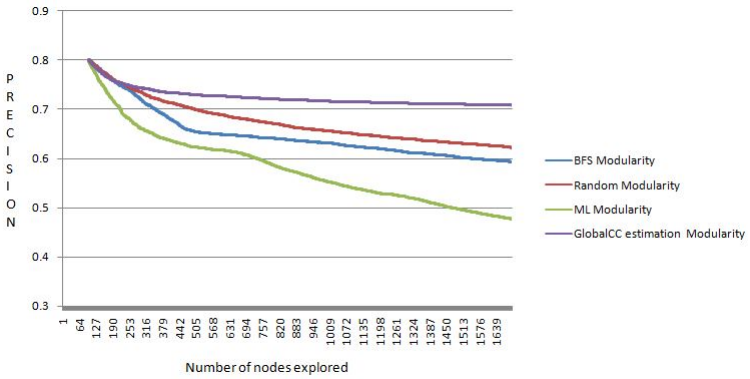

Figure 4.21: Precision for different network expansion techniques

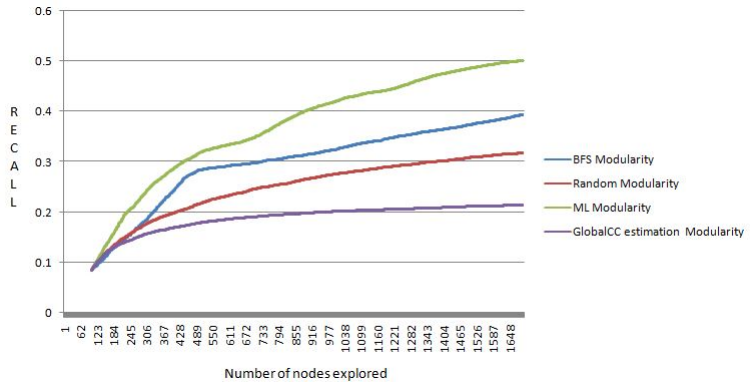

Figure 4.22: Recall for different network expansion techniques

Machine learning can be seen to give better recall value as compared to other network expansion techniques.

LFR Artificial dataset

The trends for LFR dataset with community detection algorithm as modularity maximization resemble the results obtained by using hopcount approach.

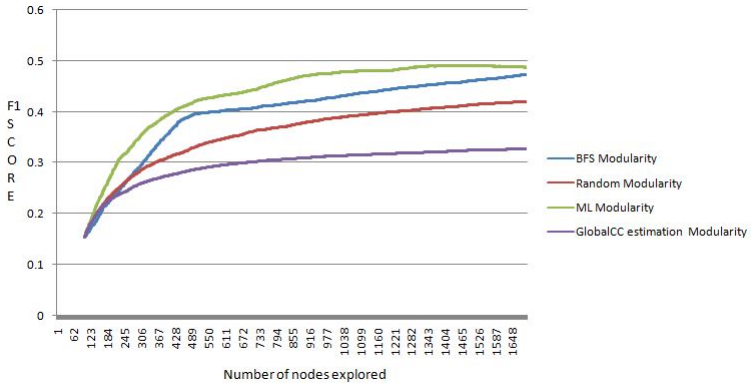

Figure 4.23: F1score for different network expansion techniques

4.4.3 Detection technique used: Permanence

In this subsection, the results are for different network exploration technique with community detection technique as maximizing permanence.

DBLP

Figure 4.24 to 4.26 show precision, recall and f1-score for dblp dataset. In these figures the cost of exploring a node is equal to 1.

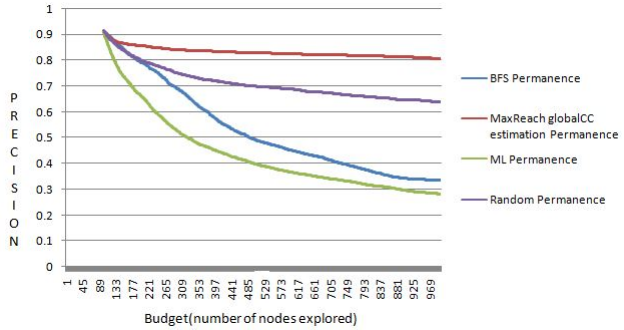

Figure 4.24: Precision for different network expansion techniques

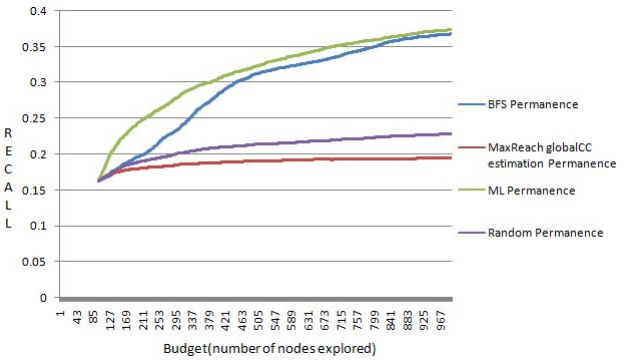

Figure 4.25: Recall for different network expansion techniques

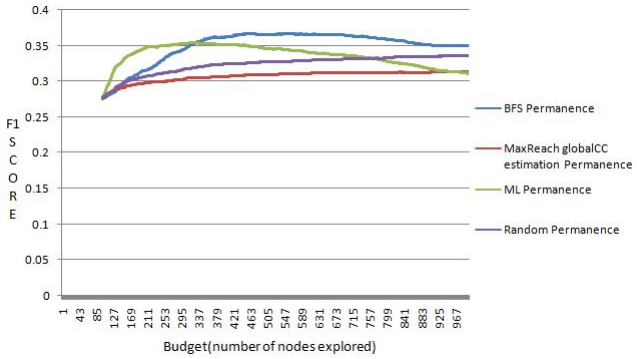

Figure 4.26: F1score for different network expansion techniques

The trends for permanence are identical to the trends observed for other community detection algorithms with machine learning performing better for small budgets and BFS surpassing it for large budgets.

Amazon dataset

Figure 4.27 to 4.29 show precision, recall and f1-score for amazon dataset. In these figures the cost of exploring a node is equal to 1.

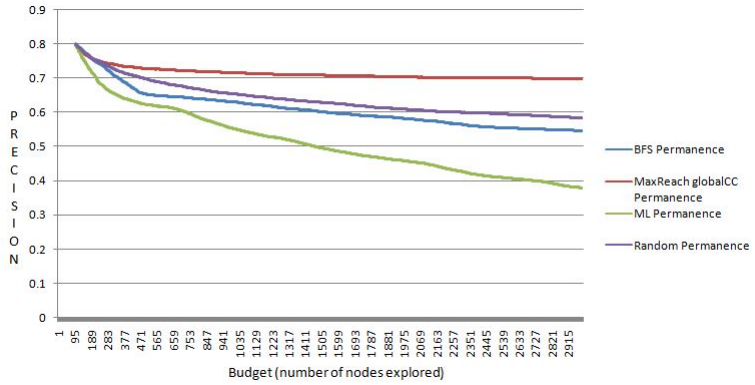

Figure 4.27: Precision for different network expansion techniques

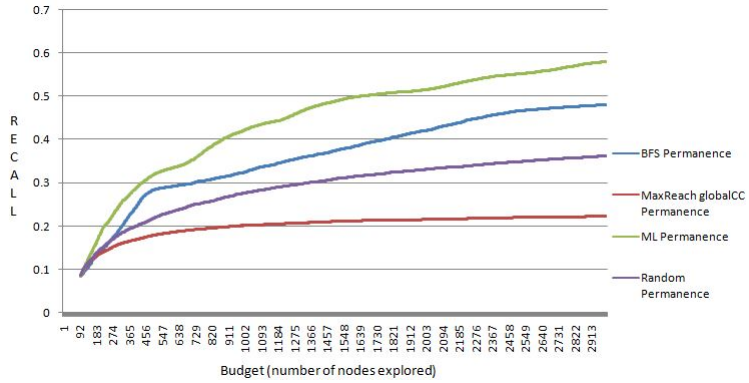

Figure 4.28: Recall for different network expansion techniques

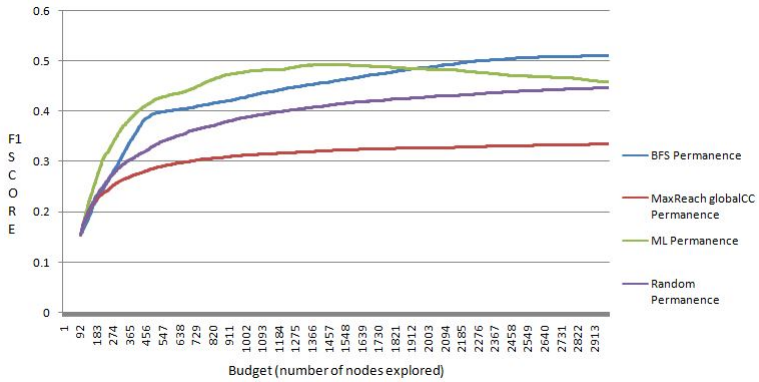

Figure 4.29: F1score for different network expansion techniques

LFR Artificial dataset

The trends for LFR dataset with community detection algorithm as permanence maximization resemble the results obtained by using other two approaches.

4.5 Comparison of community detection techniques

4.5.1 DBLP dataset

BFS

Figure 4.30 to 4.32 shows the comparison of the community detection algorithms with network expansion approach as BFS. Hopcount method gives slightly better recall and slightly lower

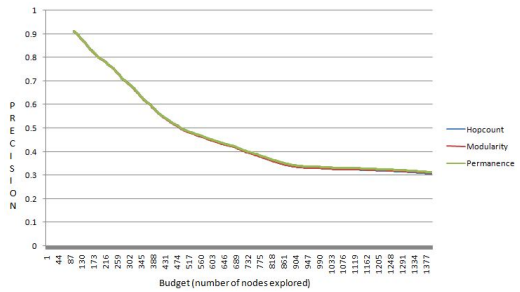

Figure 4.30: Precision for BFS

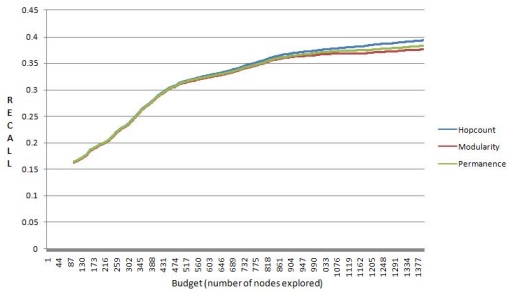

Figure 4.31: Recall for BFS

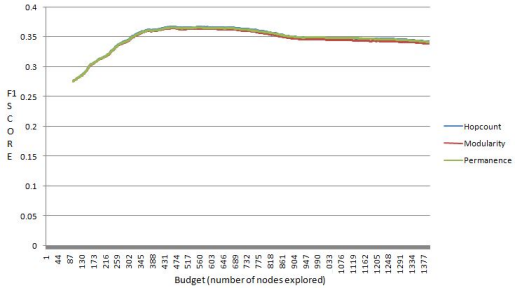

Figure 4.32: F1score for BFS

precision than the other methods.This is because each node is assigned to atleast some seed node community. In the modularity maximization and permanence maximization algorithms, there is a chance that some of the nodes are not assigned to any of the seed nodes community.

Random

Figure 4.33 to 4.35 shows the comparison of the community detection algorithms with network expansion approach as random. All three community detection algorithms with random approach

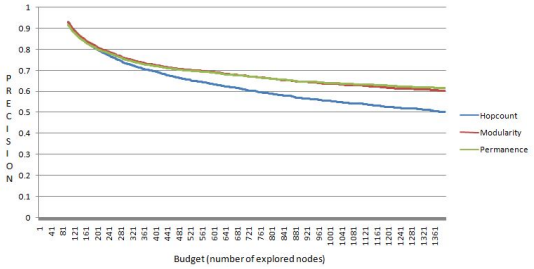

Figure 4.33: Precision for Random approach

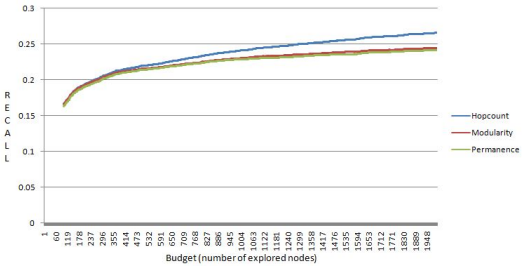

Figure 4.34: Recall for Random approach

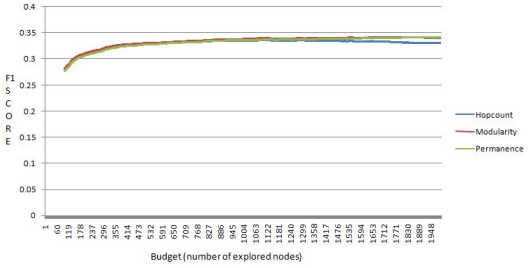

Figure 4.35: F1score for Random approach

follow the same trend as the BFS approach. However, the difference is quite evident in these trends compared to the BFS approach.

Clustering coefficient

Figure 4.36 to 4.38 shows the comparison of the community detection algorithms with network expansion approach as highest clustering coefficient. Hopcount method gives slightly better recall and lower precision than the other methods because each node is assigned to atleast some seed node community. In the modularity maximization and permanence maximization algorithms, there is a chance that some of the nodes are not assigned to any of the seed nodes community. This would increase the precision value.

ML

Figure 4.39 to 4.41 shows the comparison of the community detection algorithms with network expansion approach as ML classifier. The trends in ML approach are quite similar to the BFS approach.

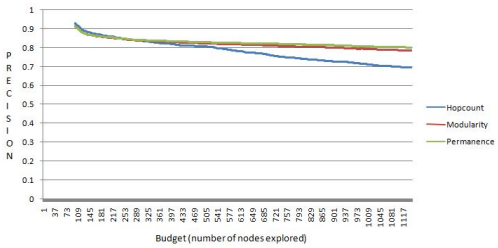

Figure 4.36: Precision for Clustering coefficient approach

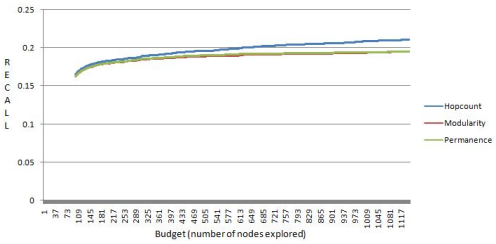

Figure 4.37: Recall for Clustering coefficient approach

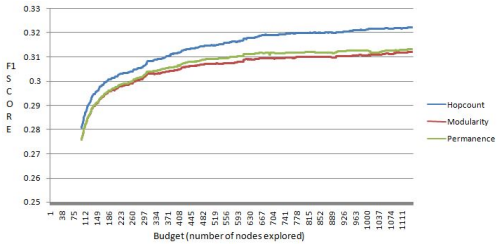

Figure 4.38: F1score for Clustering coefficient approach

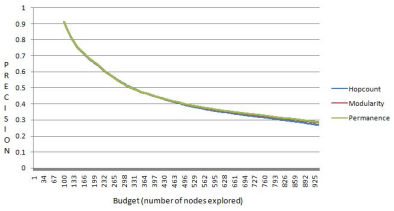

Figure 4.39: Precision for ML approach

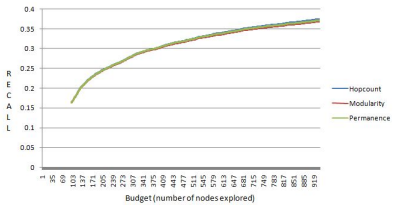

Figure 4.40: Recall for ML approach

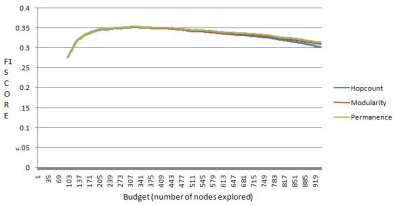

Figure 4.41: F1score for ML approach

4.5.2 Amazon dataset

Random

Figure 4.42 to 4.44 shows the comparison of community detection techniques when the network expansion is done by selecting the node randomly. Hopcount method gives slightly better recall and lower precision than the other methods because each node is assigned to atleast some seed node community. In the modularity maximization and permanence maximization algorithms, there is a chance that some of the nodes are not assigned to any of the seed nodes community.

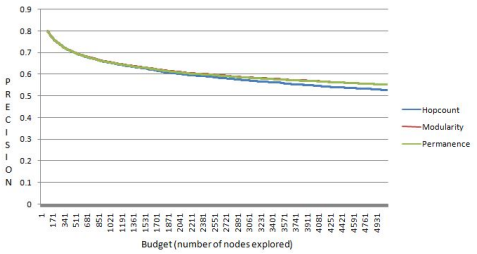

Figure 4.42: Precision for Random approach

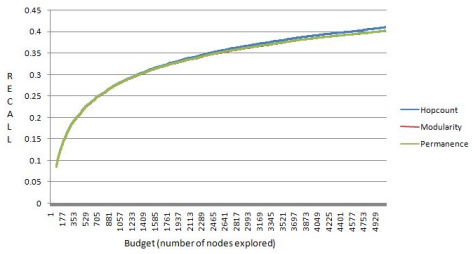

Figure 4.43: Recall for Random approach

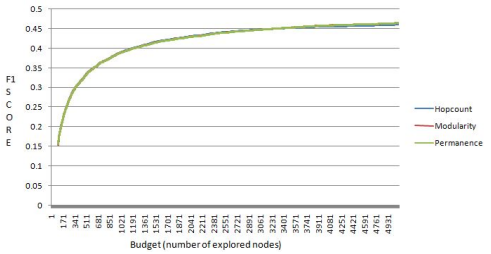

Figure 4.44: F1score for Random approach

Clustering coefficient

Figure 4.45 to 4.47 shows the comparison of community detection techniques when the network expansion is highest clustering coefficient approach. The trends are same as those for other

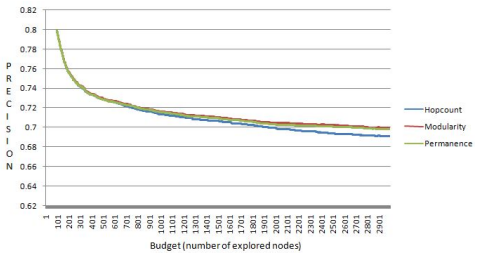

Figure 4.45: Precision for Clustering coefficient approach

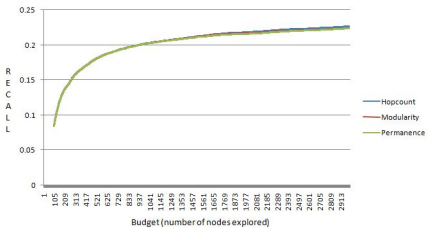

Figure 4.46: Recall for Clustering coefficient approach

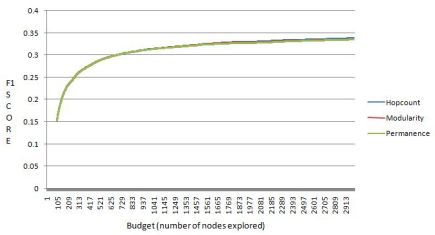

Figure 4.47: F1score for Clustering coefficient approach

network expansion approaches. Hopcount giving marginally high recall and permanence giving slightly high precision.

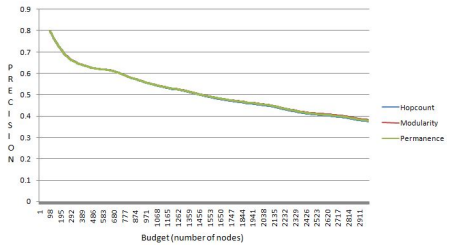

Figure 4.48: Precision for ML approach

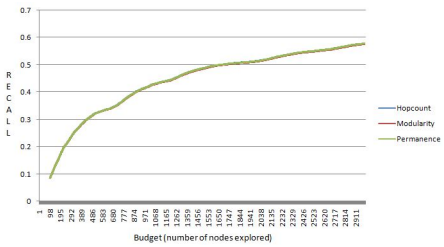

Figure 4.49: Recall for ML approach

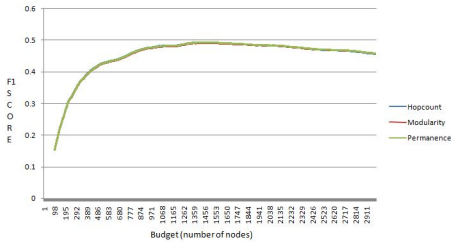

Figure 4.50: F1score for ML approach

ML

Figures 4.48 to 4.50 show results for comparison of community detection algorithm for ML approach. Community detection algorithms with ML approach as network expansion technique is giving results similar to otherapproaches.

4.6 Comparison of metric on varying the number of neighbors retrieved at a time

In this section, we show results for varying the number of neighbors that are retrieved at a time.

4.6.1 DBLP

Figures 4.51 to 4.53 show precision, recall and f1score for BFS as network expansion approach and hopcount as community detection approach. We observe that higher the value of k in k-neighbor probing, higher is the recall and f1score and lower is the precision. As more nodes are brought at the same time, there is higher probability that nodes within the desired community will be retrieved. This trend might not be constant as the nodes that are brought into the network are chosen randomly and it would also depend on the network.

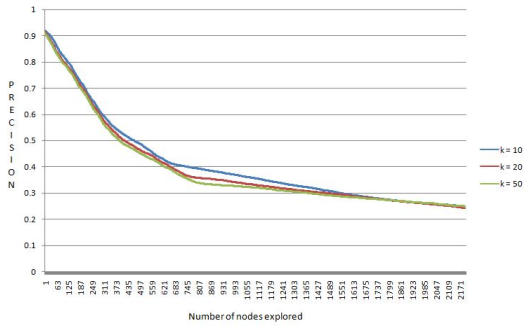

Figure 4.51: Precision for k-neighbor probing

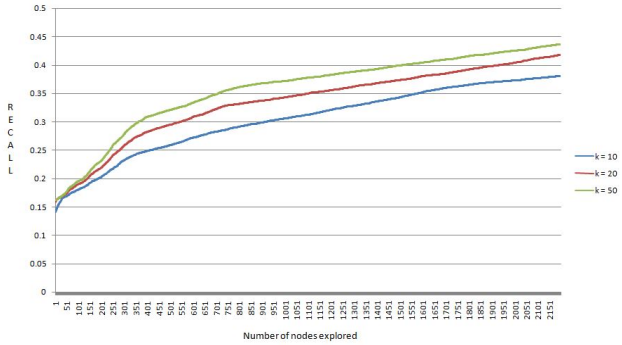

Figure 4.52: Recall for k-neighbor probing

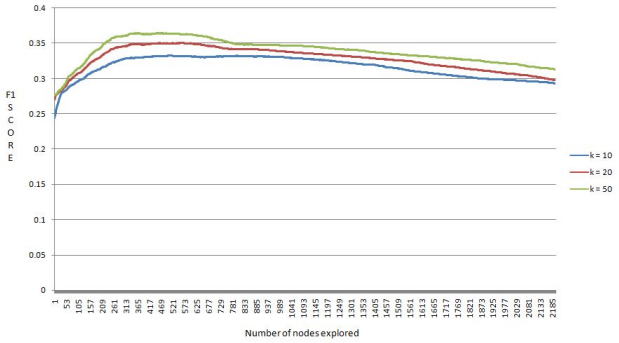

Figure 4.53: F1score for k-neighbor probing

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Machine learning approach was able to achieve a higher recall by exploring least number of nodes and bringing least number of nodes into the seed network. F1 score initially increases however after a point it starts to decrease. This is because initially the increase in the recall is large but later on the decrease in precision overpowers the slight increase observed in recall. We also observed that lesser the ratio of number of nodes in the incomplete network, better recall and f1-score can be obtained using machine learning approach till a higher budget. Once the ratio is beyond a certain threshold, we can observe that BFS approach gives better result. We may conclude that no single network expansion approach is able to maximize all three evaluation metrics simultaneously.

We could also observe that community detection algorithms (hopcount, maximizing modularity, maximizing permanence) are performing equally well with respect to precision and f1score on our seed networks. Hopcount methodology can be seen to get a slightly better recall. This is because every node that is brought into the seed network is assigned to the seed nodes community in hopcount approach, whereas in modularity and permanence based methods there is a likelihood that some of the nodes might form their own community. Permanence maximization method provides slightly better precision and f1score compared to the other methods.

5.2 Limitation and Future work :

Limitation : In network expansion and community detection techniques, seed network plays a critical role. The results are highly sensitive to the initial seed network. Also, the machine learning approach works for only small budget.

Future Work : Future work may include improving the classifier, machine learning approach works for larger budget. We can extend the application of machine learning approach for community detection in dynamic networks. We can train the model with community detection metrics as features.

Bibliography

[1] Fortunato, S. and Hric, D., 2016. Community detection in networks: A user guide. Physics Reports, 659, pp.1-44.

[2] Nguyen, N.P., Dinh, T.N., Shen, Y. and Thai, M.T., 2014. Dynamic social community detection and its applications. PloS one, 9(4), p.e91431.

[3] Chakraborty, T., Dalmia, A., Mukherjee, A. and Ganguly, N., 2017. Metrics for community analysis: A survey. ACM Computing Surveys (CSUR), 50(4), p.54.

[4] Soundarajan, S., Eliassi-Rad, T., Gallagher, B. and Pinar, A., 2016, August. MaxReach: Reducing network incompleteness through node probes. In Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on (pp. 152-157). IEEE.

[5] Bliss CA, Danforth CM, Dodds PS. Estimation of global network statistics from incomplete data. PLoS One. 2014;9(10) e108471. doi:10.1371/journal.pone.0108471. PMID: 25338183; PMCID: PMC4206292.

[6] Grando, F. and Lamb, L.C., 2015, July. Estimating complex networks centrality via neural networks and machine learning. In Neural Networks (IJCNN), 2015 International Joint Conference on (pp. 1-8). IEEE.

[7] Newman, M. E. J. & Girvan, M. (2004) Phys. Rev. E 69, 026113.

[8] Chakraborty, T., Srinivasan, S., Ganguly, N., Mukherjee, A. and Bhowmick, S., 2014, August. On the permanence of vertices in network communities. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1396-1405). ACM.

[9] Agarwal, P., Verma, R., Agarwal, A. and Chakraborty, T., 2018, June. DyPerm: Maximizing Permanence for Dynamic Community Detection. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 437-449). Springer, Cham.

[10] Blondel, V.D., Guillaume, J.L., Lambiotte, R. and Lefebvre, E., 2008. Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment, 2008(10), p.P10008.

[11] Andersen, R. and Lang, K.J., 2006, May. Communities from seed sets. In Proceedings of the 15th international conference on World Wide Web (pp. 223-232). ACM.

[12] Yang, J. and Leskovec, J., 2015. Defining and evaluating network communities based on ground-truth. Knowledge and Information Systems, 42(1), pp.181-213.

[13] Kim, M. and Leskovec, J., 2011, April. The network completion problem: Inferring missing nodes and edges in networks. In Proceedings of the 2011 SIAM International Conference on Data Mining (pp. 47-58). Society for Industrial and Applied Mathematics.

[14] Leskovec, J. and Mcauley, J.J., 2012. Learning to discover social circles in ego networks. In Advances in neural information processing systems (pp. 539-547).

[15] Ye, Z., Hu, S. and Yu, J., 2008. Adaptive clustering algorithm for community detection in complex networks. physical review E, 78(4), p.046115.

[16] Liu, J., Aggarwal, C. and Han, J., 2015, February. On integrating network and community discovery. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (pp. 117-126). ACM.

[17] https://snap.stanford.edu/data/