

Max-Node Sampling: an Expansion-Densification Algorithm for Data Collection

Katchaguy Areekijseeree, Ricky Laishram, Sucheta Soundarajan
Department of EECS, Syracuse University USA.
{ kareekij, rlaishra, susounda }@syr.edu

Problem and Motivation

The rise of online social networking sites in recent years has produced a gold mine of data.

- Researchers can understand the interesting behaviors and phenomena which happen in real world systems.

Before data can be analyzed, it must first be collected !

The social networking platforms provide a API to access their data. Unfortunately, the APIs come with a limitation.

For example,

- Twitter allows only 15 requests per 15 minutes for crawling following/follower relationships.
- LinkedIn allows around 1,000 requests for the same interval.

It takes almost six days to collect all the friends and followers of 8,000 unique users on Twitter.

How should one determine which nodes to query so that the resulting sample is optimal with respect to a desired goal?

Problem Definition

Use case scenario:

- Suppose that we want to obtain data from Twitter.
- We have 24 hours for collecting data.
- We expect to get as many Twitter users as possible.
- Start the data collection process by selecting one known user account.
- Query a list of following or follower users via Twitter API.
- The server returns a list of users. These users are stored in the list.
- Pick a user from the list for next query.
- Repeat the process until run out of time.

Problem Formulation:

Suppose there is a true, underlying undirected network $G(V, E)$, where V is set of nodes (*users*), E is a set of edges (*activities*). We assume that we have no information about G .

Given:

- A starting node (n_{start})
- A number of API requests (*budget*)

Goal:

- Collect a sample graph $S(V', E')$ where $V' \subseteq V$ and $E' \subseteq E$.
- Maximize $|V'|$ within a budget.

Overview: Max-Node

Intuition:

- Max-Node is based on the intuition that real networks exhibit community structure.
- The existing algorithm, like MOD algorithm, may get '*stuck*' in a dense cluster or community.

Max-Node thus consists of two phases:

- **Densification**, which queries nodes in the observed region to fill out that region.
- **Expansion**, which transitions the sampling algorithm to a new region of the graph.

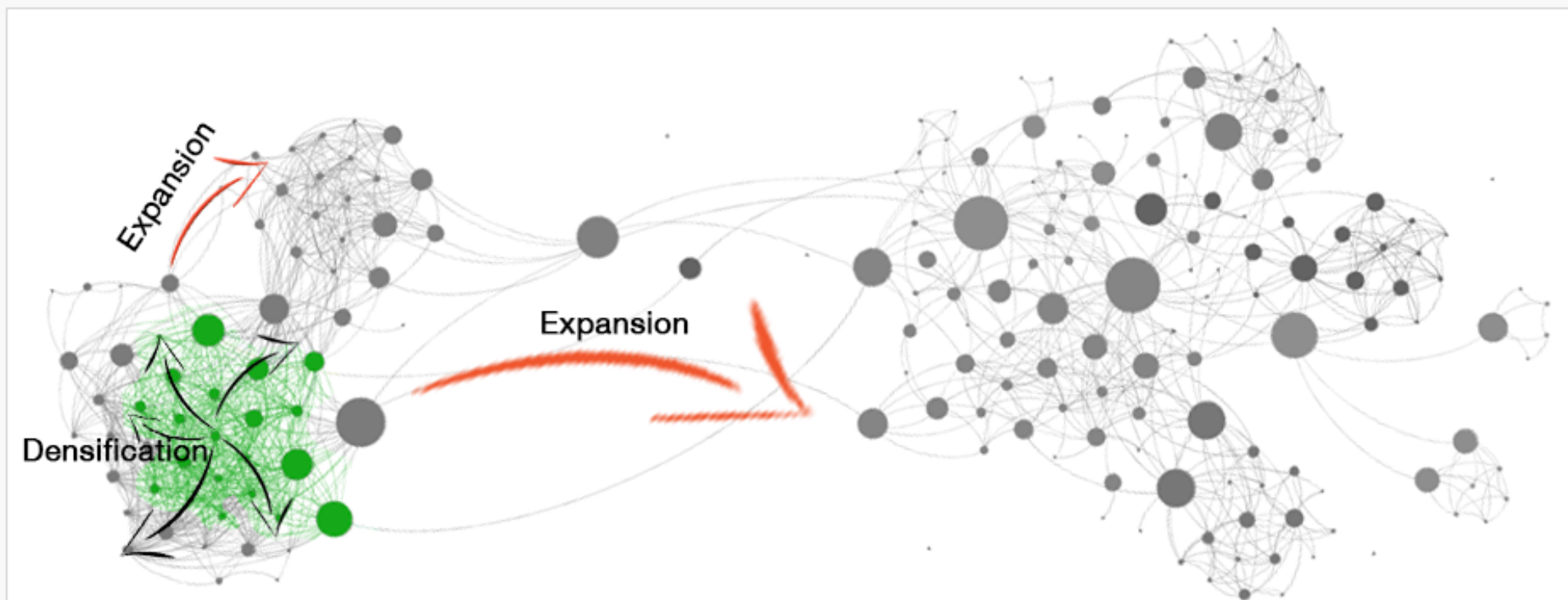


Figure 1. Concept of Expansion-Densification Sampling

Max-Node

Densification:

To expand a sample within a region, we adopt *Maximum Observed Degree* (MOD), as it outperforms other algorithms in the same class.

- In each iteration, the node with maximum degree is selected from S_s^o and the algorithm requests its neighbors through the API.
- Nodes(N_q) and edges(E_q) are returned and added to sub-sample (S_s).

Expansion:

The algorithm tries to escape from the current region of the network. The algorithm selects a node that will lead to another dense area.

- In the spirit of explore-exploit algorithms, one naive approach is to pick a node uniformly at random from S^o .
- In our future work, we examine other strategies for Expansion.

Switching Phases:

Intuitively, in each step, the number of closed nodes increases while a number of new nodes added decreases over time (diminishing marginal returns). Two scores are calculated in each iteration of Densification, sc_{den}^t and sc_{exp}^t . These scores give us an approximation of number of nodes left unexplored.

- When $sc_{exp}^t > sc_{den}^t$, the algorithm switches from Densification to Expansion phase.

sc_{den}^t measures how many new nodes are added to the sample after a request, divided by the number of closed nodes.

sc_{exp}^t is the fraction of the number of edges (e_{ij}) connecting a *closed node* to an *open node*, divided by the number of open nodes in sub-sample.

- If the number of edges (e'_{ij}) increases, the number of open nodes also increases.
- If not, it means the algorithm already found most of the nodes.

Note:

- A *closed node* is a node that has already been queried.
- An *open node* is a node that has been observed, but not queried.

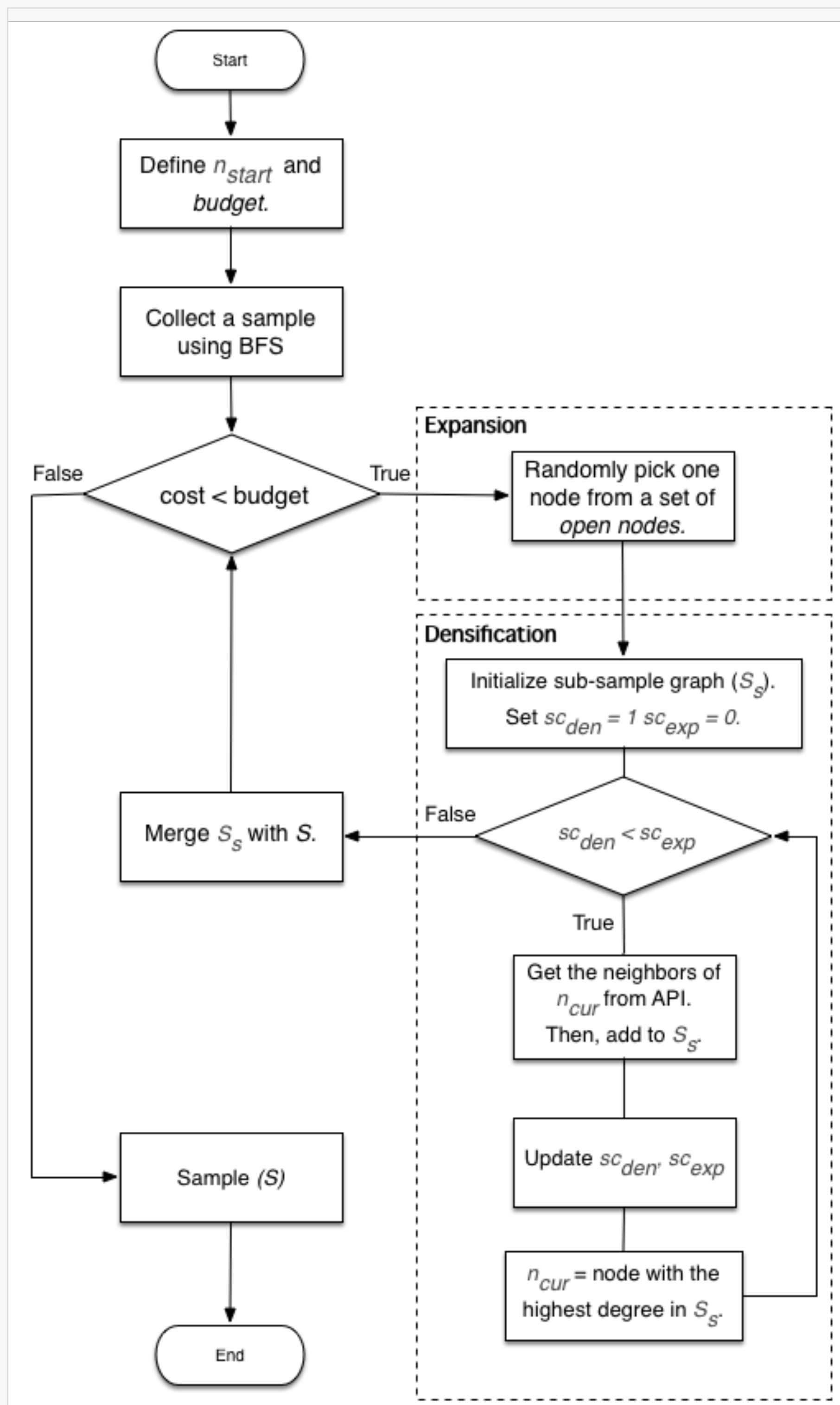


Figure 2. Pseudocode of the Max-node algorithm

Experiments

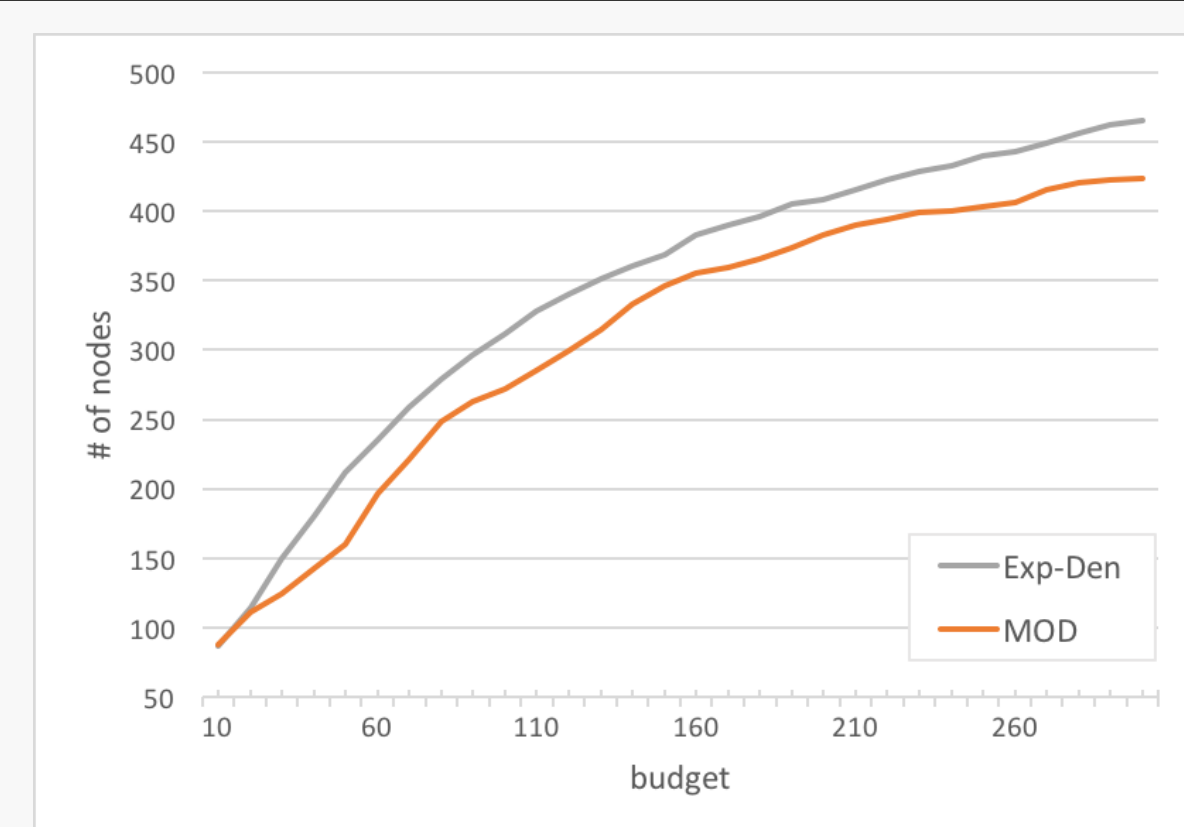
To mimic the process of querying an API, we simulate sampling from an existing network dataset. We use four different datasets, described in Table 2.

- *Grad* and *Undergrad* are the Facebook networks.
- *Enron-Email* is an email communication network.
- Twitter is a friend-follower network that we collected via Twitter API.

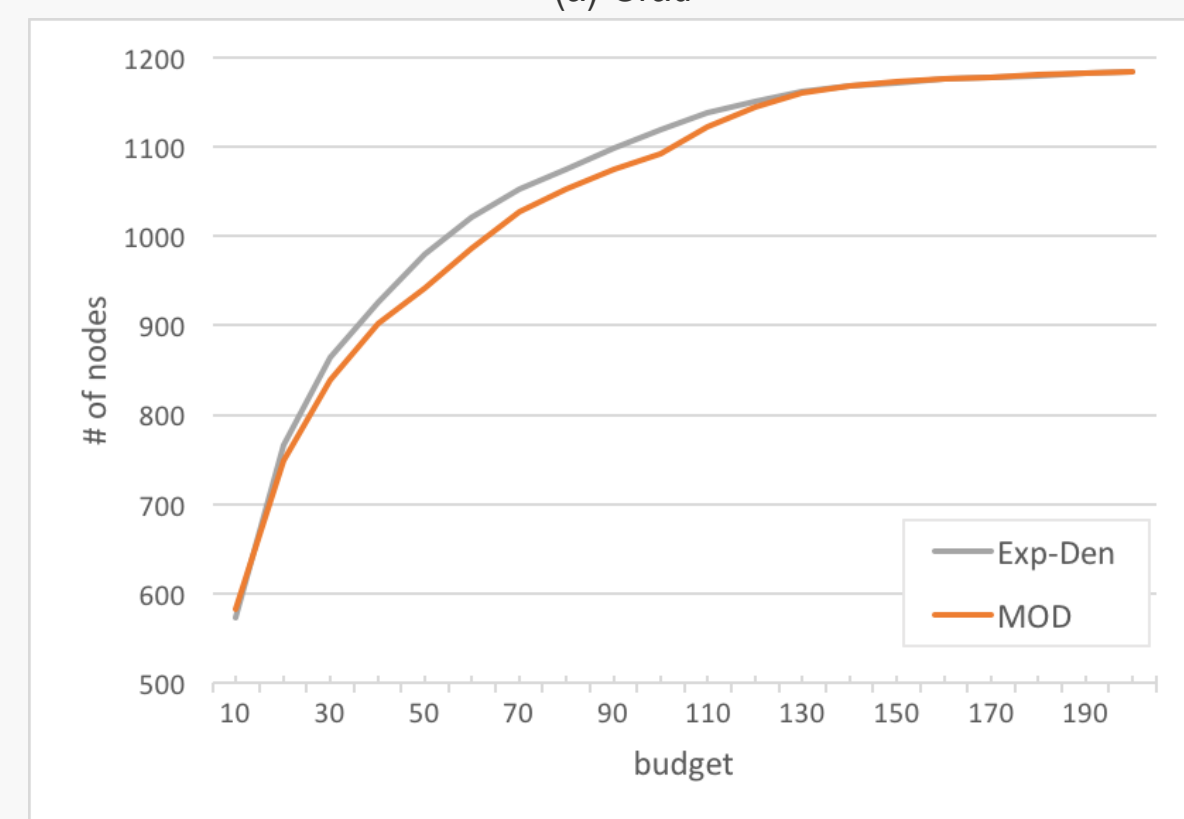
Datasets:

Network	# nodes	# edges	Global CC.	Modularity
Grad	503	3256	0.4792	0.6915
Undergrad	1220	43208	0.2980	0.3937
Twitter	12230	50884	0.1117	0.6371
Enron-Email	36692	183831	0.4970	0.5975

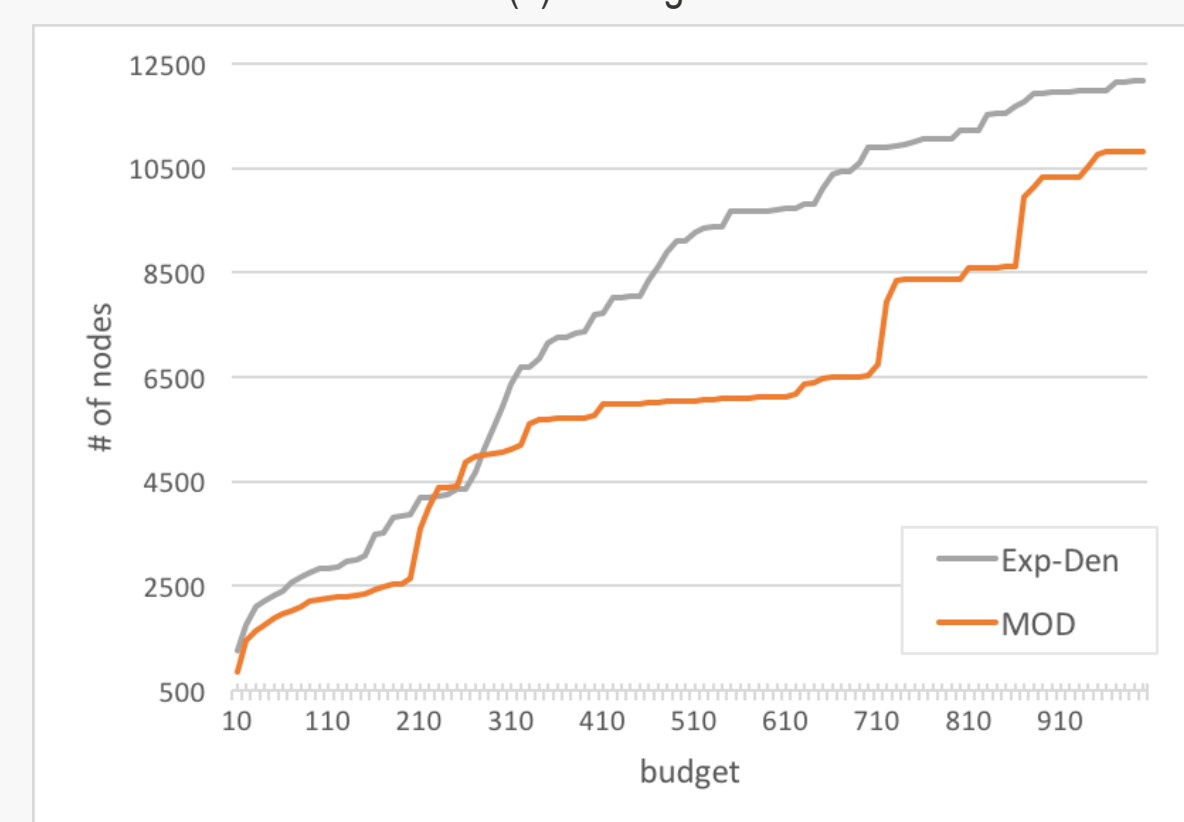
Results



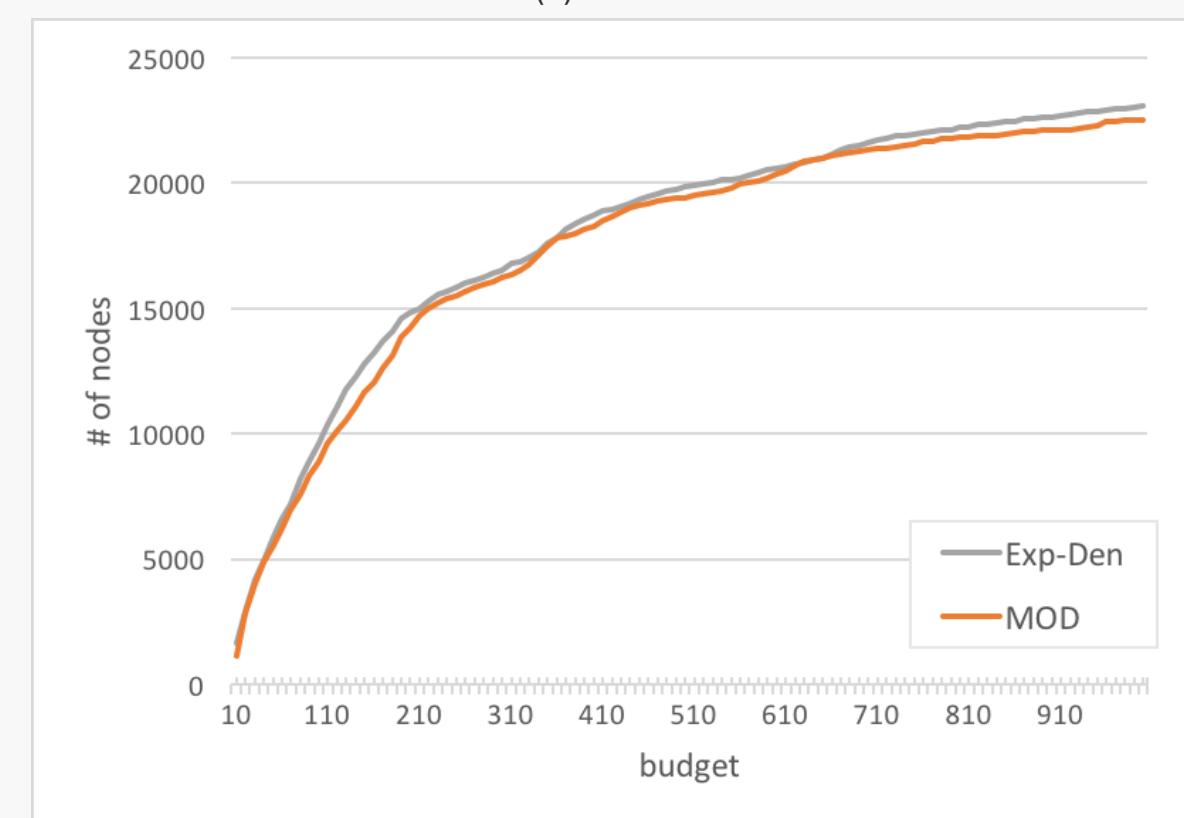
(a) Grad



(b) Undergrad



(c) Twitter



(d) Enron-Email

The results show an improvement of up to 40% vs. the baseline.

Conclusion

This gives us strong evidence that Max-Node algorithm is able to collect more nodes than MOD at the same amount of budget.

With a budget constraint, Max-Node performs well. Our future work includes improving the Expansion strategy with different switching criteria.

Reference

K. Areekijseeree, R. Laishram, and S. Soundarajan, "Max-Node Sampling: An Expansion-Densification Algorithm for Data Collection", IEEE BigData. 2016.