Comparing Landmark Selection Strategies for Fast Shortest Path Distance Estimation in Large Networks

Social Network Analysis for Computer Scientists — Course paper

Andrew Caruana
a.caruana@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

Van Nguyen
v.nguyen.2@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

ABSTRACT

This paper tackles the problem of exact shortest paths calculations in large networks being too taxing to compute in a reasonable manner of time. Our proposal uses landmarks, a small set of nodes whose distances to all other nodes are precomputed. These landmarks can then be used to approximate the shortest path between two nodes. This paper mainly deals with experimenting with different types of landmark selection techniques, and aims to build upon previous work by combining various landmark selection techniques, and checking their approximation quality amongst other methods. The results showed the these hybrid methods did not perform very well, and the overall best performers were measures based on betweenness centrality.

KEYWORDS

shortest path, betweenness centrality, social network analysis, landmark detection, network science

ACM Reference Format:
Andrew Caruana and Van Nguyen. 2023. Comparing Landmark Selection Strategies for Fast Shortest Path Distance Estimation in Large Networks: Social Network Analysis for Computer Scientists — Course paper. In *Proceedings of Social Network Analysis for Computer Scientists Course 2023 (SNACS '23).* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

In this information age, social networks have emerged as our main tool for communication, information sharing and especially social interaction. Platforms like Instagram, LinkedIn or X (formerly known as Twitter) have changed how people connect. These connections promote both strong relationships between others, but also bring about new difficulties. Relationships that can give each side new ideas or meaningful connections. This is the surface of social networks, at least. However, the structure behind them, and how they work so well, makes them an area of particular interest throughout multiple fields of academia. In the case of computer science, academics are particularly interested in the graphical structure of these networks, which gives rise to the field of social network analysis.

SNACS '23, Master CS, Fall 2023, Leiden, the Netherlands

Within social network analysis, one important area of exploration is the computation of a shortest path. In the real world, this could be finding the shortest route between two cities, or finding out how many mutual friends it takes to reach a particular person on a social network. In addition, the shortest path in a network is an important piece in calculating other measures relating to network analysis, such as betweenness centrality, or calculating the average distance across the entire graph.

At surface level, this task is not very difficult, and is well understood. However, when networks reach the size that social networks have nowadays, approximation techniques, such as the method proposed by Potamias et al, have to be used. [11] which will be discussed in further detail in Section 2. However, one critical concept of this paper is the concept of a landmark within a network. A landmark is a node within a network that can be used to estimate shortest paths within said network. Several landmarks would be. Instead of computing the shortest distances between each node, the shortest distances to the landmarks are computed, which are then used to approximate the distance between any two nodes in the network.

The main aim of this paper is to investigate how different landmark selection strategies fare well in approximating shortest distances within the network. The landmark selection strategies that are being considered are based on the work by Potamias et al. [11] along with the work done by Wang et al. [15], and aims to combine the various strategies used within these papers.

Most of these landmark selection strategies are based on various centrality concepts. Namely, degree, closeness, and betweenness centrality. All of these are various strategies for determining the importance of a node in various contexts. Nodes with a high degree centrality tend to be more important on the micro scale, whereas closeness centrality takes the average distance to all other nodes into consideration. Betweenness centrality, on the other hand, gives the number of shortest path that go through the given node. These nodes are then identified as bridges or connectors within the network, which makes them ideal for potential landmarks. However, calculating the betweenness centrality would again need to be approximated, due to the sheer scale of modern day large networks.

Various works have impplemented these centrality measures as landmark selection strategies independently [11, 14, 15]. However, none have combined various centrality measures into one landmark selection strategy, which is what we are interested in investigating

within this paper.

This hybrid method, as it shall be called, would be a combination of the other centralities. Since each centrality shows some value of importance within a node, it would be interesting to see if you could mix and match landmark selection strategies and merge them together. For example, if one had d landmarks, and $x + y = d$, then x landmarks would be selected based on one centrality measure and y landmarks based on another.

The methodology of evaluating these selection strategies would be based on accuracy, by comparing it to a sample of exact shortest paths that will be calculated.

Following this Introduction section, we shall discuss some related work done previously, followed by the Preliminaries section to introduce some necessary framework for our paper. Next, we shall discuss our approach to the problem and the datasets used in our work. Following this, we shall discuss our experimentation, the results yielded from it, and our interpretations of them. Lastly, we will conclude the paper by stating some limitations and some potential future work.

2 RELATED WORK

Dijkstra's paper [2] describes his algorithm of finding the shortest path between two nodes in graph considering weighted edges. This algorithm can be computed at complexity of $O(n^2)$ for general graphs and for sparse graphs can be reduced to $O(m + n\log(n))$. It is also important to highlight that this method provides the exact calculation, which can prove to be very computationally taxing especially when dealing with large networks.

The A* algorithm was proposed Hart et al. [5], which in addition to using Dijkstra's algorithm [2], they also proposed the usage of additional heuristic to improve Dijkstra's algorithm efficiency. The additional heuristic they proposed is the function denoted as $h(n)$ where n is a node in the graph. This function provides an optimistic estimate of the cost to reach the target node from the current node, along with the actual cost from the start node to the current node $g(n)$, and also estimates $h(n)$ determines what nodes to explore. Similarly to Dijkstra's algorithm, the implementation of the A* algorithm on large networks can also take quite a while to compute.

Work done by Potiamas et al. [11] describes the landmark process mentioned previously, and is the main inspiration for this paper. The authors utilise a few landmark selection techniques, namely, random selection, degree centrality based selection and closeness centrality based selection.

Brandes et al. [1] dive deeper into landmark (also known as a pivot) selection strategies. Unlike from the paper from Potamias et. al. [11], this paper focuses more on the the methods within closeness and betweenness centrality such as Max Sum Strategy. This strategy considers the sum of distances as an indicator of how badly covered a vertex is by the current set of pivots. The next pivot is selected to maximize the sum of distances. This strategy computes the sum of distances and indicates how badly the network is covered by current pivots. The pivots are basically chosen to maximize the sum of distances.

Wang et al. [15] also wrote a similar paper to work done by Brandes et al.[1]. This paper describes methods for computing and discovering paths from nodes to another. They also propose a landmark based framework to optimize distance computation, and utilise centrality as one of the landmark selection criteria. This is as NP-hard problem but they propose a heuristic distributed strategy to ensure the approximation ratio. Paper concludes with confirming their hypothesis about effectiveness of betweenness based landmark selection with their added heuristic method.

Lastly, Takes et al. [14] aimed is to create an adaptive landmark selection technique in order to select central nods, along with ensuring that the landmarks are spread out across the network. In addition to closeness and degree centrality, the authors also make use of betweenness centrality and the PageRank algorithm [9] as landmark selection strategies. The authors found that the performance of their method to outperform those based on centrality.

As can be seen, lots of work in this topic already exists, however, a lot of these centrality measures have not been exhaustively compared. Furthermore, no previous work has been found where a combination of landmark selection strategies were used at once.

3 PRELIMINARIES

In this section we will discuss some necessary mathematical framework for our approach, along with discussing the centrality measures we shall be using and their definitions.

3.1 Framework

Let us consider an unweighted, undirected graph $G(V, E)$ that contains n nodes and m edges. Let $u, v \in V$, the shortest path between these nodes shall be denoted as $d(u, v)$.

Regarding landmarks, let us consider a small set of nodes D, which contains all of the landmarks for the graph. The distances from all landmarks to every node is then pre-computed, and stored in hash table of vectors. Each key contains a tuple of two nodes $[l, v]$ where l is a landmark and v is a regular node. Each value will then be the distance between l and v.

As also done by Potamias et al. [11], we will also be using the upper bounds for distance estimation. To give a brief overview:

$$\max_i |u_i - v_i| \leq d(u, v) \leq \min_j \{u_j + v_j\}$$

The left-hand-side of the inequality is known as the lower bound, and the right-hand-side, the upper bound. The value in the middle ($d(u, v)$ is the actual distance between u and v. As previously stated, we are mainly interested in the upper bound. This is due experimentation done by Potamias et al. [11] where they find that for the centrality measures, the upper bound estimation performs better than the lower bound one. This value is calculated by going

through each landmark, and summing the distance from the current landmark to u and to v. The smallest value found between these additions indicate what landmark to estimate the distance for those particular nodes.

3.2 Centrality Measures

Furthermore, let $deg(u)$ denote the degree of a node, that is, how many nodes are adjacent to u. Degree centrality can then be defined as:

$$C_d(u) = \frac{\deg(u)}{n-1}$$

This centrality measure has a constant time complexity, making it very cheap to compute.

Next, closeness centrality is based on the average distance from a given node to any other node. Computing it can be done in linear time for one node, and with a time complexity of $O(mn)$ for all n nodes in the network. This can be formally defined as:

$$C_c(u) = \left(\frac{1}{n-1} \sum_{v \in N} d(u,v) \right)^{-1}$$

Betweenness centrality is defined as the number of shortest paths that run through a node and has a time complexity of $O(2mn)$, as it has to compute two breadth first searches for each node. This can be expressed as:

$$C_b(u) = \sum_{\substack{v,w \in N \\ v \neq w, u \neq v, u \neq w}} \frac{\sigma_u(v,w)}{\sigma(v,w)}$$

Here, $\sigma(v,w)$ is the number of shortest paths from v to w, and $\sigma_u(v,w)$ would be the number of these shortest paths that go through u.

Intuitively, one would expect betweenness centrality to perform the best as a landmark selection criterion out of these centrality measures, since its definition is based on how many shortest paths run through a node.

4 APPROACH

The first step in our approach is to replicate the work done by Potamias et al [11]. This includes all their previous landmark selection strategies (random, degree, closeness and border), along with implementing partitioning and distance constraints. Everything regarding these measures is being kept the same, in order to ensure that our results are comparable to those obtained in their paper.

That being said, we will also be using two additional landmark selection strategies. These being approximated betweenness centrality, and hybrid centrality. The approximate betweenness centrality is calculated by selecting a random sample of seed nodes from the graph, performing several BFSs, and keeping track of which nodes are dependent on others during this process. These dependencies are then normalised by dividing them by the number of seed nodes selected. The pseudocode for this algorithm can be seen in Algorithm 1.

Data: graph, num_seeds=20
Result: approx_betweenness_centrality

Input:
seed_nodes ← `random_sample(`*graph.nodes(), num_seeds*`)` ;
approx_betweenness_centrality ←
`create_hash_table(`*initial_value=0.0, keys=graph.nodes()*`)` ;

Algorithm:
for *each seed_node in seed_nodes* **do**
  paths ← `single_source_shortest_path(`*graph, source=seed_node*`)` ;
  dependencies ← `create_hash_table(`*initial_value=0, keys=graph.nodes()*`)` ;
  for *each path in paths* **do**
    for *each node in path[1:-1]* **do**
      dependencies[node] += 1 ;
    end
  end
  for *each node in graph.nodes()* **do**
    if *node ≠ seed_node* **then**
      approx_betweenness_centrality[node] +=
       dependencies[node] ;
    end
  end
end

Normalisation:
for *each node in graph.nodes()* **do**
  approx_betweenness_centrality[node] ÷= num_seeds ;
end

Output:
return *approx_betweenness_centrality* ;

Algorithm 1: Approximate Betweenness Centrality

With regards to hybrid centrality, we will experiment with different combinations of the other centrality measures. These being an even three way split between degree, closeness and betweenness centrality, along with a 50/50 split between each pair of centralities. The intuition behind this being that different combinations might vary in effectiveness depending on the graph. However, it is expected that the even three way split will be the most consistent.

Following the implementation of these landmark selection strategies, we will evaluate their performance based on accuracy. To perform this evaluation, we shall compare our approximated distances with their exact values for a select amount of nodes in the graph and determine the average error.

5 DATA

Five datasets are being used for this paper, all of which being social network graphs. These being the Twitch Gamers Social Network (TG) [13], DBLP [18], Facebook Pages (FP) [12], Catser Friends (CF) [7] and Hyves [7, 19]. Some statistics about these datasets can be seen in Table 1.

The first three datasets were obtained from SNAP [8] and the last two from KONECT [7]. In addition, all of these graphs are undirected, unweighted and do not contain multiple edges between the same nodes. It should also be noted that for Catster, we will be using the largest weakly connected component for our experiments, as the graph is disconnected.

Twitch Gamers: This dataset is a graph of Twitch users which was collected from the public API in 2018. Each node represents a user, and edges are mutual follower relationships between said users [13].

DBLP: A co-authorship network between computer science publications. Each node represents an author of a paper, and nodes form an edge if they publish at least one paper together [18].

Facebook Pages: This data represents the relationships of verified Facebook pages. Each node is a page, and an edge is created between two nodes if they have mutual likes among them. This dataset is also split into 8 categories, however, only the artist category is being considered as it is the largest one [12].

Catster: This dataset contains a graph showing user friendship on the pet magazine site Catster. Each node represents a user and edge edge, a friendship between users [7].

Hyves: Hyves was a Dutch social network site in the early 2000s. This dataset is similar to others where it represents friendship between users of the site. Each node represents a user, and each edge a friendship between users [7, 19].

Dataset	Nodes	Edges	t_{BFS} (seconds)	Density
TG	168,114	6,797,557	2.47	0.00048
DBLP	317,080	1,049,866	7.65	0.00002
FB	50,515	819,306	0.33	0.00064
Catster	148,826	5,448,486	1.25	0.00049
Hyves	1,402,673	2,777,419	26.1	0.000003

Table 1: Dataset Statistics

6 EXPERIMENTS

In this section we shall discuss the specifics of our implementation, followed by the experiments performed and elaborate on our results.

Our approach was implemented using Python 3.10.0. The hardware used had an average clock speed of around 3GHz and 16GB of RAM. We also utilised several Python libraries, including NetworkX [3], Numpy [4], Pandas [10, 17], Matplotlib[6] and Metis [16]

The first experiment was calculating the approximation error for each landmark selection method for each dataset. In order to have actual values to compare the approximation to, 500 random nodes pairs were selected and their exact shortest paths were calculated. For each node pair, this was done using the formula:

$$\frac{|\hat{l} - l|}{l}$$

Where $\hat{l}$ represents the approximated shortest path, and l represents the exact value.

After this being calculated for each node pair, the average was taken and recorded. This was done for 24 and 102 total landmarks, these specific numbers were chosen as they are both divisible by 2 and 3, which is important for hybrid landmark selection to have even distributions between centrality measures. This is due to different combinations of centralities being experimented on. It should also be noted that the seed for the random node pair selection was made constant (seed: 69) such that the node pairs would remain the same between experiments. The results can be seen below in Tables 1 and 2.

24 Landmarks	DBLP	TG	FB	Catster	Hyves
Random	0.551	0.659	0.554	0.665	0.594
Degree	0.285	0.274	0.250	0.272	0.210
Closeness	0.294	0.275	0.249	0.272	0.211
Betweenness	0.262	**0.244**	**0.223**	**0.247**	0.207
Degree/1	0.281	0.282	0.259	0.298	0.211
Closeness/1	0.286	0.287	0.305	0.317	0.210
Betweenness/1	**0.250**	0.270	0.262	0.308	**0.209**
Degree/P	0.360	0.272	0.256	0.257	0.251
Closeness/P	0.584	0.288	0.372	0.275	0.401
Betweenness/P	0.376	0.274	0.262	0.261	0.260
Border/P	0.296	0.273	0.255	0.256	0.227
Hybrid/DCB	0.308	0.282	0.264	0.284	0.211
Hybrid/DC	0.281	0.279	0.254	0.294	0.210
Hybrid/CB	0.312	0.279	0.260	0.275	0.212
Hybrid/DB	0.311	0.279	0.255	0.262	0.212

Table 2: 24 Landmarks Average Approximation Error

Within the tables, /1 indicates that the distance constraint strategy is being applied, and no neighbours of D are being considered as landmarks. /P indicates that the partitioning strategy is being applied, with a landmark being selected from each partition. Lastly, the different combinations of letters after Hybrid indicate which centrality measures were used in that hybrid. For example, Hybrid/DB being a 50/50 combination of degree and betweenness centrality.

In both Tables 1 and 2, it can be seen that, as hypothesised, betweenness and betweenness/1 were the only measures to get the best results. It is also interesting to notice that in 4 out of the 5 datasets, the best landmark selection strategy remained the same,

102 Landmarks	DBLP	TG	FB	Catster	Hyves
Random	0.492	0.580	0.421	0.644	0.500
Degree	0.258	0.248	0.226	0.249	0.208
Closeness	0.263	0.256	0.228	0.250	0.209
Betweenness	0.264	**0.247**	**0.227**	**0.247**	**0.207**
Degree/1	0.256	0.267	0.236	0.296	0.210
Closeness/1	0.259	0.278	0.266	0.312	0.208
Betweenness/1	**0.251**	0.273	0.238	0.308	0.209
Degree/P	0.324	0.255	0.234	0.251	0.304
Closeness/P	0.448	0.386	0.307	0.263	0.378
Betweenness/P	0.386	0.263	0.255	0.263	0.230
Border/P	0.263	0.256	0.228	0.252	0.211
Hybrid/DCB	0.260	0.265	0.240	0.251	**0.207**
Hybrid/DC	0.259	0.253	0.232	0.254	**0.207**
Hybrid/CB	0.269	0.265	0.230	0.256	0.209
Hybrid/DB	0.264	0.251	0.232	0.257	**0.207**

Table 3: 102 Landmarks Average Approximation Error

which may indicate that the amount of landmarks does not play a role in finding out which landmark selection strategies are best. Furthermore, the accuracy did not improve much, if at all between the 24 landmark and 102 landmark experiments.

The hybrid landmark selection methods did not perform very well in most cases. This could be because whilst one centrality measure might perform well, the other ones, generally speaking, tend to not perform well. Thus, leading the hybrid measure to also be dragged down. That being said, hybrid also was not the worst performer in most cases. However, if one is going through the effort of using certain landmark selection methods, it seems to be likely that it might not be worth the effort to compute various hybrid computations.

Regarding the other landmark selection techniques, the partitioning methods besides Border/P also did not perform very well, however all centrality based measures outperformed random landmark selection. In addition, the results we have achieved are quite different from those found in the paper by Potamias et al. [11] where they found that closeness centrality was quite a good performer. In almost all cases degree outperformed closeness, however, this could be attributed to the fact that we are using different datasets, as it is already known that the graphs one uses very much influence the optimal landmark selection strategy [11, 14].

It is also worth noting that we took note of the time it takes to compute the upper bound estimation, and while the time was still quite low, the exact shortest path calculations for the 500 nodes were still lower at under a second. Following this, we ran the test with 10,000 nodes and the same result happened. This could simply be because these numbers of nodes are still a low enough sample that exact methods outperform approximations.

That being said, the Hyves dataset had some interesting results, all of the centrality measures besides degree/P, closeness/P and betweenness/P performed extremely well, to the point where three out of the four hybrid measures were tied for the best accuracy. This could be due to Hyves being an extremely sparse graph compared to the others somehow, however that is just an observation.

7 CONCLUSION

The results of the paper were quite mixed, betweenness centrality proved to be an overall high performing landmark selection strategy, as expected. However, the hybrid centrality measures did not perform very well and in reality are likely not worth computing when one is already computing the base centrality measures, one of which is likely to perform better than the hybrids.

One potential way this work could be expanded upon by implementing it on much larger datasets in order to fully get a grasp as to how much more efficient these landmark selection strategies are than the exact methods. Furthermore, one could experiment with further alternative methods of landmark selection, such as one based on eccentricity centrality.

ACKNOWLEDGMENTS

We would like to thank Dr. Frank Takes and all of the teaching assistants for the SNACS course who helped us and provided meaningful feedback throughout the creation of this paper.

REFERENCES

[1] Ulrik Brandes and Christian Pich. 2007. Centrality Estimation in Large Networks. *International Journal of Bifurcation and Chaos* 17, 07 (2007), 2303–2318.
[2] Edsger W Dijkstra. 2022. A Note on Two Problems in Connexion with Graphs. In *Edsger Wybe Dijkstra: His Life, Work, and Legacy*. 287–290.
[3] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, Gaël Varoquaux, Travis Vaught, and Jarrod Millman (Eds.). Pasadena, CA USA, 11 – 15.
[4] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (Sept. 2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2
[5] Peter E Hart, Nils J Nilsson, and Bertram Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE transactions on Systems Science and Cybernetics* 4, 2 (1968), 100–107.
[6] J. D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, 3 (2007), 90–95. https://doi.org/10.1109/MCSE.2007.55
[7] Jérôme Kunegis. 2013. KONECT – The Koblenz Network Collection. In *Proceedings of the International Conference on World Wide Web Companion (WWW Companion)*. 1343–1350. http://konect.cc/
[8] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.
[9] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. *The PageRank Citation Ranking: Bring Order to the Web.* Technical Report. Technical report, stanford University.
[10] The pandas development team. 2020. *pandas-dev/pandas: Pandas.* https://doi.org/10.5281/zenodo.3509134
[11] Michalis Potamias, Francesco Bonchi, Carlos Castillo, and Aristides Gionis. 2009. Fast Shortest Path Distance Estimation in Large Networks. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 867–876.
[12] Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. 2019. GEMSEC: Graph Embedding with Self Clustering. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2019*. ACM, 65–72.
[13] Benedek Rozemberczki and Rik Sarkar. 2021. Twitch Gamers: a Dataset for Evaluating Proximity Preserving and Structural Role-based Node Embeddings. arXiv:2101.03091 [cs.SI]

[14] Frank W Takes and Walter A Kosters. 2014. Adaptive Landmark Selection Strategies for Fast Shortest Path Computation in Large Real-World Graphs. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Vol. 1. IEEE, 27–34.

[15] Qing Wang, Shengyi Ji, Peng Peng, Mingdao Li, Ping Huang, and Zheng Qin. 2020. Optimizing Distance Computation in Distributed Graph Systems. *IEEE Access* 8 (2020), 191673–191682.

[16] Ken Watford. 2012. *METIS for Python.* https://metis.readthedocs.io/en/latest/ Accessed on 17/12/2023.

[17] Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman (Eds.). 56 – 61. https://doi.org/10.25080/Majora-92bf1922-00a

[18] Jie Yang and Jure Leskovec. 2012. Defining and Evaluating Network Communities based on Ground-truth. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*.

[19] Reza Zafarani and Huan Liu. 2009. Social Computing Data Repository at ASU. http://socialcomputing.asu.edu.