Network Lens: Node Classification in Topologically Heterogeneous Networks

Kshiteesh Hegde 1  Malik Magdon-Ismail 1

Abstract

We study the problem of identifying different behaviors occurring in different parts of a large heterogeneous network. We zoom in to the network using lenses of different sizes to capture the local structure of the network. These network signatures are then weighted to provide a set of predicted labels for every node. We achieve a peak accuracy of \(\sim 42\%\) (random=11\%) on two networks with \(\sim 100,000\) and \(\sim 1,000,000\) nodes each. Further, we perform better than random even when the given node is connected to up to 5 different types of networks. Finally, we perform this analysis on homogeneous networks and show that highly structured networks have high homogeneity.

1. Introduction

Large networks, which are a direct result of the ever-expanding big data world, are commonplace in almost every domain. Social networks such as Facebook and Twitter, customer purchase data on e-commerce platforms like amazon.com, author citation records like DBLP, road networks in any country/state are all popular examples of large networks. In a heterogeneous setting, one could be dealing with the situation where parts of a network are behaving differently. For example, in a social network, some parts may behave collaboratively, some may be terrorist-like, etc. Conventional graph classification approaches will fail to notice such differences in a network. Identifying the behavior of different parts of a network is a key problem. One of the applications of solving this node classification problem is clustering. We illustrate this in Figure 1.

We start with a network that potentially is exhibiting different behaviors in different parts. The model takes in subgraphs of different sizes using random walks depending on the lens size from various parts of the network. We call it the lens since one can do a random walk starting from any node in the network to capture local structure in that region much similar to how one can hover a metaphorical lens on any area of the network to zoom in and see that area in more detail.

Then, a set of labels with associated probabilities (confidence of the model) is outputted. We can visualize this in the colored graph in Figure 1 which illustrates the output on a toy example. The model has identified three different behaviors in the network: a star graph, a wheel graph and a ladder graph denoted by cyan (S), yellow (W) and green (L) respectively. Thus, our node classification model has clustered the given graph into three differently behaving parts. Clustering is just one of the potential applications of our node classification lens.

We study the general problem: Given a large, potentially heterogeneous, network, can one identify the different behaviors occurring in the network? The outline of our ap-
The network signatures shown in Figure 2 as introduced in (Wu et al., 2016) are the structured adjacency matrices of subgraphs picked up by lenses of different sizes. In the image, a black pixel at position \((i, j)\) denotes an edge between nodes \(i\) and \(j\). It is structured according to the ordering scheme presented in (Wu et al., 2016) and is presented in more detail in Section 3.3. They are a powerful representation (Hegade et al., 2018) of networks since they are agnostic to the type of the network. They can be applied to a wide variety of networks including, but not limited to, social, information, transportation and even terrorist networks. One of the applications of this representation is subnetwork classification. They are good for machine learning algorithms and have an intuitive visual representation.

A signature of a network comes primarily from its function. For example, the function of a road network is transportation. The functions of a transportation network include having the ability to connect local places in a city as well as distant cities via highways. It also needs to manage traffic during rush hours. It need not have connections from every place to every other place but the it has a large cut. The networks that evolve to support a transportation function are grid-like. That is the signature of a road network (Figure 2d). The connections in a network evolve in a particular way to support a particular function and develop a signature. Similarly, different networks with different functions evolve in different ways and develop different signatures.

In this work, we use the signatures to build a network lens. We use models previously trained on these signatures (see Section 3) from the homogenous setting to obtain a set of predicted labels from each of the lenses for each node. We then use linear programming to arrive at the optimal weights for labels of different lenses and construct a final set of predicted labels for each node.

To solve the general problem of node classification via network lens, we need new ways to:

1. Classify a small subnetwork of a network into one of several types
2. Test the accuracy of such algorithms
3. Evaluate performance on real networks

**Our Contributions**

1. We transform the problem of graph classification in to one of image classification. We show that even at a tiny local scale of up to 8 nodes we can classify nodes in a heterogenous network with \(~ 1,000,000\) nodes and achieve accuracies up to 42% which is well above random performance of 11%.

**Figure 2.** Signatures of select networks to demonstrate the structured image embedding feature.

**Figure 3.** Workflow of Network Lens. Image embeddings of local adjacency matrices are the input to a classifier, which produces node labels.
2. We note that when a node is more diverse (having multiple connections to different types of networks), it is harder to predict that node’s type correctly. However, our lossless image feature is powerful enough that even when a node is connected to up to 5 different types of networks, we perform better than random. In the real world, where a node with two types of connections is the most common scenario, our technique is significantly better than random with $\sim$32% accuracy.

3. Finally, we test our model on real networks to study their degree of heterogeneity. We find that some networks are highly homogenous whereas others have a high degree of heterogeneity.

2. Related Work

We study the problem of identifying different behaviors in a network by using image classification to categorize local structure.

The idea of using the image embedding of the adjacency matrix as a feature was first introduced in (Wu et al., 2016). Based on this idea, authors in (Hegde et al., 2018) showed with great success that parent networks of tiny subgraphs (as small as 8 nodes) can be identified. They also used Caffe (Jia et al., 2014) to show that the structured image embedding features can be used for classification in a transfer learning setting. In this work, we use the idea to create a lens that can be used on heterogeneous networks to see the different behaviors exhibited in different parts of a network.

The most popular approaches to graph classification are feature selection and kernel methods. Authors in (Kong & Yu, 2010) perform semi-supervised feature selection by searching for optimal subgraph features. They define a metric that governs how features are selected. There is also the idea of using pattern recognition along with feature selection where the idea is that graphs from the same class should have similar attributes (Li et al., 2012). Spatial distribution of subgraphs is used as features in (Fei & Huan, 2008). In a similar vein, (Jin et al., 2009) introduces a pattern exploration scheme that looks for co-occurring features in subgraphs to perform binary classification. It is unclear how multi-class classification can be achieved (if at all) using this approach. In (Kong & Philip, 2010), the authors talk about extracting important features in a multi-label setting. They assume that the given data is already labeled (multiple times) and the task is to choose the correct label from the set. All the above methods require construction of features that are dependent on the given data. This can be non-trivial in cases where one has to deal with a diverse set of data as is the case in this study. Developing a one size fits all kind of a set of features is near impossible. In case of pattern recognition, if a new pattern or set of patterns emerge only in the test set, then the chances of catching them drastically decreases.

Many graph kernels based on walks, subtrees, cycles, shortest paths etc. have been proposed (Borgwardt & Kriegel, 2005; Gärtner et al., 2003; Kashima & Inokuchi, 2002; Kashima et al., 2003; Kudo et al., 2004; Riesen & Bunke, 2009). The kernel function computes the similarity between two graphs and then a classifier such as SVM is used for classification. As evidenced by the abundance of different types of kernel functions, it is difficult to come up with a kernel that ticks all the boxes for a given classification problem. The size and domain of the network, complexity of the kernel function all affect the decision of choosing the right kernel. So, kernel methods are also affected by the same problems as feature selection methods.

All of the above mentioned literature assume a friendly setting where one network contains only one type of network. They are of little use when different types of subgraphs are connected to each other in the same network. This amounts to different parts behaving differently. This setting is much more difficult than the friendly setting as we demonstrate later.

The lossless structured image embedding feature used in this work, solves the above problems. It focuses on the structure that networks exhibit at a local level independent of the domain of origin of the network. As we show later, this approach works even when different classes of subgraphs are connected to each other in the same network. Since, we could not find similar methods introduced by previous researchers, we believe this is a significant result in the field of heterogenous node classification.

3. Data and Methodology

3.1. Data

We used 9 real world networks to construct our heterogeneous network. The networks are from a diverse set of domains like e-commerce, social, web, roads etc. Table 1 provides the number of nodes and edges in each of the individual networks.

3.2. Construction of Heterogenous Networks

Our first task is to construct a heterogeneous testbed using real networks. Each of the real world networks in Table 1 behaves differently and has a different individual signature (Hegde et al., 2018). We take several snapshots of each of these networks and splice them together to obtain one big heterogeneous network. This ensures that different regions of the spliced network possess different local signatures.

First, we extract 100 subgraphs with 8, 16, 32 and 64 nodes...
from each of the networks shown in Table 1 yielding 3600 (100 × 4 × 9) disjoint subgraphs in total. Next, we choose a pair of subgraphs at random and choose a node from each of these two subgraphs randomly and introduce an edge between them. This edge-introduction process is repeated 10 × 3600 times resulting in a connected heterogeneous network with 108,000 nodes and 294,841 edges. We constructed a bigger heterogeneous network similarly, but with 1000 subgraphs resulting in a network with 1,080,000 nodes and 2,951,234 edges. This process in illustrated in Figure 4.

3.3. Graph Image Embeddings

We briefly describe the process of converting adjacency matrices to lossless image features (Wu et al., 2016) here. The adjacency matrices can be visualized as images by simply treating 1s as black pixels and 0s as white pixels. However, the same adjacency matrix can be mapped to different images by permuting the rows. Using the image from a random permutation of the rows as input directly to a classifier such as a Convolutional Neural Network (CNN) results in very poor results. It is necessary to first re-order the nodes in a canonized form. We use the ordering scheme shown in (Wu et al., 2016), to make sure that all permutations of a given adjacency matrix map to the same structured image making it permutation invariant. When these structured images are fed to a CNN, classification performance is significantly improved. Neural networks show tremendous accuracy when it comes to recognizing real world images (Jia et al., 2014). As shown in (Hegde et al., 2018), they do very well with homogenous networks as well. Different subgraphs from the same network are different at the microscopic level but are similar on a macroscopic level. We use the image embeddings of local subgraphs to identify the different parts of heterogeneous graphs. Figure 5 is a visualization of the process.

3.4. Pre-trained Models

In (Hegde et al., 2018), authors test several classifiers with the task of discriminating between the real world networks mentioned in Table 1. CNN performs best with about 86% accuracy. The model is trained on the subgraphs extracted from the homogenous networks separately and it learns each of the individual network’s signature. We use the CNN model trained in this setting here. However, in our current setting the test data consists of snapshots taken from the heterogeneous network constructed as described in Section 3.2. We use the model trained in the homoge-
4. Experiments and Results

4.1. Node Classification in Heterogenous Networks

We perform random walks starting from every node in the network. Then, we obtain the structured image embedding of each subgraph. This is fed as a test sample to the already trained CNN model (on the real world networks) to get a label. We assign this label to the starting node of the random walk and all the other nodes in the test subgraph. However, we maintain these two sets of labels for each node separately. One set has the label a node receives when it is the starting node in the random walk and the second set contains all the labels it receives when it is not. Finally, we repeat the process with random walk lengths (lenses) of 8, 16, 32 and 64. Thus, each node gets 8 sets of labels. We show the individual accuracies of each of the lenses in Table 2. One can see that lens sizes 16 and 32 perform better than the smallest lens (8) and the largest lens (64). This is because the smallest lens zooms in too much into the network and the local signature is not captured optimally. Similarly, with the biggest lens, it looks at more than one local network signature in one snapshot which causes error. Each node gets a set of labels from different lenses. Rather than just using one of the lenses as the final classifier, we can aggregate all the labels to get a more accurate classification that incorporates the information from all the lenses. To this end, we split the nodes into training and test sets and use the training set to learn the optimal weights to weigh the label sets from each of the lenses. Consider the matrix $X_m \in \mathbb{R}^{8 \times 9}$ that is maintained for each of the $M$ nodes where the $ij$th entry denotes the number of times lens $i$ gave the label $j$ to node $m$.

Table 2. Performances (percent correct) of different lenses. First column: predicted label is assigned to all nodes in the test subgraph. Second column: predicted label is assigned only to the starting node of the test subgraph.

| Lens Sizes | Label assigned to all nodes | Label assigned to starting node only |
|------------|-----------------------------|-------------------------------------|
| 8          | 27.55                       | 29.08                               |
| 16         | 32.94                       | 33.96                               |
| 32         | 33.42                       | 34.53                               |
| 64         | 30.64                       | 30.65                               |

Now, we assign weights $p_i$ to the lenses such that the weighted sum of the column corresponding to the correct label is maximum. Let $y_m$ denote this column for node $x_m$. This condition can be written as:

$$1 \cdot y_m^T \cdot p \geq X_m^T \cdot p$$

To allow for error, we introduce slack variables $\xi_m$ and require $1 \cdot y_m^T \cdot p \geq X_m^T \cdot p - \xi_m$. The objective is to minimize the sum of errors which gives a linear program:

$$\text{minimize: } \sum_m \xi_m$$
$$\text{subject to: } (1 \cdot y_m^T - X_m^T) \cdot p \geq -\xi_m,$$
$$\sum_i p_i = 1,$$
$$0 \leq p_i \leq 1,$$
$$\xi_m \geq 0$$

Alternatively, one can naively assign the accuracy score of individual lenses as shown in Table 2 as the weights to the corresponding lenses. By applying the weights, we obtain a probability distribution over the 9 possible labels for each node. We classify the node as the top-$k$ labels, where $k$ is
4.1. Diversity of Nodes’ Network Connections

We repeat the experiments in Section 4.1 but with each of the individual networks that were spliced together to form the big heterogeneous network. The purpose of these experiments is to study how homogenous real networks are, for example, how much of the ‘Facebook’ network, actually behaves like a ‘Facebook’ network.

First, we look at Table 3 which shows test performance on each of the networks. The networks with high accuracy like Road Network, Facebook and DBLP have high homogeneity. We also show the mode of the incorrect labels for each network. This is the label that was assigned to a network in a setting where all the subgraphs are of the same size, the previously mentioned scenario can cause error.

In Figure 8, we observe that networks like Amazon and Wikipedia do have some heterogeneity compared to, say, Facebook and the road network. This is because, the labels with the top weight for a node in the Amazon network only have about 78.05% accuracy compared to near 100% accuracy with Facebook. This shows that the Amazon network has less inherent structure where as Facebook has high inherent structure and hence is more homogenous.
The number of different types of networks a node is connected to is referred to as its node diversity.

\[
\begin{array}{cccc}
\text{Node Diversity} & \text{Top Label Correct (%)} & \text{Average Top Weight} & \text{Average Entropy of Weights (normalized)} \\
1 & 50.39 & 0.53 & 0.57 \\
2 & 32.69 & 0.47 & 0.63 \\
3 & 19.85 & 0.44 & 0.67 \\
4 & 13.00 & 0.42 & 0.69 \\
5 & 12.30 & 0.41 & 0.70 \\
\end{array}
\]

Correlation with Node Diversity [-1, 1]  
-0.1867  +0.2466

**Figure 7.** Node diversity - as captured by our Network Lens.

In Table 5 we present the raw numbers behind the homogeneity analysis. Table 5 must be read row-wise. Entry \( j \) in row \( i \) represents the total reward the nodes of network \( i \) received for being type \( j \). When \( i = j \), the reward represents correct classification. Note that this is not a confusion matrix, but can be thought of as a confusion “row”. This is because there is only one class in each of the test sets since networks are tested one at a time. Also, every row sums to a different number since each network is of a different size resulting in different sized test sets (20%). The optimal weights are learned from the training set and their performance on the test set is shown in Figure 8.

**5. Conclusion and Future Work**

In summary, we successfully used a new way to classify a small subnetwork of a topologically heterogeneous network using structured image embeddings. We showed that this technique is highly scalable since we achieved high accuracies on a million node network. We believe our simple and easy to understand model coupled with its strong performance will pave the way for new research and applications. A future direction is to increase the database of pre-trained classifiers to improve the diversity.

**Acknowledgment**

This research was supported by the Army Research Laboratory under Cooperative Agreement W911NF-09-2-0053 (the ARL-NSCTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation here on.
Table 4. Number of different labels assigned to each type of node in the million node heterogeneous network by all the lenses

|        | Amazon | Terrorist Net. | Citation | DBLP | Facebook | Gowalla | Road Net. | Web | Wikipedia |
|--------|--------|----------------|----------|------|----------|---------|-----------|-----|-----------|
| Amazon | 17377.5| 20.5           | 1885.5   | 2026.5| 1034.3   | 196.5   | 444.5     | 816 | 98.5      |
| Terrorist Net. | 8423   | 7107.5         | 935      | 4752  | 1151.5   | 1080    | 115.5     | 366.5| 10        |
| Citation | 9424.5 | 15.5           | 9283.5   | 1737  | 1892.5   | 378.5   | 469.5     | 43  | 10.5      |
| DBLP   | 9999   | 42.2           | 710.5    | 1530  | 16451    | 1080    | 115.5     | 366.5| 10.5      |
| Facebook | 4398   | 6.5            | 710.5    | 1530  | 16451    | 1080    | 115.5     | 366.5| 10.5      |
| Gowalla | 9622.3 | 83             | 2163.5   | 3394  | 4201     | 128     | 2497.5    | 706.5| 72        |
| Road Net. | 9517   | 16.5           | 1776.7   | 1843  | 1707     | 280     | 7854      | 777  | 43        |
| Web    | 7517.8 | 7              | 809      | 2208.5| 779.5    | 292.5   | 706.5     | 777  | 43        |
| Wikipedia | 9615.3 | 15             | 5342.5   | 1105  | 987      | 1588.5  | 495.5     | 530.5| 4188.5    |

Table 5. Number of different labels assigned to the nodes of each of the homogeneous networks by all the lenses

|        | Amazon | Terrorist Net. | Citation | DBLP | Facebook | Gowalla | Road Net. | Web | Wikipedia |
|--------|--------|----------------|----------|------|----------|---------|-----------|-----|-----------|
| Amazon | 51262  | 199            | 4349.5   | 1400 | 112.5    | 2291    | 2121      | 2250| 2987      |
| Terrorist Net. | 0      | 41             | 0        | 0    | 2        | 11      | 0         | 0   | 0         |
| Citation | 1162   | 1.5            | 4871     | 96.5 | 82.5     | 390     | 12        | 47.5| 246       |
| DBLP   | 2368.5 | 59             | 966      | 58900.5| 45.5     | 430     | 4         | 581.5| 61        |
| Facebook | 2      | 1              | 1        | 1    | 801      | 1       | 0         | 1   | 0         |
| Gowalla | 3392   | 4.5            | 4697     | 629  | 142      | 27835.3 | 256.2     | 1728.2| 633.8     |
| Road Net. | 173    | 3              | 320      | 73   | 0        | 0       | 0         | 216996| 3         |
| Web    | 12109  | 453            | 4774.5   | 7775 | 3179.5   | 443     | 646       | 140672.5| 5090.5    |
| Wikipedia | 24     | 1              | 226      | 0.5  | 0.5      | 4.5     | 0         | 8   | 656.5     |

References

Borgwardt, K. M. and Kriegel, H.-P. Shortest-path kernels on graphs. In Data Mining, Fifth IEEE International Conference on, pp. 8–pp. IEEE, 2005.

Cho, E., Myers, S. A., and Leskovec, J. Friendship and mobility: user movement in location-based social networks. KDD, 2011.

Fei, H. and Huan, J. Structure feature selection for graph classification. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, pp. 991–1000, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3. doi: 10.1145/1458082.1458212. URL http://doi.acm.org/10.1145/1458082.1458212.

Gärtner, T., Flach, P., and Wrobel, S. On graph kernels: Hardness results and efficient alternatives. In Learning Theory and Kernel Machines, pp. 129–143. Springer, 2003.

Gehrke, J., Ginsparg, P., and Kleinberg, J. Overview of the 2003 KDD cup. SIGKDD Explor. Newsl., 2003.

Gehde, K., Magdon-Ismail, M., Ramanathan, R., and Thapa, B. Network signatures from image representation of adjacency matrices: Deep/transfer learning for subgraph classification. arXiv:1804.06275, 2018.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, pp. 675–678. ACM, 2014.

Jin, N., Young, C., and Wang, W. Graph classification based on pattern co-occurrence. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09, pp. 573–582, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3.

JJATT. http://doitapps.jjay.cuny.edu/jjatt/data.php, 2009.

Kashima, H. and Inokuchi, A. Kernels for graph classification. International Workshop on Active Mining, 2002.

Kashima, H., Tsuda, K., and Inokuchi, A. Marginalized kernels between labeled graphs. ICML, 2003.

Kong, X. and Philip, S. Y. Multi-label feature selection for graph classification. In Data Mining (ICDM), 2010 IEEE 10th International Conference on, pp. 274–283. IEEE, 2010.

Kong, X. and Yu, P. S. Semi-supervised feature selection for graph classification. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, pp. 793–802, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1.
Kudo, T., Maeda, E., and Matsumoto, Y. An application of boosting to graph classification. *NIPS*, 2004.

Leskovec, J. and Mcauley, J. J. Learning to discover social circles in ego networks. *NIPS*, 2012.

Leskovec, J., Kleinberg, J., and Faloutsos, C. Graphs over time: Densification laws, shrinking diameters and possible explanations. *KDD*, 2005.

Leskovec, J., Adamic, L. A., and Huberman, B. A. The dynamics of viral marketing. *TWEB*, 2007.

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 2009.

Li, G., Semerci, M., Yener, B., and Zaki, M. J. Effective graph classification based on topological and label attributes. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(4):265–283, 2012.

Riesen, K. and Bunke, H. Graph classification based on vector space embedding. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(06):1053–1081, 2009.

West, R. and Leskovec, J. Human wayfinding in information networks. *WWW*, 2012.

West, R., Pineau, J., and Precup, D. Wikispeedia: An online game for inferring semantic distances between concepts. *IJCAI*, 2009.

Wu, K., Watters, P., and Magdon-Ismail, M. Network classification using adjacency matrix embeddings and deep learning. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 299–306, Aug 2016.

Yang, J. and Leskovec, J. Defining and evaluating network communities based on ground-truth. *ICDM*, 2012.