ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10447223

RECOVERING MISSING NODE FEATURES WITH LOCAL STRUCTURE-BASED EMBEDDINGS

Victor M. Tenorio[*], *Madeline Navarro*[†], *Santiago Segarra*[†], *and Antonio G. Marques*[*]

[*] Dept. of Signal Theory and Communications, King Juan Carlos University, Madrid, Spain
[†] Dept. of Electrical and Computer Engineering, Rice University, Houston, USA

ABSTRACT

Node features bolster graph-based learning when exploited jointly with network structure. However, a lack of nodal attributes is prevalent in graph data. We present a framework to *recover completely missing node features for a set of graphs*, where we only know the signals of a subset of graphs. Our approach incorporates prior information from both graph topology and existing nodal values. We demonstrate an example implementation of our framework where we assume that node features depend on local graph structure. Missing nodal values are estimated by aggregating known features from the most similar nodes. Similarity is measured through a node embedding space that preserves local topological features, which we train using a Graph AutoEncoder. We empirically show not only the accuracy of our feature estimation approach but also its value for downstream graph classification. Our success embarks on and implies the need to emphasize the relationship between node features and graph structure in graph-based learning.

Index Terms— Graph Signal Processing, Local Structure Embeddings, Missing Feature Generation

1. INTRODUCTION

For practical applications in chemistry [1,2], medicine [3], and many others [4], data can be naturally represented as interconnected entities using graphs. Supervised learning on graphs aims to predict characteristics using both graph structure and, in some cases, node features, also known as graph signals. These features can improve graph-based predictions when jointly used with graphs, not only when the nodal values are semantically relevant but also when the observations on nodes and the graph structure are dependent [5].

The relationship between node features and their underlying graph is well-studied for node-level tasks. These tasks typically entail predicting nodal characteristics for a single graph where a subset of features is known. Classifying nodes in the semi-supervised learning setting requires the influence of the underlying graph on nodal values to propagate known information to unlabeled nodes [6].

This work was partially supported by the NSF under award CCF-2008555, by the Spanish AEI (AEI/10.13039/ 501100011033) grants PID2019-105032GB-I00, PID2022-136887NB-I00 and FPU20/05554, by the Young Researchers R&D Project, ref. num. F861 (CAM and URJC), and by the Comunidad de Madrid (Madrid ELLIS Unit). Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-17-S-0002. 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 Office or the U.S. Army or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. Emails: victor.tenorio@urjc.es, nav@rice.edu, segarra@rice.edu, antonio.garcia.marques@urjc.es

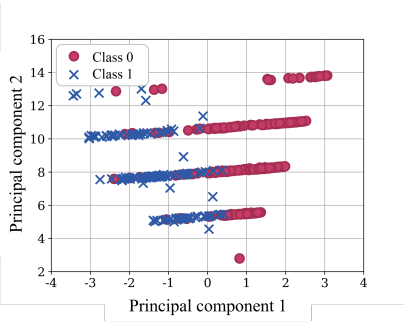

Fig. 1: Embeddings for nodes in graphs from the AIDS molecule dataset. Each point is a node embedding based on local structural characteristics, such as degree. Nodes corresponding to graphs of different classes are shifted in the embedding space, implying that local structure is correlated with molecule class for the AIDS dataset.

Graph signal reconstruction is a common task in graph signal processing in which partially observed features and a known graph signal model are applied together for downstream tasks, such as approximating hidden node values [7–9]. In these cases, a portion of the features is known, even if a minority, and the graph topology informs how existing values provide information about those hidden.

The task of recovering completely missing graph signals for a given graph is far less explored. As nodal values are critical for predictions and necessary for the implementation of graph neural networks (GNNs), we require a method to accurately estimate unknown node features [5,10]. Unlike node-level predictions for which we have partial nodal observations, we cannot use the underlying graph structure to propagate existing information and infer missing values [6, 11]. In such cases, the graph may belong to a family of graphs whose structural and nodal characteristics are related. Many graph-level tasks consist of such data, such as predicting molecular structures and identifying characteristics of social networks [2, 4]. Existing works often characterize graph families by shared random graph models [12] or latent embedding spaces [6]. However, these works enforce a global relationship between the graphs and their signals, requiring knowledge of the entire graph.

Even under the assumption of a shared graph family, the relationship between each of the graphs and its node features is largely decoupled for graph-level learning tasks. For example, methods that interpolate between labeled graphs for improving classification typically treat graphs and node features separately [13–15]. Approaches that do aim to replace missing graph signals typically rely on values that possess solely topological features with no additional nodal information [5, 16], and many use unrelated values that do not incorporate structure [17, 18]. We empirically show that such approaches

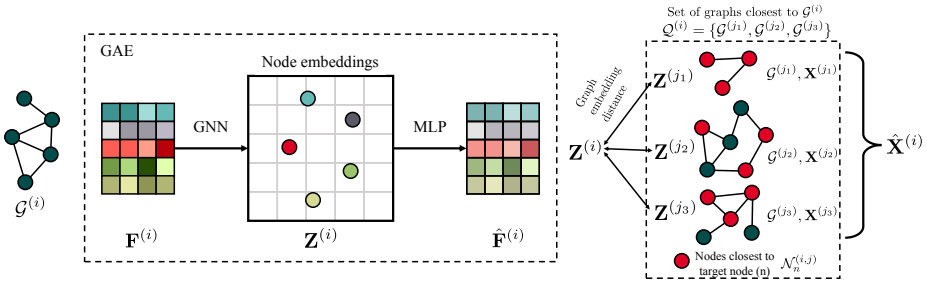

Fig. 2: Schematic of the proposed methodology. First, we compute a feature matrix $\mathbf{F}^{(i)}$ for each graph $\mathcal{G}^{(i)}$ based on structural characteristics. Then, we train a GAE on $\mathbf{F}^{(i)}$ to produce node embeddings $\mathbf{Z}^{(i)}$ and corresponding graph embeddings. The latter are used to measure similarity among graphs, and find the most similar graphs to $\mathcal{G}^{(i)}$, collected in $\mathcal{Q}^{(i)}$. The node embeddings are used to find, among the nodes of the graphs in $\mathcal{Q}^{(i)}$, those nodes which are the closest to each node n from $\mathcal{G}^{(i)}$, collected in $\mathcal{N}_n^{(i,j)}$. Finally, we estimate node features in $\mathcal{G}^{(i)}$ by averaging the features of the closest graphs and their nodes, resulting in realistic yet accurate node feature estimates.

are suboptimal, even when we include structural information.

Given a set of graphs where only a subset has known node features, we present a framework to recover completely missing node features. In particular, we propose *node feature recovery for graph-level tasks incorporating both graph topology and known feature values*. We demonstrate an implementation of this approach assuming that node features depend on local graph structural characteristics, differently from previous approaches that largely rely on aggregating information from a node's neighborhood, for example via low-pass graph filters [19, 20]. Thus, we train a Graph AutoEncoder (GAE) to learn a node embedding space that preserves the local structural characteristics for each node, visualized in Fig. 1. In this setting, feature values are assumed to be closer when node neighborhoods are similar, so nodes with known features can effectively provide the most realistic feature estimates for those with similar local topologies.

Our contributions are as follows.

(i) We present an approach to learn completely missing node features whose values are assumed to be dependent on graph structure, and we exhibit the approach in practice through the setting where features depend on local structure.

(ii) We demonstrate that for many graph classification benchmark datasets, local node structure is indeed indicative of class.

(iii) We empirically validate the ability of our method to not only accurately learn missing node features using a set of graphs with known features, but we also demonstrate the value of recovering accurate node features for downstream tasks.

2. BACKGROUND

In this section, we provide the necessary background on graph-based learning, along with a review of existing approaches on node representation learning and addressing missing node features. We start with some basic notation. A graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ comprises a set of nodes $\mathcal{V} = \{1, \ldots, N\}$ and a set of edges $\mathcal{E} = \{(n_1, n_2) | n_1, n_2 \in \mathcal{V}\}$. Graphs can be conveniently represented by the so-called adjacency matrix $\mathbf{A}$. For edges in $\mathcal{G}$, $(n_1, n_2) \in \mathcal{E}$ iff $A_{n_1,n_2} = 1$. In machine learning and signal processing setups, data is often associated with each of the nodes. In particular, let $\mathbf{X} \in \mathbb{R}^{N \times F}$ be a data matrix, whose entry $X_{n,f}$ represents the value of feature (signal) f at node n. The nth row of $\mathbf{X}$ is typically referred to as the data features associated with node n and the fth column of $\mathbf{X}$ as the fth graph signal. On top of these *node features*, one can also associate a number of *topological features*, such as centrality values or clustering coefficients, with each of the nodes [21].

Node representation learning. Learning node representations (embeddings) has been a prevalent topic of research in the GSP literature almost since its inception [22, 23]. Since the adjacency matrix $\mathbf{A}$ is an alternative representation of $\mathcal{G}$, each node can be (perfectly) represented in an N-dimensional vector space using the corresponding row of $\mathbf{A}$. As a result, node representation algorithms typically aim to learn representations in a lower-dimensional space, that is, they aim at learning a matrix $\mathbf{Z} \in \mathbb{R}^{N \times P}$ with $P < N$. The ultimate goal when designing $\mathbf{Z}$ is to sufficiently characterize nodal behavior in the context of the graph application at hand. Countless approaches to learn nodal representations include algorithms from random walks [24] to GNNs [6, 16].

Most of these approaches learn node embeddings based on node proximity: the closer nodes n and n' are in the graph, the more similar their embeddings $\mathbf{z}_n$ and $\mathbf{z}_{n'}$ are. Recent works have begun to emphasize learning node embeddings based on the role of each node in the graph, guided by its topological features [5, 21]. Under this setting, we may transform structural similarities between nodes into geometric relationships in the embedding space. Inspired by this concept, we draw on such structure-based embeddings to identify which nodes are similar for sharing feature values.

Missing node features. Previous works dealing with missing feature data consider partially missing node features, where only some entries of the feature matrix $\mathbf{X}$ are observed. In such formulations, the task, known as feature imputation [25–27] or graph signal interpolation [7–9], is to learn the missing entries of $\mathbf{X}$. The full matrix $\hat{\mathbf{X}}$ (which contains now both the given and estimated values) is then applied to a downstream task, usually for node-level tasks on a single graph, such as node classification. Feature imputation has been approached by graph spectral approaches [7], kernel approaches [9], propagating the known features [8, 25] or using GNNs [26, 27]. Differently from these works, we aim to solve a more difficult graph-level version of this problem, where for a subset of (or all the) features, we do not have access to any of the nodes, and we must infer the entire set of nodal values (either multiple columns of $\mathbf{X}$ or the full matrix itself) from data associated with other graphs. As detailed in Sec. 3, this paper considers the case of having access to a set of graphs, a subset of which have no observed node features (that is, no access to any of the values of $\mathbf{X}$), which is common in social networks, for example [10].

In the setting of completely missing node features, other works replace these features with carefully crafted random matrices [17, 18], position-dependent values [28–30], or structural properties, such as the degree [5, 16], surprisingly showing that GNNs are still able to obtain great performance without meaningful

	MUTAG	AIDS	PROTEINS	ENZYMES	ogbg-molbbbp	ogbg-molbace
Zeros	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Ones	2.450 ± 0.00	6.083 ± 0.00	1.414 ± 0.00	1.414 ± 0.00	0.806 ± 0.00	0.803 ± 0.00
Random	1.510 ± 0.004	3.550 ± 0.003	0.965 ± 0.002	0.961 ± 0.003	0.896 ± 0.000	0.895 ± 0.000
Degree	0.938 ± 0.001	1.414 ± 0.006	0.899 ± 0.003	0.891 ± 0.001	0.975 ± 0.000	0.984 ± 0.000
LSE-NG ($\bar{Q} = 1$)	0.631 ± 0.040	0.790 ± 0.005	0.739 ± 0.010	0.715 ± 0.005	0.288 ± 0.002	0.264 ± 0.002
LSE-NG ($\bar{Q} = 3$)	0.607 ± 0.012	0.746 ± 0.005	0.719 ± 0.006	0.716 ± 0.007	0.251 ± 0.003	0.228 ± 0.001
LSE-NN ($\bar{Q} = 1$)	$\mathbf{0.119 \pm 0.017}$	$\mathbf{0.555 \pm 0.005}$	0.708 ± 0.010	$\mathbf{0.654 \pm 0.013}$	0.177 ± 0.002	$\mathbf{0.113 \pm 0.001}$
LSE-NN ($\bar{Q} = 3$)	0.128 ± 0.024	0.562 ± 0.005	$\mathbf{0.696 \pm 0.007}$	0.664 ± 0.011	$\mathbf{0.175 \pm 0.004}$	0.126 ± 0.002

Table 1: Normalized feature generation error $\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 / \|\mathbf{X}\|_F^2$. For LSE-NG and LSE-NN, we predict missing node features for each graph from the $\bar{Q}$ nearest graphs in the dataset with respect to the graph embeddings in the same class. The top performing methods are **bolded**.

node features and only using the graph structure. However, random and constant features are independent of class labels, and we empirically demonstrate that using random or structural node features is suboptimal. Moreover, our approach using a realistic estimate of node features exhibits superior performance.

3. METHODOLOGY

We introduce our proposed approach to estimate missing node features using structural information and known node features, which is visualized in Fig. 2. While the schematic in Fig. 2 illustrates the use of local structure for sharing nodal values, other assumptions can easily be made to associate nodes for feature learning.

Consider a graph dataset $\mathcal{T} = \{(\mathcal{G}^{(i)}, \mathbf{X}^{(i)}, y^{(i)})\}_{i=1}^{T}$, where for the i-th sample we have the graph $\mathcal{G}^{(i)}$ with N_i nodes, $\mathbf{X}^{(i)} \in \mathbb{R}^{N_i \times F}$ is a matrix of node features of length F, and $y^{(i)}$ is the associated label. Let $\mathcal{T}_{\text{miss}} \subset \mathcal{T}$ be a subset of $\mathcal{T}$ with missing features, where for every $(\mathcal{G}, \mathbf{X}, y) \in \mathcal{T}_{\text{miss}}$, we only know the duplex $(\mathcal{G}, y)$, and define $\mathcal{T}_{\text{full}} = \mathcal{T} \backslash \mathcal{T}_{\text{miss}}$. Our focus in this work is to recover the missing features $\mathbf{X}$ for every $(\mathcal{G}, \mathbf{X}, y) \in \mathcal{T}_{\text{miss}}$, resulting in a set $\hat{\mathcal{T}}_{\text{miss}}$ consisting of triplets $(\mathcal{G}, \hat{\mathbf{X}}, y)$ with approximated features $\hat{\mathbf{X}}$. Subsequently, we may use $\hat{\mathcal{T}} = \mathcal{T}_{\text{full}} \cup \hat{\mathcal{T}}_{\text{miss}}$ for downstream tasks such as graph classification.

Our approach consists of two steps: we first learn a node embedding space preserving graph structural information through which we compute node similarity, and then we predict the values of missing node features using nearby node embeddings. While we select local structural characteristics as the topological features of interest, note that our framework is amenable to any choice of embedding space that allows us to relate nodes based on similarity.

Node embedding space. We first obtain a latent space with which we compute node similarity. We train a GAE to generate node embeddings characterized by node roles, that is, their local structure [21]. More precisely, the GAE consists of a graph convolutional network (GCN) as the encoder to learn from structural characteristics, both global and local, and a multilayer perceptron (MLP) as the decoder to invert the embedding process. For a graph $\mathcal{G}^{(i)}$, the GCN encoder takes as input $\mathcal{G}^{(i)}$ and a corresponding feature matrix $\mathbf{F}^{(i)} \in \mathbb{R}^{N_i \times F}$ containing structural information from F features. We apply the features in [21], including local characteristics such as node degree and clustering coefficient, although any features may be used to emphasize different structural behavior. The parameters $\mathbf{\Theta}$ of the GAE $f_{\mathbf{\Theta}}$ are trained to minimize the loss between the output of the GAE and the input feature matrix

$$\min_{\mathbf{\Theta}} \|\mathbf{F}^{(i)} - f_{\mathbf{\Theta}}(\mathbf{F}^{(i)}, \mathcal{G}^{(i)})\|_F^2,$$

where $f_{\mathbf{\Theta}}$ represents the GAE, whose output is computed as $f_{\mathbf{\Theta}}(\mathbf{F}, \mathcal{G}^{(i)}) = \text{MLP}_{\mathbf{\Theta}_2}(\text{GNN}_{\mathbf{\Theta}_1}(\mathbf{F}, \mathcal{G}^{(i)}))$, where $\mathbf{\Theta} = \{\mathbf{\Theta}_1, \mathbf{\Theta}_2\}$ and the GNN is defined via the following recursion [6]

$$\mathbf{H}^{(\ell)} = \sigma\left(\tilde{\mathbf{A}}\mathbf{H}^{(\ell-1)}\mathbf{\Theta}^{(\ell)}\right), \tag{1}$$

where $\mathbf{H}^{(\ell)}$ are the hidden features at layer ℓ; $\mathbf{\Theta}_1 = \{\mathbf{\Theta}^{(\ell)}\}_{\ell=1}^{L}$ are the learnable parameters of the L layers; and σ is a pointwise non-linearity. We let $\mathbf{A}$ denote the adjacency matrix of the input graph $\mathcal{G}^{(i)}$, and we define $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\hat{\mathbf{D}} = \text{diag}(\hat{\mathbf{A}}\mathbf{1})$, and $\tilde{\mathbf{A}} = \hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}$. Note that the matrix multiplication in (1) can be understood as a low-pass graph filtering [19], where the nodes average their own value with the values of their neighbors. As a result, the embeddings based on (1) will promote similar representations for nodes whose local structural features are similar. A visualization of the resultant node embeddings from graphs in the molecular classification dataset AIDS [31] is shown in Fig. 1.

Given the GAE, we obtain node embeddings for graph $\mathcal{G}^{(i)}$ as $\mathbf{Z}^{(i)} = \text{GNN}_{\mathbf{\Theta}_1}(\mathbf{F}^{(i)}, \mathcal{G}^{(i)}) \in \mathbb{R}^{N_i \times P}$, where P is the dimension of the embedding space. We further define graph embeddings $\mathbf{z}^{(i)} \in \mathbb{R}^P$ by computing the average across the node dimension, that is, $\mathbf{z}^{(i)} = \frac{1}{N_i}\sum_{k=1}^{N_i} \mathbf{z}_k^{(i)} \in \mathbb{R}^P$ where $\mathbf{z}_k^{(i)} \in \mathbb{R}^P$ is kth row of $\mathbf{Z}^{(i)}$.

With the node and graph embeddings for every graph in $\mathcal{T}$, we associate $\mathcal{G}^{(i)} \in \mathcal{T}_{\text{miss}}$ with nearby graphs and nodes with respect to the embedding space. Let $\mathcal{Q}^{(i)} \subset \mathcal{T}_{\text{full}}$ be the set of the $\bar{Q}$ nearest graphs to $\mathcal{G}^{(i)}$ with the same label, that is, for every $\mathcal{G}^{(j)} \in \mathcal{Q}^{(i)}$, we have that $y^{(i)} = y^{(j)}$. More specifically, all the graphs in $\mathcal{Q}^{(i)}$ are closer to $\mathcal{G}^{(i)}$ than those with the same label but not in $\mathcal{Q}^{(i)}$ using as distance the error between their embeddings $\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\|_2$. For nearby nodes, we similarly let $\mathcal{N}_n^{(i,j)}$ be the set containing the $\bar{N}$ nodes of graph $\mathcal{G}^{(j)} \in \mathcal{Q}^{(i)}$ closest to node n of graph $\mathcal{G}^{(i)} \in \mathcal{T}_{\text{miss}}$. That is, all of the $\bar{N}$ nodes from graph $\mathcal{G}^{(j)}$ in $\mathcal{N}_n^{(i,j)}$ are closer (in terms of the same distance previously defined) to node n in graph $\mathcal{G}^{(i)}$ than those not in $\mathcal{N}_n^{(i,j)}$.

Predicting missing graph signals. Once we are able to compute node similarity, we predict the missing node features using the nearest graphs and nodes. We propose to generate the features $\hat{\mathbf{X}}^{(i)}$ of $\mathcal{G}^{(i)} \in \mathcal{T}_{\text{miss}}$ as the average of the features in the closest nodes and graphs to $\mathcal{G}^{(i)}$ in $\mathcal{T}_{\text{full}}$ with respect to their embeddings. Let $\mathbf{C}^{(i,j)} \in \mathbb{R}^{N_i \times N_j}$ be the transformation matrix mapping the features from $\mathcal{G}^{(j)} \in \mathcal{Q}^{(i)}$ to $\mathcal{G}^{(i)}$, where the entry at the nth row and ℓth column is

$$C_{n,\ell}^{(i,j)} = \begin{cases} \frac{1}{|\mathcal{N}_n^{(i,j)}|} & n \in \{1, ..., N_i\}, \ell \in \mathcal{N}_n^{(i,j)} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

For each node n in $\mathcal{G}^{(i)}$, the product $\mathbf{C}^{(i,j)}\mathbf{X}^{(j)}$ computes the average of the features from the nodes in $\mathcal{N}_n^{(i,j)}$, that is, the closest nodes to n from $\mathcal{G}^{(j)}$. Given the set of closest graphs $\mathcal{Q}^{(i)}$ to $\mathcal{G}^{(i)}$, we compute our node feature estimates for $\mathcal{G}^{(i)}$ as

$$\hat{\mathbf{X}}^{(i)} = \frac{1}{\bar{Q}} \sum_{j \in \mathcal{Q}^{(i)}} \mathbf{C}^{(i,j)}\mathbf{X}^{(j)}. \tag{3}$$

	MUTAG	AIDS	PROTEINS	ENZYMES	ogbg-molbbbp	ogbg-molbace
True Features	83.25 ± 9.52	97.00 ± 1.49	72.74 ± 3.75	36.17 ± 4.97	85.53 ± 2.38	74.38 ± 3.49
Zeros	78.75 ± 9.34	95.92 ± 1.87	70.09 ± 5.25	25.50 ± 7.15	77.15 ± 2.63	53.33 ± 5.25
Ones	82.25 ± 9.55	94.45 ± 1.86	69.25 ± 5.26	24.08 ± 5.10	77.77 ± 3.26	53.55 ± 4.50
Random	81.50 ± 8.23	94.30 ± 2.80	70.75 ± 4.66	22.92 ± 6.34	77.61 ± 3.46	53.73 ± 4.20
Degree	81.00 ± 12.41	90.60 ± 4.88	69.65 ± 5.38	25.42 ± 6.32	79.71 ± 3.40	56.34 ± 4.77
Not using $\mathcal{T}_{\text{miss}}$	$\mathbf{84.75 \pm 8.73}$	95.62 ± 1.56	71.28 ± 4.87	21.83 ± 6.75	83.27 ± 2.13	67.23 ± 5.27
LSE-NG ($Q = 1$)	83.00 ± 7.48	95.75 ± 1.65	71.68 ± 3.76	24.08 ± 5.01	82.94 ± 1.90	69.02 ± 4.10
LSE-NG ($\bar{Q} = 3$)	81.50 ± 9.89	95.17 ± 1.70	$\mathbf{72.79 \pm 4.89}$	24.42 ± 5.30	83.72 ± 1.37	70.28 ± 2.94
LSE-NN ($Q = 1$)	81.50 ± 9.63	$\mathbf{96.10 \pm 1.44}$	71.15 ± 4.57	23.83 ± 6.10	83.01 ± 1.54	$\mathbf{71.63 \pm 4.12}$
LSE-NN ($\bar{Q} = 3$)	78.25 ± 8.26	95.97 ± 1.63	70.53 ± 4.07	$\mathbf{26.00 \pm 6.13}$	$\mathbf{84.08 \pm 2.10}$	70.33 ± 3.57

Table 2: Accuracy in the test dataset obtained by the baselines and by the approach presented in this work in the downstream task (graph classification). The best performances (excluding those obtained using the true features) are **bolded**.

These features complete the triplet $(\mathcal{G}^{(i)}, \hat{\mathbf{X}}^{(i)}, y^{(i)})$ for every estimate in $\hat{\mathcal{T}}_{\text{miss}}$, allowing us to use these graphs for downstream tasks such as graph classification. Observe that our method alters the dataset by computing the topological features, training the GAE, and estimating missing features from the nearest graphs and nodes. Thus, only model training is affected, and the complexity of any downstream task is unchanged.

4. RESULTS

We showcase the capabilities of our proposed approach for missing feature generation. We demonstrate our method in comparison with several baselines in numerical experiments, both for node feature learning and downstream graph classification.

Datasets. We use six real-world data benchmarks: MUTAG, AIDS, PROTEINS and ENZYMES from the TUDataset collection [31], and ogbg-molbace and ogbg-molbppp from the OGBG collection [32]. Not only are these standard benchmark datasets for classification tasks, they also provide node features, which allows us to test the hypothesis that node features are relevant for the classification task as well as to evaluate our proposed approach. The datasets contain either graphs representing molecules (MUTAG, AIDS, ogbg-molbbbp and ogbg-molbace), where nodes represent atoms and edges represent chemical bonds, or proteins (PROTEINS) and enzymes (ENZYMES), where nodes represent structural elements and edges encode node proximity.

Experimental setup. We split the dataset with 10% of the data for validation; 10% for testing; 30% for training, or $\mathcal{T}_{\text{full}}$; and 50% as missing, that is, the set of graphs with missing node features $\mathcal{T}_{\text{miss}}$. We conduct 15 random realizations of these splits, and the results presented in Tables 1 and 2 list the mean and standard deviation of the metric of interest (to be defined next) across every realization.

We demonstrate the efficacy of our method on both feature generation and downstream graph classification. Our approach that uses the local structure-based node embeddings of nearest neighbors, denoted "LSE-NN", is compared to several baselines. Alternatives to "LSE-NN" include classical approaches: "Degree" denoting node degree, "Ones" for features of all ones, "Zeros" for features of all zeros, "Random" with features sampled uniformly at random on $[0, 1]$.

Our method "LSE-NN" exploits not only similar graphs for estimating missing node features but also nodes with similar local structures. We highlight the benefits of such an approach by also comparing it to a modification denoted "LSE-NG". For this variant, we obtain the nearest graphs $\mathcal{Q}^{(i)}$ for $\mathcal{G}^{(i)}$ as for "LSE-NN", but we then assign feature values to each node k in $\mathcal{G}^{(i)}$ by taking node features uniformly at random from the nodes of graphs $\mathcal{G}^{(j)} \in \mathcal{Q}^{(i)}$. This is equivalent to replacing $\mathbf{C}^{(i,j)}$ in (3) with a random permutation matrix $\mathbf{P}^{(i,j)} \in \{0, 1\}^{N_i \times N_j}$. Thus, "LSE-NG" predicts node features using similar graphs but does not align nodes by local structure.

Feature generation performance. We first compare the ability of each method to recover the original node features. The results presented in Table 1 show the node feature estimation error as $\|\hat{\mathbf{X}} - \mathbf{X}\|_F^2 / \|\mathbf{X}\|_F^2$, where $\hat{\mathbf{X}}$ denotes the estimated features and $\mathbf{X}$ the true ones. We see that our architecture consistently beats the alternatives, achieving a lower error in every dataset considered, in some cases by a large margin. This shows that the true node features, which are assumed to be the optimal features for downstream tasks, are best recovered by our architecture, without the need for partial observations on the graphs with missing features.

Graph classification performance. We also assess the utility of the estimated features for graph classification. The results are shown in Table 2, where we present label prediction accuracy using a GNN model trained with the estimated features. We choose the Graph Isomorphism Network (GIN) [16] as our GNN architecture, a standard model for graph classification. For this task, we also add an additional baseline "Not using $\mathcal{T}_{\text{miss}}$", where we train the GIN only on the subset of graphs $\mathcal{T}_{\text{full}}$ with known node features, ignoring the graphs in $\mathcal{T}_{\text{miss}}$ with missing features. In all cases but the MUTAG dataset, the best performance is achieved using the learned GAE embeddings, either from randomly copying nodes from the nearest graphs "LSE-NG", which enjoys the best performance on PROTEINS dataset, or by using the nearest nodes "LSE-NN", which obtains superior performance for all other datasets. Thus, not only do we infer missing node features accurately, but the estimates are sufficiently realistic to bolster classification performance when we do not observe the node features of many graphs. Moreover, the superior performance of "Not using $\mathcal{T}_{\text{miss}}$" over the true features for MUTAG implies that node features may not be semantically relevant. Note that while we could improve performance by training the GAE and GIN end-to-end, our focus is node feature recovery, while graph classification serves as a relevant downstream task.

5. CONCLUSION

In this work, we proposed a framework to recover completely missing node features for a set of graphs. We implemented this framework for estimating features that are characterized primarily by local graph structure. To this end, we presented a node embedding space using only local topological features. The embedding space provided a node similarity metric with which we estimated missing node features using similar nodes from nearby graphs. Our estimates aid graph classification when features are missing, emphasizing the need for accurate nodal characteristics. In the future, we will generalize to applications such as graph data augmentation, where we can generate synthetic graphs with realistic node features. Our work connecting node features and graph structure can bolster the success of graph-based learning by exploiting not only structural information but also values explicitly embedded therein.

6. REFERENCES

[1] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Intl. Conf. on Machine Learning (ICML)*, vol. 70, pp. 1263–1272, PMLR, 2017.

[2] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in Neural Info. Process. Syst.*, vol. 28, 2015.

[3] S. I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker, and D. Rueckert, "Distance metric learning using graph convolutional networks: Application to functional brain networks," in *Medical Intl. Joint Conf. on Artif. Intell. and Computer Assisted Intervention (MICCAI)*, vol. 10433, pp. 469–477, 2017.

[4] J. Kim and M. Hastak, "Social network analysis: Characteristics of online social networks after a disaster," *Intl. J. of Info. Management*, vol. 38, no. 1, pp. 86–96, 2018.

[5] H. Cui, Z. Lu, P. Li, and C. Yang, "On positional and structural node features for graph neural networks on non-attributed graphs," in *ACM Inter. Conf. on Info. & Knowledge Man.*, CIKM, p. 3898–3902, 2022.

[6] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Intl. Conf. on Learning Representations (ICLR)*, 2017.

[7] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, 2015.

[8] D. Ramirez, A. G. Marques, and S. Segarra, "Graph-signal reconstruction and blind deconvolution for structured inputs," *Signal Process.*, vol. 188, p. 108180, 2021.

[9] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Trans. Signal Process.*, vol. 65, pp. 764–778, 2016.

[10] C. Cai and Y. Wang, "A simple yet effective baseline for non-attributed graph classification," *arXiv preprint arXiv:1811.03508*, 2018.

[11] X. Chen, S. Chen, J. Yao, H. Zheng, Y. Zhang, and I. W. Tsang, "Learning on attribute-missing graphs," *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 44, no. 2, pp. 740–757, 2022.

[12] M. Navarro and S. Segarra, "Joint network topology inference via a shared graphon model," *IEEE Trans. Signal Process.*, vol. 70, pp. 5549–5563, 2022.

[13] X. Han, Z. Jiang, N. Liu, and X. Hu, "G-Mixup: Graph data augmentation for graph classification," in *Intl. Conf. on Machine Learning (ICML)*, vol. 162, pp. 8230–8248, PMLR, 2022.

[14] X. Ma, X. Chu, Y. Wang, Y. Lin, J. Zhao, L. Ma, and W. Zhu, "Graph interpolation via fast Fused-Gromovization," *arXiv preprint arXiv:2306.15963*, 2023.

[15] M. Navarro and S. Segarra, "GraphMAD: Graph mixup for data augmentation using data-driven convex clustering," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, pp. 1–5, 2023.

[16] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *Intl. Conf. on Learning Representations (ICLR)*, 2019.

[17] R. Abboud, İ. İ. Ceylan, M. Grohe, and T. Lukasiewicz, "The surprising power of graph neural networks with random node initialization," in *Intl. Joint Conf. on Artif. Intell.*, 2021.

[18] R. Sato, M. Yamada, and H. Kashima, "Random features strengthen graph neural networks," in *SIAM Intl. Conf. on Data Mining (SDM)*, 2021.

[19] E. Isufi, F. Gama, D. I. Shuman, and S. Segarra, "Graph filters for signal processing and machine learning on graphs," *arXiv preprint arXiv:2211.08854*, 2022.

[20] H. Liu, A. Scaglione, and H.-T. Wai, "Blind graph matching using graph signals," *arXiv preprint arXiv:2306.15747*, 2023.

[21] X. Guo, W. Zhang, W. Wang, Y. Yu, Y. Wang, and P. Jiao, "Role-oriented graph auto-encoder guided by structural information," in *Database Syst. for Advanced Applications*, pp. 466–481, 2020.

[22] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowledge and Data Engineering*, vol. 30, pp. 1616–1637, 2017.

[23] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.

[24] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Intl. Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 701–710, ACM, 2014.

[25] E. Rossi, H. Kenlay, M. I. Gorinova, B. Chamberlain, X. Dong, and M. M. Bronstein, "On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features," *arXiv preprint arXiv:2111.12128*, 2021.

[26] J. You, X. Ma, D. Ding, M. Kochenderfer, and J. Leskovec, "Handling missing data with graph representation learning," *Advances in Neural Info. Process. Syst.*, 2020.

[27] I. Spinelli, S. Scardapane, and A. Uncini, "Missing data imputation with adversarially-trained graph convolutional networks," *Neural Netw.*, vol. 129, pp. 249–260, 2020.

[28] J. You, R. Ying, and J. Leskovec, "Position-aware graph neural networks," in *Intl. Conf. on Machine Learning (ICML)*, vol. 97, pp. 7134–7143, PMLR, 2019.

[29] P. Li, Y. Wang, H. Wang, and J. Leskovec, "Distance encoding: Design provably more powerful neural networks for graph representation learning," in *Advances in Neural Info. Process. Syst.*, vol. 33, pp. 4465–4478, 2020.

[30] J. You, J. M. Gomes-Selman, R. Ying, and J. Leskovec, "Identity-aware graph neural networks," *AAAI Conf. on Artif. Intell.*, vol. 35, no. 12, pp. 10737–10745, 2021.

[31] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "TUDataset: A collection of benchmark datasets for learning with graphs," in *Intl. Conf. on Machine Learning (ICML)*, 2020.

[32] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open Graph Benchmark: Datasets for Machine Learning on Graphs," *arXiv preprint arXiv:2005.00687*, vol. abs/2005.00687, 2020.