HAL open science

Boundary heat diffusion classifier for a semi-supervised learning in a multilayer network embedding

Mohan Timilsina, Vít Nováček, Mathieu D'aquin, Haixuan Yang

To cite this version:

Mohan Timilsina, Vít Nováček, Mathieu D'aquin, Haixuan Yang. Boundary heat diffusion classifier for a semi-supervised learning in a multilayer network embedding. Neural Networks, 2022, 156, pp.205-217. 10.1016/j.neunet.2022.10.005 . hal-03919966

HAL Id: hal-03919966

https://inria.hal.science/hal-03919966v1

Submitted on 4 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Boundary Heat Diffusion Classifier for a Semi-Supervised Learning in a Multilayer Network Embedding

Mohan Timilsina[a,*,1], Vít Nováček[a,b,c,3], Mathieu d'Aquin[a] and Haixuan Yang[c]

[a]Data Science Institute, Insight Centre for Data Analytics, National University of Ireland Galway, Ireland
[b]Faculty of Informatics, Masaryk University Brno, Czech Republic
[c]Masaryk Memorial Cancer Institute, Brno, Czech Republic
[c]School of Mathematics, Statistics and Applied Mathematics, National University of Ireland Galway, Ireland

ARTICLE INFO

Keywords:
Multiplex Network
Diffusion
Heat
Prediction
Label

ABSTRACT

The scarcity of high-quality annotations in many application scenarios has recently led to an increasing interest in devising learning techniques that combine unlabeled data with labeled data in a network. In this work, we focus on the label propagation problem in multilayer networks. Our approach is inspired by the heat diffusion model, which shows usefulness in machine learning problems such as classification and dimensionality reduction. We propose a novel boundary-based heat diffusion algorithm that guarantees a closed-form solution with an efficient implementation. We experimentally validated our method on synthetic networks and five real-world multilayer network datasets representing scientific coauthorship, spreading drug adoption among physicians, two bibliographic networks, and a movie network. The results demonstrate the benefits of the proposed algorithm, where our boundary-based heat diffusion dominates the performance of the state-of-the-art methods.

1. Introduction

Real-world networks[1] often demonstrate a layered structure in which links in each layer reflecting the interaction of nodes in different environments [1]. These interconnected networks are often called *network of networks* [2] or multilayer networks. Multilayer networks provide better modeling for complex networks [3], enabling multiple network layers to represent features of natural systems [4]. Therefore, the complexity associated with interactions is captured by multilayer networks [5, 6]. The computational problems in Multilayer networks such as link prediction [7] and community detection [6, 5] are also actively researched in this domain.

A multilayer network can be generalized as a "multi-relational network" in a data mining community [8]. In a multilayer network, the same nodes are linked by different networks (layers). For instance, multilayers are good descriptions of a scientist's social network, where the nodes represent the scientist, and the different layers correspond to different types of scientific conferences they attend. For illustration purposes, Figure 1 (A) shows how the scientists are talking to each other in three different Artificial Intelligence (AI) conferences (ICML, NeurIPS, IJCAI), represented by different colors. These three different conferences are the three different layers in the multilayer networks. The important observation in this multilayer network is that the same scientist in every layer or conference can be captured by interlayer edges connecting each scientist to its copies in other layers. In most cases, the multilayer network analysis is done in an aggregated network by flattening the whole layers into a single layer. Figure 1 (B) demonstrates the projection of a multilayer network of AI scientists onto graph structure by summing up all the adjacency matrices of the layers to a composite matrix. The main problem with doing this is the loss of information on layers. So performing any diffusion algorithms for node classification on the aggregate graph structure may not perform accurately. Thus the underlying problem is that the matrix representation is not expressive enough to model a multilayer network without information loss.

Thus to apply label propagation algorithms in a multilayer network of scientists described above, we need to transform the multilayer network into a homogenous network without compromising its distinctive layer properties. Figure 1 (C) demonstrates the feature learning of the scientist in a multilayer graph via tensor embedding methods. Tensors,

mohan.timilsina@insight-centre.org (M. Timilsina); vit.novacek@insight-centre.org (V. Nováček); mathieu.daquin@insight-centre.org (M. d'Aquin); haixuan.yang@nuigalway.ie (H. Yang)
ORCID(s):

[1]Note that the words "networks" and "graphs" are interchangeably used in the paper.

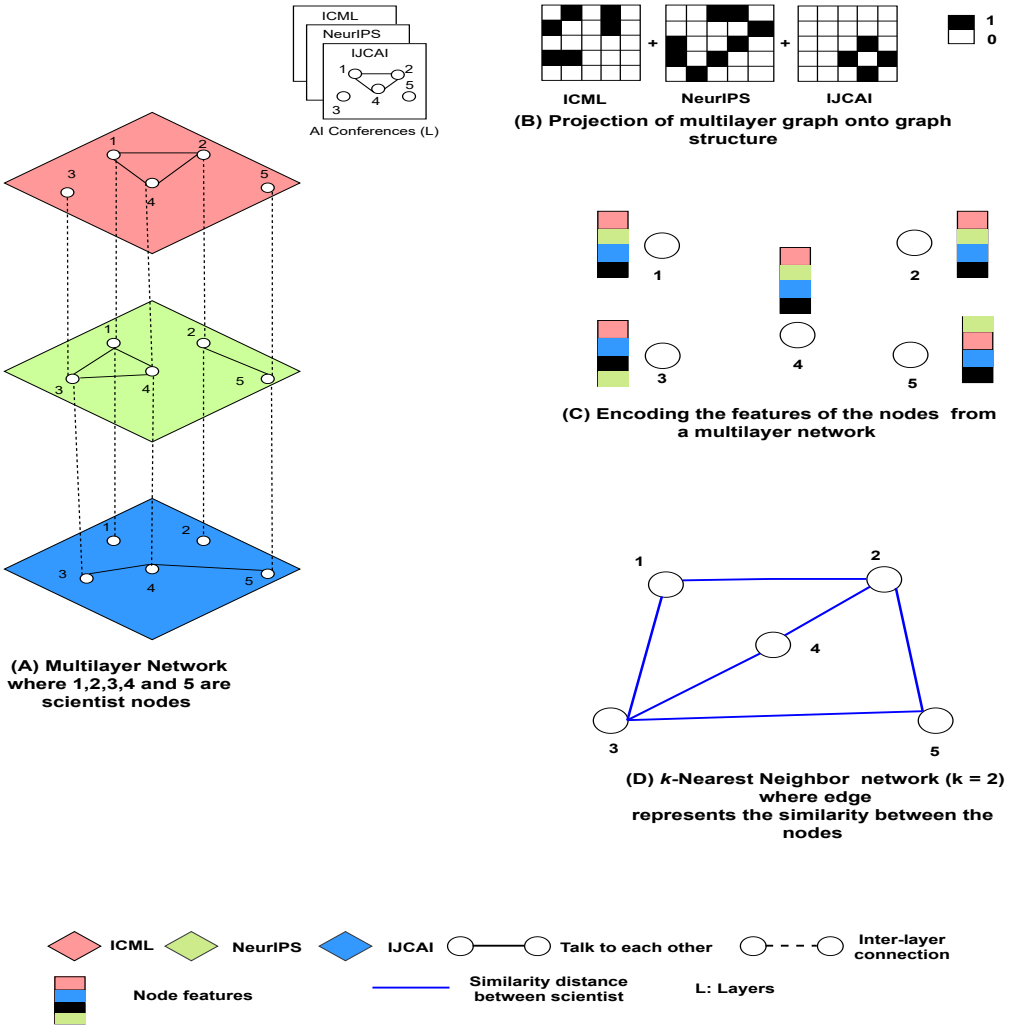


Figure 1: (A) Multilayer network of AI scientists talking to each other. Each layer represents an AI conference (ICML, NeurIPS, IJCAI). (B)Projection of multilayer graph onto a graph structure (C) Node feature extraction using graph embedding method. (D) Constructing the homogeneous graph of the extracted features using kNN method.

which are n-modal generalizations of matrices, naturally adapt the graph's multilayer representation and allow us to learn the node's embeddings. Similarly, Figure 1 (D) demonstrates the transformation of such learned features or embeddings into a simple k-nearest neighbor (kNN) network of scientists based on some measure of similarity between features. Another critical problem in a multilayer network is the speed of the diffusion process. Multilayer networks can have an enhanced-diffusive behavior, which means that the time scale associated with it is shorter than that occurring on a single-layer network [9]. For example, a scientist discussing significant findings in the ICML conference may be quickly diffused in NeurIPS or IJCAI conferences. It means information travels very fast in such a network. If a node classification has to be done in such a network, then the most crucial property to take care of is the time. The diffusion kernels [10, 11] used in the standard label propagation algorithm capture the long-range relationships (global information) between nodes in the network. However, some specific real-world multilayer networks tend to link related entities by shorter diffusion paths [12], which favor short-range diffusion. Thus we need to consider the time parameter in the multilayer graph, which can adapt to long and short-range diffusion for node classification.

The social and technological innovation brought by, for example, the world wide web, biomedicine, and social networks have exposed the need to consider that networks might be made up of many different layers of interactions. Compared to single-layer networks, a multilayer network's topological and dynamic properties are different [13]. Therefore,

studying propagation processes in multilayer networks is a rapidly evolving research area. For instance, a diffusion process can have an enhanced-diffusive behavior on a multilayer network, which means that the time scale associated with it is shorter than that occurring on a single layer network [9]. Due to this, it is essential to consider how label propagation algorithms work in a multilayer network.

Although label propagation algorithms work reasonably well in most networks with a single layer, we do not explicitly know how these algorithms behave in a multilayer network. In such a network, where nodes overlap between the layers, there is a high possibility of node misclassification using ordinary label propagation algorithms [14]. The diffusion kernels [10] used in the label propagation algorithm capture the long-range relationships (global information) between nodes in the network. Due to this reason, long-range diffusion puts more emphasis on random walks that explore more of the multilayer network, which eventually leads to misclassification. However, specific real-world multilayer networks tend to link related entities by shorter diffusion paths. For example, proteins that have similar functions are often linked by the shortest paths in a network [12].

Another example is image segmentation, which is one of the critical areas for extracting information from the images in a computer vision problem. Currently, the use of a Multilayer network [15] has improved the precision of image segmentation because it allows the analysis of networks with multiple resolutions. However, the multilayer networks applied in image segmentation [16] use a classical label propagation algorithm applicable for a single-layer network that might misclassify the labels of the nodes shared across layers.

The main advantage of the heat diffusion algorithm over other label propagation algorithms is two folds. First, it utilizes the heat diffusion process to determine neighboring nodes that reflect the local structure of the target node and the relevant information of smoothness manifested in graph structure [17]. Second, the heat propagates the information faster than the standard label propagation algorithm by penalizing the shorter walks heavily in the graph [18]. This property enables the faster convergence of the heat diffusion algorithm.

Motivation: Using traditional label propagation algorithms, the main problem is that the diffusion process undergoes a unique stationary distribution. It is often called deep or long-range diffusion. This property emphasizes random walks that explore more of the network. The study by Gomez et.al [9] has also shown that diffusive processes in multilayer networks are faster than in any single-layer networks. Therefore, the unique stationary distribution of random walks might cause misclassification of nodes in a layered network. This problem will further enable us to adopt shallow or short-range diffusion. It is because heat diffusion has the property of determining neighboring nodes that reflect the local structure of the target node, which will further control the heat propagation in the layers and efficiently classify the nodes in a multilayer network.

Main idea: In this paper, we propose a novel node classification algorithm that handles the abovementioned concerns. Our algorithm uses the intuitive and natural model of a physical heat diffusion system with boundary conditions. The heat flow can be captured by measuring the (i) heat between points in the network and (ii) the heat amount added and removed from the system. Here, the points at which heat is measured can be represented by nodes in a graph, and edges are associated with heat flows between those points. The injection and extraction points can be viewed as the boundaries of the system. The diffusion time models the range of diffusion, small time for a short-range diffusion and large time for a long-range one. Based on this idea, heat diffusion with boundary conditions will control the heat flow, making it ideal for node classification in a multilayer network.

Contributions: Our contributions based on heat diffusion with boundary condition (BHD) are summarized as follows:

1. We provide a hybrid approach for combining the graph embedding and the diffusion method in a multilayer network.
2. Theoretically, we show that the popular transductive semi-supervised kernel known as the harmonic function is the limit case of BHD.
3. We develop an iterative method to BHD, whose computation complexity is linear to the number of edges.

Consequently, BHD has the following advantages for its applications:

1. Accuracy: Our algorithm achieves improved accuracy on different label propagation tasks in a multilayer network when compared to state-of-the-art label propagation methods.
2. Scalable: it can be applied to a large graph as BHD is linear in the number of edges.
3. Parameter estimation: BHD has just one parameter chosen using cross-validation from a training set.

Moreover, we performed extensive experiments in synthetic datasets and five multi-layered networks for the node classification task. The five different labeled multilayer networks include a scientific coauthorship network, a diffusion

innovation in a physician network, two bibliographic networks, and a movie network. The results demonstrated that our algorithm often outperforms the state-of-the-art label propagation algorithms in terms of top p% label prediction and classification accuracy. To the best of our knowledge, our algorithm is the first solution to handle node classification using label propagation in a multilayer network relying only on the graph structure.

Outline. The rest of the paper is organised in a rather standard way: related work, problem definition, method description, experiments and conclusion.

2. Related Work

2.1. Label Propagation

Label propagation technique is very powerful [19], it has been identified and re-identified in numerous fields under different forms [20, 21, 22]. For instance, theoretical graph scientists explored random walks on graphs [23, 24]; the data science community applies variants of the Google PageRank search algorithm [21]; statistical physicists examined heat diffusion processes [25]; electrical engineers calculate minimum energy states within an electrical circuit [26]; and the machine learning (ML) community considers different forms of graph kernels [10].

Several algorithms can solve the node classification problem from a Label Propagation (LP) perspective. Blum et al. proposed Mincut [27], and Zhu et al. proposed LP [11] known as harmonic function (HMN), which is one of the most well-known graph-based semi-supervised learning algorithms in the Artificial Intelligence (AI) community for transductive learning. Similarly, the Local and Global Consistency (LGC) method proposed by Zhou et al. [28] is based on the assumption that nearby points (local) are likely to have the same label; and points on the same structure (global) are also likely to have the same label also known as homophily. On a similar note, Heat Diffusion (HD) proposed by Yang et al. [29] has also been successfully applied to node classification tasks [30, 31]. OMNI-Prop [32] is an LP-like algorithm that applies to both *homophilic*, and *heterophilic* labeled network, where dissimilar nodes are more likely to be related than similar ones. Among semi-supervised graph learning methods, LP [33, 28, 34] has shown good adaptability, scalability, and efficiency for node classification. One of the advantages of the LP-based technique is that they have a small memory requirement and a fast convergence rate [35] which makes it attractive to apply in large graphs. Thus in the real-world network datasets, LP has shown to be very beneficial, such as social networks, web pages, protein-protein interactions, citation, and anti-money laundering in the Bitcoin network [36, 37, 38, 39, 40]. Furthermore, LP has shown huge advantage in drug mechanism of action prediction [41, 42], spammer reviewer detection [43], and object recognition [44]. Recently, heat diffusion with a boundary-based approach has also shown good performance in semi-supervised regression setting [45]. Song et al. [46] has provided the latest comprehensive survey of the graph-based semi-supervised learning.

Most of the LP method described contributes related and mathematically equivalent techniques, including random walks on a graph, diffusion processes on a graph, and current computations in electric networks. A brief demonstration of such technique on how they use the similarity matrix and weight normalization is shown in Table 1.

LP Variants	Similarity Matrix	Normalized Weight	Relevant Methods
Random Walk	W^k	$W = AD^{-1}$	Electric Network; HMN
Random Walk with Restart	$\alpha(I-(1-\alpha)W)^{-1}$	$W = AD^{-1}$; $W = D^{-1/2}AD^{-1/2}$	Insulated diffusion; personalized PageRank; LGC
Diffusion Kernel	$e^{-\alpha W}$	$W = D - A$	HD

Table 1
k denotes the number of time steps for propagation; A denotes the adjacency matrix, which could be weighted or unweighted; D denotes the diagonal degree matrix; I denotes the identity matrix; α is the smoothing parameter.

Table 2 demonstrates the qualitative comparison of the popular LP algorithms that are applied in a node classification problem. We observed that most of these state-of-the-art methods have the closed form solution and scales linearly to the large graphs. All the models above have proven useful in single layer network analysis, but their performance on multilayer networks has not been explored.

	Mincut	HMN	LGC	HD	OMNI	Proposed Boundary Heat Diffusion (BHD)
Closed form solution	✓	✓	✓	✓	✓	✓
Convergence	✓	✓	✓	✓	✓	✓
Parameter tuning	×	×	✓	×	×	✓
Single layer propagation	✓	✓	✓	✓	✓	✓
Multi layer propagation	×	×	×	×	×	✓
Diffusion control	×	×	✓	×	×	✓

Table 2
Qualitative comparison between different state-of-the-art label propagation algorithms for the node classification task.

2.2. Label Propagation Combined with Other Methods

Recently, LP has been addressed from the game theory perspective known as *Graph Transduction Game (GTG)* [47]. In the framework, the transduction problem is formulated in terms of a non-cooperative multiplayer game whereby equilibria correspond to the consistent labeling of the data. GTG approach is also applied in challenging bioinformatics problem [48] known as protein function prediction beating state-of-the-art graph-based methods. Recently, the domain adaptation branch of transfer learning in combination with label propagation has shown promising results in visual recognition [49] in an unsupervised setting. It demonstrates that the LP can easily integrate with other methods to improve predictive accuracy.

The boom of neural networks [50] has inspired its application in graph structured data. One promising technique, known as graph convolutional networks (GCNs) [51], has achieved impressive node classification performance. Similarly, the graph neural networks (GNNs) [52, 53] have demonstrated high competitiveness in classifying node labels. Most GNN models adopt a message passing strategy [54]: each node aggregates features from its neighbors and then performs a layer-wise projection function with a non-linear activation to combine the information. Thus, GNNs can exploit both graph structure and node feature information in their models. However, the combination of graph topology, node features, and projection matrices in GNNs leads to a complicated prediction mechanism and could not take full advantage of prior knowledge lying in the data [55]. For instance, homophily assumption adopted in label propagation methods represents structure-based prior and has been shown to be underused [56] in graph convolutional network (GCN) [51]. GCN integrates node attributes, node labels, and edges in model learning. However, GCNs fall apart to employ the label distribution in the graph structure. Moreover, the dependency on labels as supervision limits the label update efficiency and prohibits a minimal label rate scenario, e.g., only one labeled example per class. LP, in combination with Convolutional Neural Network (CNN), has gathered significant interest in classification task [57]. Similarly, the study by Douze et al. [58] performed LP on a large image dataset with CNN descriptors for a few shot learning having state-of-the-art accuracy.

2.3. Multilayer Network Analysis

In the data mining and machine learning community, the notion of a "multilayer network" is used in various tasks, including community detection, node classification, and link prediction. With the recent increase in works using graph neural network embedding-based methods, such tasks tend to be carried out by learning a low dimensional representation of nodes in the network, preserving the network structure [59]. The node embedding techniques [60] have demonstrated high accuracy in link prediction and node classification tasks. Most of the embedding methods employ learned embeddings from the nodes in a multilayer network, and train supervised classifiers for node classification [8]. However, they do not use the manifold structure exhibited by embedding structures, a promising method for semi-supervised node classification using graph-based diffusion for data with a low ratio of labeled to unlabeled nodes.

The existing studies of random walks adopt a global strategy for navigation in a multilayer network [61]. In most of the works in label propagation in multilayer network analysis, network aggregation is performed, to aggregate data from the different layers of a multilayer network into a single layer by averaging their adjacency matrices. To generate a weight matrix $W = \frac{1}{k}\sum_{i=1}^{k}[W_{ijk}]$, where k is the number of layers for a single-layer network. Following this strategy, there is a loss of information from the original multiple layers. Also, in Multilayer networks, the inter-layer

edges disappear due to aggregation processes because self-edges (nodes having links to themselves) cannot account for information propagation to other nodes [62].

Network aggregation and long-range diffusion are the approaches taken for node classification in multilayer networks. However, long-range diffusion on a global scale stresses random walks that explore more of the multilayer network, leading to node misclassification. Another problem is the loss of layer information by aggregation. Thus in this work, we propose a general solution for multilayer network analysis from the perspective of network embedding to transform it into a homogeneous graph and control the diffusion range by a novel label propagation algorithm. Our algorithm provides closed-form solutions, guarantees convergence, and adapts the diffusion range to be suitable for various node classification problems.

3. Problem Formulation

This section details terms and introduces the node classification problem in a multilayer network. Suppose $\mathcal{N}$ is the list of nodes and there are K different types of layers which are expressed as $G_1, \dots, G_K$. For every layered network $G_k = (\mathcal{N}_k, E_k)$, we have $E_k \subseteq \mathcal{N}_k \times \mathcal{N}_k$, and $\mathcal{N}_k \subseteq \mathcal{N}$. The set of nodes is composed of two types of components $\mathcal{N} = L \cup U$ where $L = \{n_1, \dots, n_l\}$ is a set of l labeled nodes and $U = \{n_{l+1}, \dots, n_{l+u}\}$ is the list of unlabeled nodes.

Given a set C of c possible labels, let $f = [f_{ip}]$ be a matrix, where $f_{ip} = 1$ if node $i \in \mathcal{N}$ has label $p \in C$, and 0 otherwise. We can observe f_L, a part of f, where nodes are restricted to the set L. The problem is to predict the other part f_U of f, where nodes are restricted to the set U. Thus the **node classification problem** in multilayer networks is expressed as follows:

- **Input:** A partially labeled multilayer network. That is, $G_1, \dots, G_K$, and f_L.
- **Transform:** Encode the feature of nodes from multilayer network using graph embedding method and apply kNN method to construct the homogeneous graph.
- **Score:** Find a score S_{ip} for each unlabeled node i and each $p \in C$. A good method will have the property that larger values for S_{ip} imply more probably that i takes the label p. In an evaluation, the precision can be produced based on such scores.
- **Decision:** Assess through checking whether the classification function $\arg\max_p S_{ip}$ results in the same label as the ground truth, and produce accuracy in the evaluation.

4. Solution Approach

The overall solution of our approach is illustrated in Figure 2. The main aim of this study is to predict the labels of the nodes in a multilayer graph. First of all, the primary input is the multilayer graph with few labeled and large unlabeled nodes. The red and blue are the labeled nodes, and black is the unlabeled nodes in the multilayer graph. Next, we extracted the node embeddings of a multilayer graph using the tensor factorization methods. Once we have the embeddings of the multilayer graph, then we construct a surrogate homogenous graph from the embedding using the kNN approach. Finally, we used boundary heat diffusion to propagate the heat in the kNN graph using the labeled nodes as the heat source. Once the propagation process is over, we will have the final output, the label for the unlabelled nodes.

5. Methodology

5.1. Heat Diffusion (HD)

For a known graph structure, the heat flow with initial conditions can be defined by the following second order differential equation $\frac{\partial f(x,t)}{\partial t} - \Delta f(x,t) = 0$, where $f(x,t)$ is the heat at location x at time t, and Δf is the Laplace-Beltrami operator on a function f. The heat diffusion kernel $K_t(x, y)$ is a special solution to the heat equation with a special initial condition which is a unit heat source at position y when there is no heat in another end. Heat kernels [10] have proven useful because of the physical interpretation of the optimization in label propagation in a semi-supervised machine learning process [11]. The solution to the heat diffusion equation on a graph is [29] given as $f(t) = e^{-\alpha t \Delta} f(0)$. The value $f(t)$ illustrates the heat at node v at time t, beginning from an initial distribution of heat given by $f(0)$ at time zero and Δ is the graph Laplacian, and α is the diffusion coefficient.

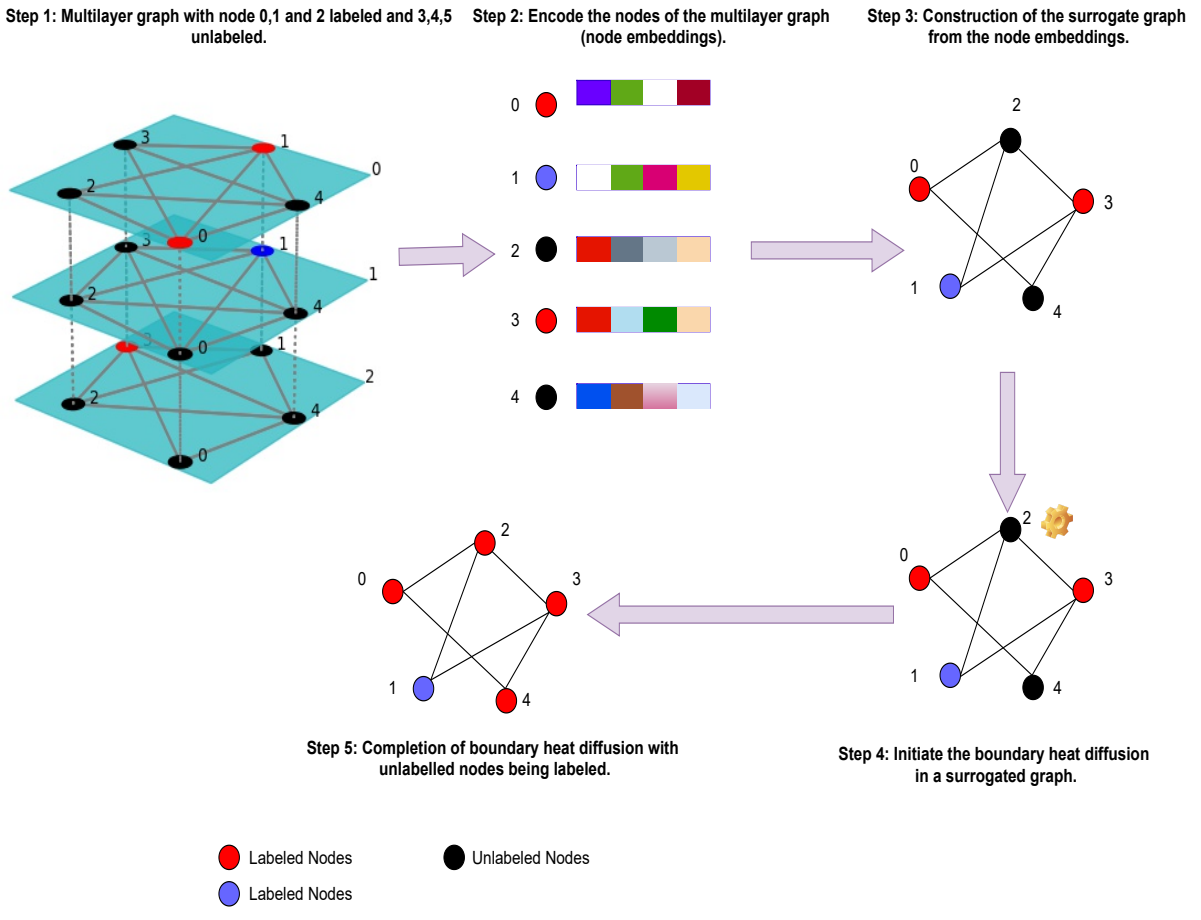


Figure 2: An illustration of applying boundary heat diffusion algorithm in a node classification problem for a multilayer network.

The exponential kernel is defined as:

$$e^{-\alpha\Delta} = \lim_{N\to\infty}\left(1 + \frac{-\alpha\Delta}{N}\right)^N \tag{1}$$

Yang et al. [63] proposed a discrete approximation to compute the heat diffusion in a graph. Using, this method heat diffusion can be computed iteratively,

$$f(1) = f(0)\left(I + \frac{-\alpha}{N}\Delta\right)^N \tag{2}$$

Where $f(0)$ is the initial heat scores of the nodes and $f(1)$ is the final heat scores after the diffusion process. In a practical setting, keeping the value of $\alpha = 1$ and $N = 30$ works in most of the cases [63]. The $f(1)$ value is used for the node label classification.

5.2. Heat Diffusion with a Boundary Condition in Graph (BHD)

To make it self-contained, we will briefly introduce the BHD model for node classification problem which is adapted from our previous work [45] that has been applied for graph-based regression problem. Let us suppose that there are l labeled and u unlabeled nodes and $N = l + u$ be the total nodes in the graph. Then $L = \{1, 2, ..., l\}$

corresponds to labeled nodes with labels $f_1, ..., f_l$, and nodes $U = \{l+1, l+2, ..., l+u\}$ refers to the unlabeled points. Our job here is to assign the labels for the nodes U. The edge of the graph is a $n \times n$ weight matrix W also known as adjacency matrix.

To formulate our model, let us assume that, at time t, each node $i \in U$ receives a certain amount of heat $M(i, j, t, \Delta t)$ from its neighbor j during a period of Δt. The heat $M(i, j, t, \Delta t)$ is proportional to the time Δt and the heat difference $f_j(t)$ - $f_i(t)$. Due to this, the heat difference at node i between time $t + \Delta t$ and time t will be equal to the sum of the heat that it receives from all of its neighbors. It is expressed as:

$$f_i(t + \Delta t) - f_i(t) = \sum_{j=1}^{n} (f_j(t) - f_i(t)) W_{ij} \Delta t \tag{3}$$

Dividing Eq. 3 by Δt on both sides, and let $\Delta t \to 0$, we have

$$\frac{df_i}{dt} = W_{i,:} f - d_i f_i \tag{4}$$

In terms of matrix operations, we split the weight matrix W also known as adjacency matrix of graph into 4 blocks after the Lth row and column:

$$W = \begin{bmatrix} W_{LL} & W_{LU} \\ W_{UL} & W_{UU} \end{bmatrix} \tag{5}$$

Note that $W_{U,:} f = \begin{bmatrix} W_{UL} & W_{UU} \end{bmatrix} \begin{bmatrix} f_L \\ f_U \end{bmatrix}$, and $\Delta_{UU} = D_{UU} - W_{UU}$. Here Δ is the combinatorial Laplacian which is given in the matrix form as $\Delta = D - W$ where $D = diag(d_i)$ also known as degree matrix of the graph. The $diag(d_i)$ is the diagonal matrix with entries $d_i = \sum_j w_{ij}$ and $W = [w_{ij}]$.
We have a matrix form:

$$\begin{aligned} \frac{df_U}{dt} &= W_{U,:} f - D_{UU} f_U \\ &= W_{UL} f_L + W_{UU} f_U - D_{UU} f_U \\ &= W_{UL} f_L - \Delta_{UU} f_U \end{aligned} \tag{6}$$

Solving this linear differential equation which is the form of $dy/dx + Py = Q$ to find the closed form solution we have:

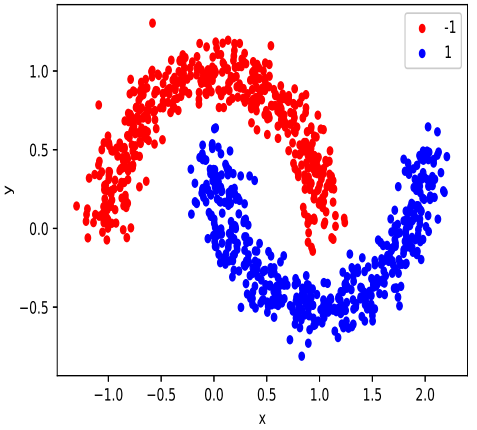
$$\frac{df_U}{dt} = W_{UL} f_L - \Delta_{UU} f_U \tag{7}$$

Here $P = \Delta_{UU}$ and $Q = W_{UL} f_L$

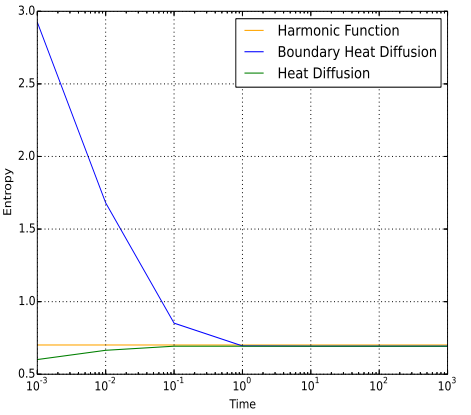
$$f_U = \Delta_{UU}^{-1} W_{UL} f_L + e^{-\Delta_{UU} t} C \tag{8}$$

This is the temperature distribution on the unlabeled nodes at time t, given the boundary condition f_L. This function is used to predict the labels for the unlabeled node. Given the initial condition $f_U\big|_{t=0} = f_U(0)$, $C = f_U(0) - \Delta_{UU}^{-1} W_{UL} f_L$. Note that, in the limit $t \to +\infty$, $\mathrm{f}_U = \Delta_{UU}^{-1} W_{UL} f_L$, which is the harmonic function.

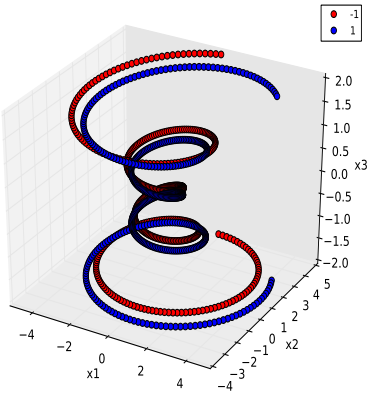
To interpret Eq. 8 and the heat diffusion with the boundary process more intuitively, we constructed two different toy classification datasets (i) two moon-shaped simulated data from 1000 points with a standard deviation of 0.1 using two features and (ii) spiral inter-wined data from 1000 points with a standard deviation of 0.1 using three features. The shaped of the dataset is shown in the Figure 3 [a] and [c]. The red data point has label -1, and the blue data point



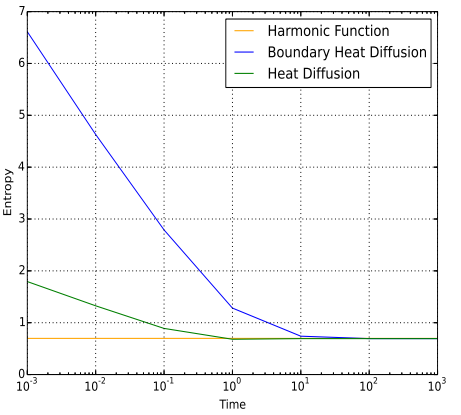
(a) 2 Moon-Shaped simulated Data.



(b) Error Curve 1.



(c) Spiral-Shaped simulated Data.



(d) Error Curve 2.

Figure 3: The error curve of different label propagation algorithms in a toy datasets.

has label +1 in both the figures. There are two visibly separate clusters in the data. We employed the Gaussian RBF Kernel $w_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$ to construct the graph between these points and randomly choose 2 points from each of the labels $[-1, 1]$ and the rest of the points as unlabeled and applied the closed-form equations for heat diffusion, harmonic function, and boundary heat diffusion. Figure 3 [b] and [d] shows the performance of these algorithms. The y-axis is the cross-entropy loss, and the x-axis is the time. The harmonic function does not have the time component in its equation, but HD and BHD have the time component. We can see from the curve that when time equals 10^{-3}, the BHD algorithm has the highest cross-entropy loss in both moon and spiral-shaped data. As time increases, BHD started to have a low cross-entropy loss. BHD starts to converge as time equals 10^0 in the moon-shaped data and time equals 10^2 in the spiral-shaped data. It means that BHD will be the same as harmonic function favoring continuous or long-range propagation for a higher value of time.

5.3. Computational Complexity

In the solution provided by Eq. 8 we have two parts: (i) the harmonic part and (ii) the exponential part. When the graph is large, the computation will be time-consuming because both terms have a $O(n^3)$ complexity. To solve this, we took an iterative approach to compute the harmonic part provided by Zhu et al. [64], which is the same as

a random walk with restart (RWR). For the exponential part we took the discrete approximations by Yang et.al [63]: $f(t) = \left(I - \frac{t}{M}\Delta_{UU}\right)^M f(0)$, where M is the number of iterations chosen as $M = 30$, same as [63], and I is the identity matrix. t is from the cross-validation in the training set ranging from $[10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$. $f(0)$ is the initial temperature and $f(t)$ is the temperature at timestamp t. Specifically, after the discrete formalization of the complexity of exponential kernel in our model is given by $O(M|E|n)$ where M is the number of iterations, n is the number nodes and $|E|$ is the number of edges in the graph. However, we used kNN graphs where each node connects to only a few nodes; we can reduce the complexity of discrete approximation. The label propagation in such sparse graphs is computationally cheaper and faster. The time complexity, in this case, depends on k being chosen. The optimum k can be chosen by using the cross-validation in training sets or domain knowledge of the data. For a small k, the graph will be sparse, which ultimately speeds up the computation time.

5.4. Space Complexity

For a fully connected graph we need to store $|E|$ number of edges and n is the length of vectors for initial temperatures that means the space complexity (S) is: $S = O(|E|) + O(n) = O(n^2) + O(n)$. However, we are using a kNN approach to build the graph from the embedding vectors; then, we can reduce the space complexity. If k is chosen small then we can reduce the space complexity from $O(n^2) + O(n)$ to $O(kn) + O(n)$.

5.5. Initial Temperature Setting

We set the initial temperature at time zero for the labeled nodes as 1. However, if network contains many false positive links, then the ideal way to make inferences about the initial value for each node in U is the mean values in Y_L. It is because we can safely assume that the value of the node appears as independent random variables from the same population [65]. Thus the best guess to initialize the initial temperature from the population mean μ, is to use the sample mean of Y_L.

6. Algorithm

This algorithm requires a $n \times n$ transition matrix, a $n \times c$ label matrix where c is the number of labels, a $n \times n$ Laplacian matrix Δ. M is the number of iterations. Once we calculate the harmonic score, we need to calculate the constant C, as shown in Equation 8. C is obtained by subtracting an initial label matrix from a harmonic score. The initial label matrix has an initial temperature for each node. We imputed the values for the unlabeled nodes as the means of the labeled nodes. This C is the initial condition of state matrix $(n \times c)$ for heat diffusion with boundary conditions. Formally, the algorithm is described in Algorithm 1

6.1. Parameter t

Parameter t has a vital role in the diffusion process. If t has a high value, heat will diffuse very quickly. In Eq. 8, if t tends to $+\infty$, then the heat diffusion with boundary condition will become a harmonic function. It means the heat will travel deeper into the graph, i.e., it will follow a long-range or global diffusion. If t is small, then heat will diffuse slowly, favoring short-range or local diffusion. Different networks require different values of t. For instance, rumors or fake news propagate faster in a social network than true stories [66]. In that case, t is high because heat immediately transfers to the rest of the neighbors, making the diffusion process faster.

7. Experiments

In this experiment[2], we answer the following questions:

- Q1: *Classes:* Does varying classes by keeping same structural properties of the graph will impact the ability of BHD in a node classification task ?
- Q2: *Accuracy:* Which graph embedding will have highest accuracy in synergy with BHD in a multilayer network?
- Q3: *Parameter:* Does the parameter t affect the performance of BHD in a multilayer network?

[2]The code and the datasets are publicly available in the Github repository https://github.com/timilsinamohan/BHDClassifier.

Algorithm 1: Heat Diffusion with Boundary Condition

Input : The transition matrix T of size $n \times n$; initial label matrix Y of size $n \times c$; Laplacian matrix Δ; M is the number of iteration chosen as 30; I is the identity matrix of size $n \times n$

Output: State matrix of size $n \times c$

Initialize U = Y
repeat
  $Y^{k+1} \leftarrow TY^k$
  Row Normalize: Y^{k+1}
  $Y^{k+1} \leftarrow Y^{k+1} + U$
  $Y^k = Y^{k+1}$
  $k = k + 1$
until error between Y^{k+1} and Y^k becomes sufficiently small
Initial_Temperature: Impute mean value for unlabeled nodes using labeled value from column of matrix U
C = Initial_temperature - Y^K
State_Matrix = C
t is a parameter in (10^{-3},10^3);
for $b = 1$ *to* M **do**
  $State_Matrix = Y^K + \left(I - \frac{t}{M}\Delta\right) State_Matrix$
end
  Row Normalize: $State_Matrix$
return $State_Matrix$

Synthetic Datasets To generate the synthetic graph, we used **Stochastic Block Model (SBM)**. SBM is a generative model for random graphs which generates graphs containing clusters. The same approach is also used to assess the performance of the Semi-Supervised Node Classification by graph-based approach [67]. In this experiment, we examined a SBM with $n = 2000$ nodes and k classes to test the BHD. It is assumed that the actual label for each node is sampled uniformly from the set $\{0, ..., k-1\}$. The two nodes falls under the same class then an edge is drawn between them with probability p; else, these nodes are connected to each other with probability q. Table 3 provides the properties of the SBM used in our dataset.

Synthetic Dataset	Nodes	Labels	p	q
k-SBM	n = 2000	$k \in \{3, 4, 5\}$	$5 \times \frac{logn}{n}$	$1 \times \frac{logn}{n}$

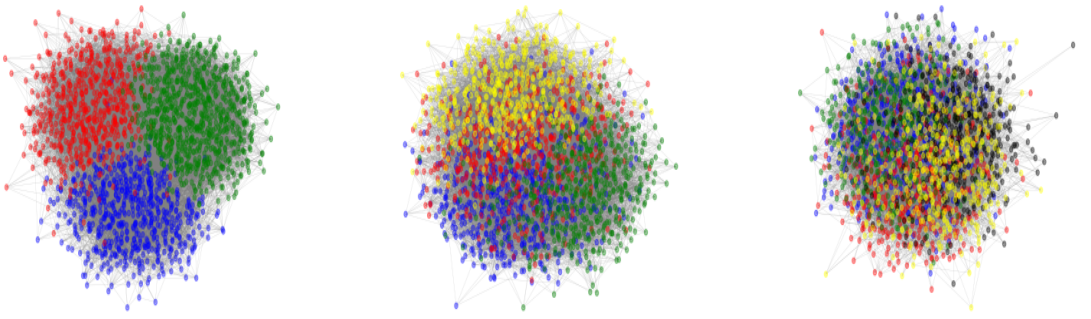
Table 3
The properties of the synthetic dataset for the semi-supervised node classification.

Figure 4 demonstrates three types of the graphs generated from a stochastic block model and visualized using NetworkX[3] graph package using the parameters in Table 3.

Multilayer Network Datasets. To evaluate the effectiveness of BHD in a multilayer network, we used publicly available data for node classification. We use two bibliographic network datasets (DBLP[4] and ACM[4]), a physician network called CKM [68], a collaboration network called Leskovec_NG[5], and a movie dataset called IMDB[4]. The statistics of the multilayer networks used in our experiments are shown in Table 4.

The CKM multilayer network is about the impact of network ties on the physician's adoption of a new drug. There are three layers in this network, and the labels are the researchers associated with their original companies. Leskovec_NG contains the coauthors of Andrew Ng and Jure Leskovec at Stanford University from 1995 to 2014. This multilayer graph is a 4-layer temporal graph. For each layer, there is an edge between two researchers if they coauthored at least one paper in the 5-year interval. DBLP data contains three types of nodes (papers, authors, conferences), and

[3]https://networkx.org/documentation/stable/reference/generated/networkx.generators.community.stochastic_block_model.html
[4]https://github.com/THUDM/cogdl
[5]https://sites.google.com/site/pinyuchenpage/datasets



(a) $k = 3$. (b) $k = 4$. (c) $k = 5$.

Figure 4: Three realizations of k-SBM with 2000 nodes and different number of classes.

Datasets	#Layers	#Nodes	#Edges	#Classes
CKM Physician Datasets	3	246	1,551	3
Leskovec_NG Datasets	3	191	1,836	2
DBLP	4	18,405	67946	4
ACM	4	8,994	25922	3
IMDB	4	12,772	37288	3

Table 4
Multilayer Network Datasets.

four types of layers and research areas of authors are as labels. ACM contains three types of nodes (papers, authors, subject), four types of layers and categories of papers are the labels. IMDB contains three types of nodes (movies, actors, and directors), and labels are genres of movies.

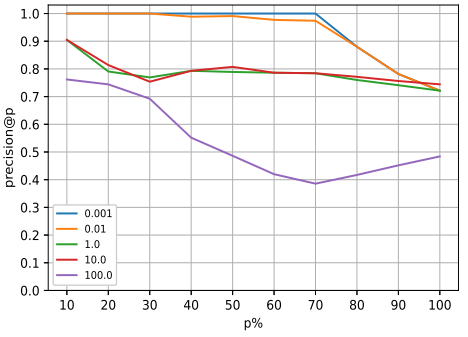
Evaluation: We hide 90% of labeled nodes in CKM Physician and Leskovec_NG Datasets and perform a ten-fold cross-validation due to its small size. For DBLP ACM, and IMDB, we use the same train, test, and validation sets provided in the benchmark datasets. The numbers of train, validation and test nodes are (800, 400, 2857) for DBLP, (600, 300, 2125) for ACM and (300, 300, 2339) for IMDB respectively. Then we applied the algorithms to infer the hidden labels. We reported (i) precision@p and (ii) accuracy metric because both of these metrics assess the performance of the label propagation algorithm. Precision@p is the precision of top p% nodes ordered by their maximum score of $max_j S_{ij}$. The accuracy score computes the subset accuracy in a multilabel classification task.

Q1:Class We applied the BHD in the three different class settings using only a 10% labeled dataset. To assess the performance of a BHD classification model, we used cross-entropy loss. It is because this metric measures the distance between the two probability distributions of predicted value and the actual label [69] and is regarded as the measure of quality of predictions rather than the accuracy of the classifier [70, 71]. A perfect classifier has the cross-entropy score 0. It is computed as:

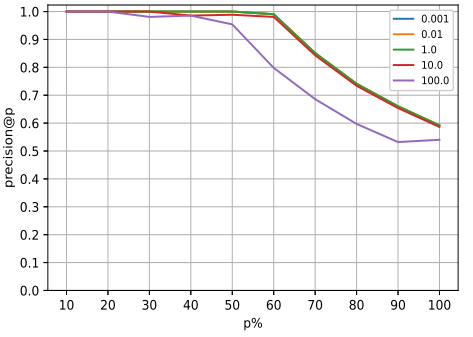
$$\text{Cross-entropy} = -\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j}\log(p_{i,j}) \tag{9}$$

where, $y_{i,j}$ denotes the true value i.e. 1 if data point i belongs to class j and 0 otherwise and $p_{i,j}$ denotes the probability predicted by the model of data point i belonging to class j.

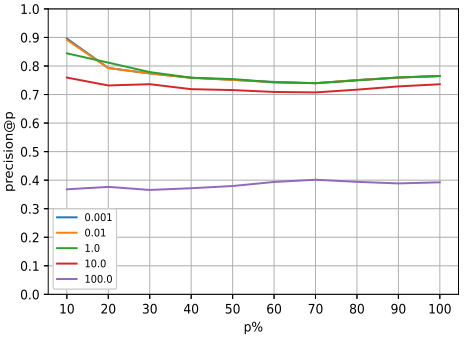
For computing cross-entropy, we vary the time in the range of $[10^{-3}, 10^{3}]$. As the time increases, the cross entropy starts to decrease shown in Figure 6 (a). A similar trend is observed for all three classes. At time = 10^0, we observed the minimum cross-entropy. After the time exceeds 10^0, the three curve starts to increase. By time = 10^1, the curves start to flatten, meaning no further reduction in cross entropy, and the algorithm is converged. As the number of classes increased, fixing the same number of nodes, we observed the cross-entropy increment. It is because the number of



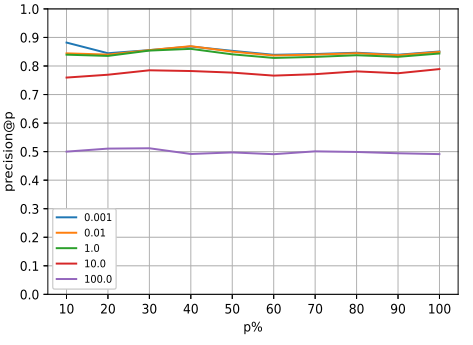
(a) CKM Physician.



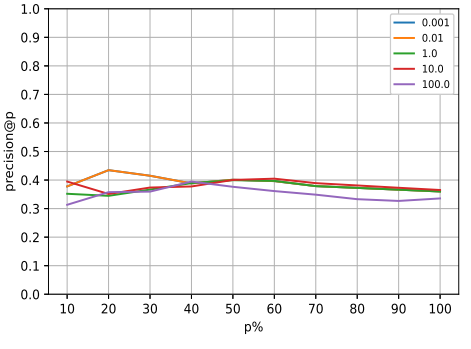
(b) NG-Lesckovec lab.



(c) ACM.



(d) DBLP.

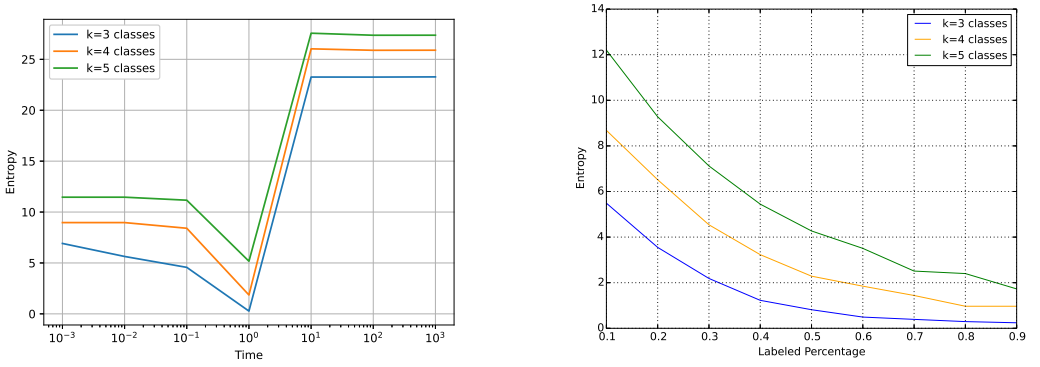


(e) IMDB.

Figure 5: Impact of parameter t in multilayer networks. X-axis is the percentage of data ranked. Y-axis is the Precision at each percentage of the data.

labeled samples in each class is identical; however, the label distribution is different, which affects the cross entropy score. We also observed that when higher time is chosen the model starts to perform poorly giving the unreliable predictions. Thus, choosing optimal t is very important for accurate prediction.

Similarly, in Figure: 6 (b), we used different percentages of the labeled data in training sets ranging from 10% to 90%. The optimum time is estimated from the training set in cross-validation. We saw that as the percentage of a labeled sample increases, the entropy decreases for all the classes. It is because as the training set increases, the BHD is able to capture more neighborhood information and enhance the prediction performance. Thus we observe the reduction of the cross-entropy score.



(a) Time versus Entropy

(b) % of labeled training nodes versus Entropy

Figure 6: Label Propagation using BHD in a SBM Graph in a 3 different label settings.

Q2:Accuracy We set the embedding dimension to 150 for all node embedding methods for a fair comparison. We employed the kNN methods to construct the node similarity graph where k is chosen from the cross-validation of training sets. We have used state-of-the-art node embedding methods Node2vec, Relational Graph Convolution Neural network (RGCN) [72], tensor factorization model known as TuckER [73], ComplEx [74] and RESCAL [75]. For the kNN graphs constructed from node embedding, we employed the label propagation algorithm HMN, LGC, CAMLP, OMNI, HD, and BHD.

	IMDB			DBLP			ACM		
	Precision@10	Precision@100	Accuracy	Precision@10	Precision@100	Accuracy	Precision@10	Precision@100	Accuracy
Tucker + HMN	0.25	0.28	0.21	0.53	0.49	0.28	0.72	0.70	0.64
Tucker + LGC	0.35	0.33	0.32	0.42	0.42	0.19	0.73	0.71	0.66
Tucker + CAMLP	0.30	0.33	0.32	0.58	0.44	0.21	0.70	0.71	0.66
Tucker + OMNI	0.31	0.30	0.26	0.62	0.45	0.24	0.63	0.61	0.54
Tucker + HD	0.33	0.33	0.32	0.69	0.42	0.19	0.70	0.71	0.66
Tucker + BHD	0.45	0.38	0.59	0.90	0.91	0.89	0.85	0.75	0.69
ComplEx + HMN	0.29	0.27	0.25	0.54	0.58	0.34	0.71	0.69	0.66
ComplEx + LGC	0.38	0.32	0.31	0.53	0.54	0.27	0.74	0.72	0.62
ComplEx + CAMLP	0.31	0.30	0.32	0.57	0.55	0.32	0.71	0.70	0.62
ComplEx + OMNI	0.32	0.30	0.28	0.66	0.49	0.34	0.65	0.62	0.60
ComplEx + HD	0.31	0.35	0.30	0.70	0.56	0.28	0.72	0.70	0.64
ComplEx + BHD	0.39	0.37	0.35	0.71	0.57	0.36	0.72	0.72	0.67
Node2Vec + HMN	0.59	0.23	0.16	0.48	0.46	0.24	0.55	0.61	0.44
Node2Vec + LGC	0.59	0.23	0.16	0.43	0.46	0.24	0.57	0.61	0.44
Node2Vec + CAMLP	0.58	0.23	0.16	0.68	0.45	0.23	0.55	0.61	0.44
Node2Vec + OMNI	0.60	0.23	0.15	0.64	0.45	0.24	0.39	0.57	0.38
Node2Vec + HD	0.60	0.23	0.16	0.71	0.46	0.24	0.57	0.61	0.44
Node2Vec + BHD	0.61	0.35	0.36	0.72	0.47	0.28	0.60	0.61	0.65
RGCN + HMN	0.47	0.23	0.20	0.44	0.46	0.25	0.84	0.74	0.67
RGCN + LGC	0.46	0.23	0.20	0.43	0.46	0.25	0.81	0.72	0.66
RGCN + CAMLP	0.46	0.23	0.20	0.40	0.45	0.24	0.82	0.73	0.66
RGCN + OMNI	0.30	0.32	0.40	0.46	0.47	0.27	0.84	0.75	0.67
RGCN + HD	0.46	0.23	0.20	0.41	0.47	0.25	0.74	0.73	0.67
RGCN + BHD	0.47	0.34	0.41	0.47	0.48	0.28	0.84	0.75	0.68
RESCAL + HMN	0.53	0.23	0.17	0.91	0.27	0.76	0.85	0.76	0.71
RESCAL + LGC	0.53	0.23	0.18	0.91	0.27	0.76	0.86	0.76	0.71
RESCAL + CAMLP	0.53	0.23	0.17	0.90	0.27	0.8	0.83	0.76	0.71
RESCAL + OMNI	0.58	0.23	0.16	0.92	0.27	0.89	0.86	0.77	0.71
RESCAL + HD	0.53	0.23	0.18	0.87	0.27	0.76	0.80	0.76	0.71
RESCAL + BHD	**0.62**	**0.58**	**0.61**	0.92	**0.93**	**0.90**	**0.89**	**0.77**	**0.72**

Table 5
Result evaluation of the embedding models in a node classification task. Best results are in boldface.

In Table 5 and 6, we can see that BHD with RESCAL node embedding outperforms most multilayer networks. Our method (RESCAL + BHD) does not use any external node feature but only relies on the graph structure and

	Leskovec_NG			CKM Physician		
	Precision@10	Precision@100	Accuracy	Precision@10	Precision@100	Accuracy
Tucker + HMN	0.98 ± 0.001	0.71 ± 0.150	0.97 ± 0.007	0.47 ± 0.407	0.57 ± 0.188	0.57 ± 0.195
Tucker + LGC	0.97 ± 0.021	0.70 ± 0.014	0.96 ± 0.002	0.50 ± 0.416	0.58 ±0.191	0.57 ±0.198
Tucker + CAMLP	0.97 ± 0.035	0.71 ± 0.148	0.95 ± 0.009	0.67 ± 0.212	0.57 ± 0.183	0.56 ± 0.193
Tucker + OMNI	0.98 ± 0.087	0.71 ± 0.149	0.97 ± 0.009	0.67 ± 0.217	0.57 ± 0.188	0.57 ± 0.197
Tucker + HD	0.97 ± 0.071	0.71 ± 0.145	0.99 ± 0.011	0.67 ± 0.220	0.56 ± 0.178	0.55 ± 0.188
Tucker + BHD	0.99 ± 0.001	0.71 ± 0.150	0.99 ± 0.007	0.87 ± 0.296	0.73 ± 0.189	0.92 ± 0.224
ComplEx + HMN	0.98 ± 0.011	0.71 ± 0.002	0.96 ± 0.114	0.46 ± 0.101	0.57 ± 0.244	0.55 ± 0.854
ComplEx + LGC	0.98 ± 0.478	0.70 ± 0.415	0.95 ± 0.122	0.49 ± 0.208	0.57 ±0.485	0.54 ± 0.744
ComplEx + CAMLP	0.97 ± 0.141	0.70 ± 0.185	0.96 ± 0.148	0.66 ± 0.105	0.56 ± 0.119	0.53 ± 0.254
ComplEx + OMNI	0.98 ± 0.987	0.70 ± 0.214	0.96 ± 0.854	0.67 ± 0.117	0.57 ± 0.278	0.71 ± 0.258
ComplEx + HD	0.96 ± 0.062	0.71 ± 0.874	0.98 ± 0.211	0.68 ± 0.118	0.56 ± 0.281	0.57 ±0.112
ComplEx + BHD	0.99 ± 0.210	0.70 ± 0.241	0.99 ± 0.227	0.81 ± 0.214	0.56 ± 0.125	0.60 ± 0.148
Node2Vec + HMN	0.59 ±0.187	0.55 ±0.073	0.55 ±0.073	0.59 ±0.401	0.54 ±0.203	0.61 ±0.206
Node2Vec + LGC	0.59 ±0.207	0.53 ± 0.098	0.53 ±0.041	0.48 ±0.388	0.52 ±0.197	0.51 ±0.202
Node2Vec + CAMLP	0.61 ±0.217	0.54 ± 0.101	0.53 ±0.079	0.62 ±0.204	0.51 ±0.166	0.57 ± 0.204
Node2Vec + OMNI	0.61 ±0.218	0.55 ±0.093	0.54 ±0.065	0.68 ±0.204	0.54 ± 0.167	0.53 ± 0.168
Node2Vec + HD	0.62 ±0.191	0.54 ±0.089	0.54 ±0.057	0.63 ± 0.258	0.52 ± 0.196	0.53 ±0.174
Node2Vec + BHD	0.63 ±0.201	0.62 ±0.161	0.80 ±0.163	0.68 ± 0.272	0.57 ±0.183	0.62 ±0.188
RGCN + HMN	0.60 ±0.339	0.55 ±0.075	0.58 ±0.095	0.58 ±0.290	0.54 ±0.185	0.54 ±0.201
RGCN + LGC	0.58 ±0.331	0.55 ± 0.072	0.59 ±0.096	0.60 ±0.291	0.54 ±0.186	0.53 ±0.201
RGCN + CAMLP	0.57 ±0.137	0.55 ±0.075	0.58 ±0.096	0.55 ±0.207	0.53 ±0.195	0.53 ± 0.211
RGCN + OMNI	0.56 ±0.092	0.58 ±0.059	0.65 ±0.061	0.63 ±0.113	0.51 ±0.039	0.60 ±0.035
RGCN + HD	0.75 ±0.078	0.55 ±0.062	0.58 ±0.094	0.54 ±0.158	0.54 ±0.187	0.53 ±0.201
RGCN + BHD	0.83 ±0.188	0.59 ±0.076	0.63 ±0.144	0.68 ±0.236	0.55 ±0.184	0.61 ±0.254
RESCAL + HMN	0.98 ±0.037	0.57 ± 0.071	0.64 ±0.075	0.63 ±0.338	0.56 ±0.199	0.57 ±0.191
RESCAL + LGC	0.98 ±0.037	0.56 ±0.068	0.66 ±0.107	0.60 ±0.351	0.56 ±0.197	0.58 ±0.194
RESCAL + CAMLP	0.97 ± 0.039	0.56 ±0.081	0.69 ±0.101	0.71 ±0.131	0.56 ± 0.201	0.58 ±0.191
RESCAL + OMNI	0.97 ±0.054	0.60 ± 0.082	0.74 ±0.149	0.77 ±0.141	0.48 ±0.185	0.52 ±0.213
RESCAL + HD	0.97 ±0.039	0.57 ±0.071	0.66 ±0.109	0.70 ±0.110	0.55 ±0.196	0.57 ±0.188
RESCAL + BHD	**0.99 ±0.017**	**0.72 ±0.158**	**0.99 ±0.003**	**0.88 ±0.157**	**0.75 ±0.206**	**0.94 ±0.114**

Table 6
Result evaluation of the embedding models in a node classification task. Best results are in boldface.

still provides competitive results compared to other methods. RESCAL uses the collective matrix factorization (CMF) method, a powerful technique to learn shared latent representations from multiple matrices. As the node in a multilayer network is shared across layers, RESCAL is ideal for extracting the node representation in such a setting. Another strength of RESCAL is that it computes a global latent-component representation of the nodes, making it ideal for retrieving similar nodes in a multilayer network. Therefore graph constructed using these embeddings compliments label propagation algorithms, which works on the assumption that similar unlabeled nodes should be given the same classification.

Another important observation is that combining the node embeddings for multilayer networks and BHD label propagation has superior performances to other label propagation algorithms, especially in Precision@10 and Accuracy. For example, in Table 5 and 6, we can see that synergy between Tucker+BHD is better than Tucker+HMN, Tucker+CAMLP, Tucker+OMNI, and Tucker+HD. A similar trend is observed for ComplEx+BHD, Node2Vec+BHD, RGCN+BHD, and RESCAL+BHD. It suggests that BHD style label propagation improves the node classification in a multilayer network than other label propagation algorithms.

Q3: Parameter In Table 5 and 6, we observed RESCAL+ BHD has better performance than other combination. In this experiment, we wanted to assess the node classification ability of our combined approach of RESCAL embedding with BHD by varying the parameter t. The parameter t is varied from 10^{-3} to 10^{2}. We report the result of all the multilayer network data used in our studies. From Fig 5, we see that the largest values of the parameter t lead to lower precision than the smallest t. The values of t less than or equal to 1 lead to high precision for small p and perform better than large t for large p. In all the data, we saw that setting t at 100 led to the precision quickly dropping. As t

increases, heat will quickly transfer over all the nodes of the network leading to miss-classification. We also see that setting a small t performed better than a large t. Thus, in these multilayer networks, node classification favors small values of t. It also means that in this kind of network, short-range diffusion is supported.

8. Conclusion

We presented a novel heat diffusion method with the boundary condition, which addresses the node classification problems in multilayer networks. This model requires a parameter t for time, which controls the range of propagation in the network. The advantages of our algorithm are:

1. Accuracy: It outperforms or equals the state-of-the-art algorithms in label propagation for node classification in multilayer networks: Table 5 and 6,.
2. Linear: Our algorithm has a closed-form solution that can be evaluated in a finite number of steps: Equation 8 and Algorithm 1.
3. Parameter: It has only one parameter t which controls the heat flow for long or short range diffusion: Figures 3 and 5.

We believe that our boundary-based heat diffusion method is simple but effective for node classification. The limitation of the study is that we have a free parameter t, which we estimated from cross-validation mode in the training set. For this, the extra computational time is required to find the optimum t. To save computational time, we intended to solve this by learning the parameter based on the network structure by automatically determining the value for t, which we consider for our future work. We used the simple kNN method, which provides us with a surrogate graph from multilayer network embeddings for label propagation. Of course, better graphs can be constructed if one can define better distance functions, connectivity, and edge weights. However, constructing efficient graphs from the features is another critical challenge in a graph-based semi-supervised classification problem.

CRediT authorship contribution statement

Mohan Timilsina: M.T conducted experiments, analysed the results and prepared the original draft.. **Vít Nováček:** V.N provided the guidance and revised the manuscript. **Mathieu d'Aquin:** M.A provided the guidance and revised the manuscript.. **Haixuan Yang:** H.Y provided the guidance and revised the manuscript..

Acknowledgement

We would like to acknowledge Science Foundation Ireland (SFI/12/RC/2289_P2) for funding this research.

References

[1] C. Buono, L. G. Alvarez-Zuzek, P. A. Macri, L. A. Braunstein, Epidemics in partially overlapped multiplex networks, PloS one 9 (3) (2014) e92200.
[2] J. Gao, S. V. Buldyrev, S. Havlin, H. E. Stanley, Robustness of a network of networks, Physical Review Letters 107 (19) (2011) 195701.
[3] Z. Hammoud, F. Kramer, Multilayer networks: aspects, implementations, and application in biomedicine, Big Data Analytics 5 (1) (2020) 1–18.
[4] A. C. Kinsley, G. Rossi, M. J. Silk, K. VanderWaal, Multilayer and multiplex networks: An introduction to their use in veterinary epidemiology, Frontiers in veterinary science 7 (2020) 596.
[5] S. Huang, Y. Wu, S. Gao, Data-driven clustering in ad-hoc networks based on community detection, in: Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers, 2021, pp. 631–636.
[6] F. Alimadadi, E. Khadangi, A. Bagheri, Community detection in facebook activity networks and presenting a new multilayer label propagation algorithm for community detection, International Journal of Modern Physics B 33 (10) (2019) 1950089.
[7] D. Malhotra, R. Goyal, Supervised-learning link prediction in single layer and multiplex networks, Machine Learning with Applications 6 (2021) 100086.
[8] H. Zhang, L. Qiu, L. Yi, Y. Song, Scalable multiplex network embedding., in: IJCAI, Vol. 18, 2018, pp. 3082–3088.
[9] S. Gomez, A. Diaz-Guilera, J. Gomez-Gardenes, C. J. Perez-Vicente, Y. Moreno, A. Arenas, Diffusion dynamics on multiplex networks, Physical review letters 110 (2) (2013) 028701.
[10] R. I. Kondor, J. Lafferty, Diffusion kernels on graphs and other discrete structures, in: Proceedings of the 19th international conference on machine learning, Vol. 2002, 2002, pp. 315–322.

[11] X. Zhu, Z. Ghahramani, J. D. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: Proceedings of the 20th International conference on Machine learning (ICML-03), 2003, pp. 912–919.
[12] X. Zhou, M.-C. J. Kao, W. H. Wong, Transitive functional annotation by shortest-path analysis of gene expression data, Proceedings of the National Academy of Sciences 99 (20) (2002) 12783–12788.
[13] H. Wang, Q. Li, G. D'Agostino, S. Havlin, H. E. Stanley, P. Van Mieghem, Effect of the interconnected network structure on the epidemic threshold, Physical Review E 88 (2) (2013) 022801.
[14] S. Fortunato, Community detection in graphs, Physics reports 486 (3-5) (2010) 75–174.
[15] H. Hu, Y. van Gennip, B. Hunter, A. L. Bertozzi, M. A. Porter, Multislice modularity optimization in community detection and image segmentation, in: 2012 IEEE 12th International Conference on Data Mining Workshops, IEEE, 2012, pp. 934–936.
[16] A. Browet, P.-A. Absil, P. Van Dooren, Community detection for hierarchical image segmentation, in: International Workshop on Combinatorial Image Analysis, Springer, 2011, pp. 358–371.
[17] B. Xu, H. Shen, Q. Cao, K. Cen, X. Cheng, Graph convolutional networks using heat kernel for semi-supervised learning, arXiv preprint arXiv:2007.16002.
[18] K. Kloster, D. F. Gleich, Heat kernel based community detection, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 1386–1395.
[19] L. Cowen, T. Ideker, B. J. Raphael, R. Sharan, Network propagation: a universal amplifier of genetic associations, Nature Reviews Genetics 18 (9) (2017) 551.
[20] J. Shrager, T. Hogg, B. A. Huberman, Observation of phase transitions in spreading activation networks, Science 236 (4805) (1987) 1092–1094.
[21] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: Bringing order to the web., Tech. rep., Stanford InfoLab (1999).
[22] J. M. Kleinberg, Authoritative sources in a hyperlinked environment, in: Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms, 1998, pp. 668–677.
[23] L. Lovász, et al., Random walks on graphs: A survey, Combinatorics, Paul erdos is eighty 2 (1) (1993) 1–46.
[24] H. Tong, C. Faloutsos, J.-Y. Pan, Random walk with restart: fast solutions and applications, Knowledge and Information Systems 14 (3) (2008) 327–346.
[25] P. L. Krapivsky, S. Redner, E. Ben-Naim, A kinetic view of statistical physics, Cambridge University Press, 2010.
[26] D. J. Klein, M. Randić, Resistance distance, Journal of mathematical chemistry 12 (1) (1993) 81–95.
[27] A. Blum, S. Chawla, Learning from labeled and unlabeled data using graph mincuts.
[28] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: Advances in neural information processing systems, 2004, pp. 321–328.
[29] H. Yang, M. R. Lyu, I. King, A volume-based heat-diffusion classifier, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (2) (2009) 417–430.
[30] M. Timilsina, M. Tandan, M. d'Aquin, H. Yang, Discovering links between side effects and drugs using a diffusion based method, Scientific reports 9 (1) (2019) 1–10.
[31] M. Timilsina, H. Yang, R. Sahay, D. Rebholz-Schuhmann, Predicting links between tumor samples and genes using 2-layered graph based diffusion approach, BMC bioinformatics 20 (1) (2019) 462.
[32] Y. Yamaguchi, C. Faloutsos, H. Kitagawa, Omni-prop: Seamless node classification on arbitrary label correlation, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[33] O. Chapelle, B. Schölkopf, A. Zien, Label propagation and quadratic criterion.
[34] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, Y. Yang, Learning to propagate labels: Transductive propagation network for few-shot learning, arXiv preprint arXiv:1805.10002.
[35] S. Ravi, Q. Diao, Large scale distributed semi-supervised learning using streaming approximation, in: Artificial intelligence and statistics, PMLR, 2016, pp. 519–528.
[36] J. Du, F. Zhu, E.-P. Lim, Dynamic label propagation in social networks, in: International Conference on Database Systems for Advanced Applications, Springer, 2013, pp. 194–209.
[37] M. Fahrbach, G. Goranci, R. Peng, S. Sachdeva, C. Wang, Faster graph embeddings via coarsening, in: International Conference on Machine Learning, PMLR, 2020, pp. 2953–2963.
[38] Y. Hu, S. Seneviratne, K. Thilakarathna, K. Fukuda, A. Seneviratne, Characterizing and detecting money laundering activities on the bitcoin network, arXiv preprint arXiv:1912.12060.
[39] M. Timilsina, H. Yang, D. Rebholz-Schuhmann, A 2-layered graph based diffusion approach for altmetric analysis, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2018, pp. 463–466.
[40] M. Timilsina, M. d'Aquin, H. Yang, Heat diffusion approach for scientific impact analysis in social media, Social Network Analysis and Mining 9 (1) (2019) 1–13.
[41] M. Timilsina, D. P. Mc Kernan, H. Yang, M. D'Aquin, Synergy between embedding and protein functional association networks for drug label prediction using harmonic function, IEEE/ACM Transactions on Computational Biology and Bioinformatics.
[42] P. Zhang, F. Wang, J. Hu, R. Sorrentino, Label propagation prediction of drug-drug interactions based on clinical side effects, Scientific reports 5 (1) (2015) 1–10.
[43] F. Zhang, X. Hao, J. Chao, S. Yuan, Label propagation-based approach for detecting review spammer groups on e-commerce websites, Knowledge-Based Systems 193 (2020) 105520.
[44] H. Li, H. Lu, Z. Lin, X. Shen, B. Price, Inner and inter label propagation: salient object detection in the wild, IEEE Transactions on Image Processing 24 (10) (2015) 3176–3186.
[45] M. Timilsina, A. Figueroa, M. d'Aquin, H. Yang, Semi-supervised regression using diffusion on graphs, Applied Soft Computing 104 (2021) 107188.

[46] Z. Song, X. Yang, Z. Xu, I. King, Graph-based semi-supervised learning: A comprehensive review, IEEE Transactions on Neural Networks and Learning Systems.
[47] A. Erdem, M. Pelillo, Graph transduction as a noncooperative game, Neural Computation 24 (3) (2012) 700–723.
[48] S. Vascon, M. Frasca, R. Tripodi, G. Valentini, M. Pelillo, Protein function prediction as a graph-transduction game, Pattern Recognition Letters 134 (2020) 96–105.
[49] C. Han, D. Zhou, Y. Xie, Y. Lei, J. Shi, Label propagation with multi-stage inference for visual domain adaptation, Knowledge-Based Systems 216 (2021) 106809.
[50] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (7553) (2015) 436–444.
[51] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.
URL https://openreview.net/forum?id=SJU4ayYgl
[52] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017.
URL https://proceedings.neurips.cc/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf
[53] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph Attention Networks, International Conference on Learning Representations.
URL https://openreview.net/forum?id=rJXMpikCZ
[54] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, G. E. Dahl, Neural message passing for quantum chemistry, in: International Conference on Machine Learning, PMLR, 2017, pp. 1263–1272.
[55] C. Yang, J. Liu, C. Shi, Extract the knowledge of graph neural networks and go beyond it: An effective knowledge distillation framework, in: Proceedings of the Web Conference 2021, 2021, pp. 1227–1237.
[56] Q. Li, X.-M. Wu, H. Liu, X. Zhang, Z. Guan, Label efficient semi-supervised learning via graph filtering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9582–9591.
[57] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Label propagation for deep semi-supervised learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5070–5079.
[58] M. Douze, A. Szlam, B. Hariharan, H. Jégou, Low-shot learning with large-scale diffusion, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3349–3358.
[59] D. Wang, P. Cui, W. Zhu, Structural deep network embedding, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2016, pp. 1225–1234.
[60] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2016, pp. 855–864.
[61] Q. Guo, E. Cozzo, Z. Zheng, Y. Moreno, Levy random walks on multiplex networks, Scientific reports 6 (2016) 37641.
[62] P. Holme, Network reachability of real-world contact sequences, Physical Review E 71 (4) (2005) 046119.
[63] H. Yang, I. King, M. R. Lyu, Diffusionrank: a possible penicillin for web spamming, in: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2007, pp. 431–438.
[64] X. Zhu, Z. Ghahramani, Learning from labeled and unlabeled data with label propagation, Tech. rep., Citeseer (2002).
[65] I. Miller, M. Miller, J. E. Freund, John E. Freund's mathematical statistics, Prentice Hall, 1999.
[66] H. Ma, H. Yang, M. R. Lyu, I. King, Mining social networks using heat diffusion processes for marketing candidates selection, in: Proceedings of the 17th ACM conference on Information and knowledge management, ACM, 2008, pp. 233–242.
[67] M. Esmaeili, A. Nosratinia, Semi-supervised node classification by graph convolutional networks and extracted side information.
[68] M. De Domenico, V. Nicosia, A. Arenas, V. Latora, Structural reducibility of multilayer networks, Nature communications 6 (1) (2015) 1–9.
[69] S. Wen, Detecting depression from tweets with neural language processing, in: Journal of Physics: Conference Series, Vol. 1792, IOP Publishing, 2021, p. 012058.
[70] C. M. Bishop, Pattern recognition, Machine learning 128 (9).
[71] J. S. Evans, M. A. Murphy, Package 'rfutilities', R package 1 (2015) 1.
[72] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: European Semantic Web Conference, Springer, 2018, pp. 593–607.
[73] I. Balažević, C. Allen, T. M. Hospedales, Tucker: Tensor factorization for knowledge graph completion, arXiv preprint arXiv:1901.09590.
[74] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, International Conference on Machine Learning (ICML), 2016.
[75] M. Nickel, V. Tresp, Tensor factorization for multi-relational learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2013, pp. 617–621.