



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Masters Thesis of Il-Jae Kwon

Solving the Cold Start Problem in Graph Learning with Multitask Strategy

그래프 학습시 콜드 스타트 문제 해결을 위한 멀티테스크
학습 전략

Il-Jae Kwon

August, 2021

Interdisciplinary Program in Neuroscience Major
Graduate School of Natural Science
SEOUL NATIONAL UNIVERSITY

Solving the Cold Start Problem in Graph Learning
with Multitask Strategy

그래프 학습시 콜드 스타트 문제 해결을 위한
멀티테스크 학습 전략

지도교수 : 장병탁

이 논문을 이학 석사 학위논문으로 제출함

July, 2021

서울대학교 자연과학대학

뇌과학 협동과정

권일재

권일재의 이학 석사 학위논문을 인준함

July, 2021

위원장	이상훈
부위원장	장병탁
위원	권준수

Abstract

Data in real world can commonly be represented as graph data. From molecules, social networks, Internet, and even natural language can be represented as graph. Therefore, generating fine representations from the graph data is critical in machine learning. One of the common attributes of most real-world graphs is that graphs are dynamic and can eventually face the cold start problem. A fundamental question is how the new cold nodes acquire initial information in order to be adapted into the existing graph. Here we postulates the cold start problem as a fundamental issue in graph learning and propose a new learning setting, “Expanded Semi-supervised Learning.” In expanded semi-supervised learning we extend the original semi-supervised learning setting even to new cold nodes that are disconnected from the graph. To this end, we propose Cold-Expand model that classifies the cold nodes based on link prediction with multiple goals to tackle. We experimentally prove that by adding additional goal to existing link prediction method, our method outperforms the baseline in both expanded semi-supervised link prediction (at most 24%) and node classification tasks (at most 15%). To the best of our knowledge this is the first study to address expansion of semi-supervised learning to unseen nodes.

Keywords: Graph Neural Networks, Cold start, Multi-task Learning, Semi-supervised Learning

Student Number: 2019-24053

Contents

Abstract	i
Chapter 1 Introduction	1
Chapter 2 Related Works	5
2.1 Semi-supervised Node Classification	5
2.2 Link Prediction	6
2.3 Cold Start Problem	8
2.4 Multi-task Learning Strategy	9
Chapter 3 Method	11
3.1 Problem Definition	11
3.2 ColdExpand: Semi-Supervised Graph Learning in Cold Start . .	12
3.2.1 Link Prediction of Cold Nodes	12
3.2.2 Node Classification of Cold Nodes	13
Chapter 4 Experiment	17
4.1 Dataset	17
4.2 Baseline	18
4.2.1 Expanded Semi-supervised Link Prediction	18

4.2.2	Expanded Semi-supervised Node Classification	18
4.3	Results	19
4.3.1	Results of Link Prediction on Cold Nodes	19
4.3.2	Results of Semi-supervised Node Classification on Cold Nodes	20
4.3.3	Qualitative Results for Node Classification in Cold Start Environment	23
4.3.4	ColdExpand also Improves Conventional Semi-supervised Node Classification	23
4.3.5	Ablation Study	25
Chapter 5	Discussion	28
초록		36

List of Figures

Figure 1.1	Left: Original Semi-supervised Learning. Right: Expanded Semi-supervised Learning. Nodes with solid lines are labeled nodes and other nodes with dot lines are unlabeled nodes. Expanded semi-supervised learning extends the original semi-supervised learning even to new nodes that are disconnected from the graph. By adding unseen cold nodes to the training we can not only expand learning to cold nodes but also use node features of cold nodes to better approximate the marginal distribution of the original graph data.	3
Figure 2.1	Hard parameter sharing for multi-task learning. Same parameters are being trained in order to solve three tasks. Reprinted from [Zhang & Yang, 2017].	10
Figure 3.1	Computational graph for the ColdExpand model	14

Figure 4.1	2D visualization of node embedding vectors from each setting using t-SNE algorithm. Different colors indicate different class labels. t-SNE mappings in the top row are from the Cora dataset while mappings in the bottom row are from Citseer.	24
Figure 4.2	Link prediction results for different λ . GAE-CE represents results using ColdExpand strategy. We added results of GAE which is invariant to λ for comparison. . .	26

List of Tables

Table 4.1	Attributes of each Dataset used in the experiment	17
Table 4.2	Results of link prediction on cold nodes in area under the ROC curve (AUC) scores. Bold score indicated the best AUC score. In every methods, adding our learning strategy generally increased performance.	20
Table 4.3	Experimental results on node classification task in comparison to various other models. Bold score indicated the best accuracy score. Our model outperformed existing GNN methods in every dataset.	21
Table 4.4	Average number of generated edges for each cold node by two different training strategies. Edges per node represents ground truth number of edges per each node. By using additional link prediction loss, ColdExpand-Unfreeze decreased at most 26.5% of new edges compared to ColdExpand-CLS which is closer to the ground truth amount.	22

Table 4.5 Node classification results of non-cold nodes in Cora dataset.
ColdExpand model remarkably enhances traditional GCN
model. % of cold nodes indicates a fraction of randomly
disconnected nodes. 25

Chapter 1

Introduction

Graph data can be easily found around our daily life. Social Network Systems, such as Facebook, Instagram and Twitter, are graph-structured data in which people in the network are considered to be nodes and relationships between people are considered to be edges: two people are friends or sharing posts, etc. In chemistry, molecules can be represented in graphs composed of atoms as nodes and bonds as edges. One step further, interaction between different kinds of molecules can also be represented as graph.

The major distinction from graph data to other domains of data is that graph data has topological structure. With structural information that graph data has, we can infer some unknown attributes of a person (node) based on the information of people he is connected to (i.e., *semi-supervised node classification*). In the case of retail applications, customers and products can be viewed as heterogeneous nodes and edges between customers and products can represent relationships between the customers and the purchased products. Such a graph can be used to represent spending habits of each customer and we

can recommend a product to a user by inferring the likelihood of connection between the user and the product (i.e., *semi-supervised link prediction*).

Recent progress on Graph Neural Networks (GNN) [Bruna *et al.*, 2013, Kipf & Welling, 2016a, Gilmer *et al.*, 2017, Veličković *et al.*, 2018, Jia *et al.*, 2019] allows us to effectively utilize the expressive power of the graph-structured data and to solve various graph related tasks. Early GNN methods tackled semi-supervised node classification task, a task to label all nodes within the graph when only a small subset of nodes is labeled, achieving a satisfactory performance [Zhou *et al.*, 2004]. Link prediction is another graph-related task that was covered comparatively less than other tasks in the field of GNNs. In the link prediction task, the goal is to estimate the likelihood of connection between two nodes given node feature data and topological structure data. Link prediction can be used in recommendation tasks [Chen *et al.*, 2005, Sarwar *et al.*, 2001, Lika *et al.*, 2014, Li & Chen, 2013, Berg *et al.*, 2017] or graph completion tasks [Kazemi & Poole, 2018, Zhang & Chen, 2018]. Most of the work on semi-supervised graph learning and link prediction assumes a static graph, that is, the structural information is at least “partially” observable in terms of nodes.

In the real world, however, new users or items can be added (as nodes) without any topological information [Gope & Jain, 2017]. This is also referred as the *cold start problem* when a new node is presented to an existing graph without a single edge. In contrast to the warm start case, in which at least some topological information is provided, the cold start problem is an extreme case where there isn’t any topological information to refer. In this setting, previous semi-supervised learning algorithms can not propagate information to the cold nodes. Even though the cold start problem is an extreme setting, it is an inevitable problem that occurs often in the real world data.

In this thesis, we postulates the cold start problem as a fundamental issue in

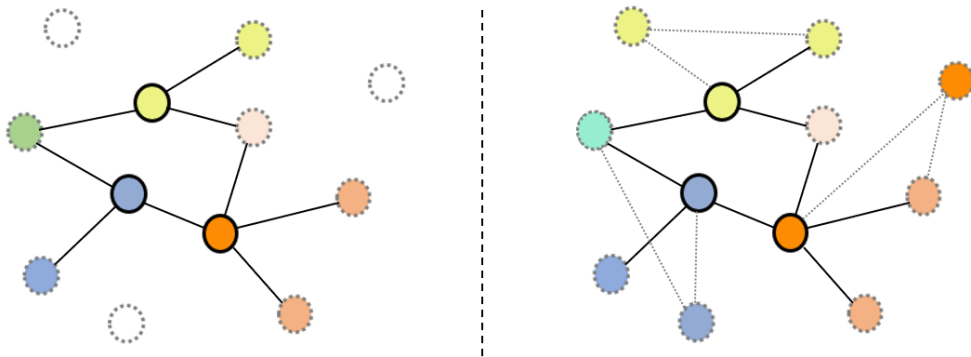


Figure 1.1 Left: Original Semi-supervised Learning. Right: Expanded Semi-supervised Learning. Nodes with solid lines are labeled nodes and other nodes with dot lines are unlabeled nodes. Expanded semi-supervised learning extends the original semi-supervised learning even to new nodes that are disconnected from the graph. By adding unseen cold nodes to the training we can not only expand learning to cold nodes but also use node features of cold nodes to better approximate the marginal distribution of the original graph data.

graph learning and propose a new learning setting, expanded semi-supervised learning. In expanded semi-supervised learning we extend the original semi-supervised learning setting even to new cold nodes that are disconnected from the graph as shown in Figure 1.1. To this end, we suggest the ColdExpand method, the method that uses multi-task learning strategy to alleviate the cold start problem that typical graph learning methods face. Multi-task learning strategy aims to solve two different task with shared parameter to make model to be generalized from task-specific noises. Our model brings link prediction task and node classification task, which are two popular tasks in graph learning, to the multi-task setting and try to solve both tasks with shared model. We experimentally prove that by adding an additional goal to the existing link prediction method, performance on the link prediction task is enhanced in every benchmark dataset and at most by 24%. We also prove that our method can expand semi-supervised node classification even to unseen, cold nodes. To the best of our knowledge, this is the first study to expand semi-supervised learning methods to unseen nodes. In the next Chapter, we briefly introduce related works. In Chapter 3, we define our problem definition. Finally, in Chapter 4 we propose our ColdExpand model followed by our experimental environments along with corresponding results.

Chapter 2

Related Works

2.1 Semi-supervised Node Classification

Semi-supervised learning is a combination of unsupervised learning and supervised learning where true labels are only partially given. Semi-supervised learning algorithms aim to improve the model's performance by using unlabeled data points on top of labeled ones to approximate the underlying marginal data distribution [Van Engelen & Hoos, 2020]. One of the most popular fields of semi-supervised learning is graph based semi-supervised node classification where all nodes should be classified when only a few node labels are given. In order to smooth the given subset of labels throughout the graph, various methods have been studied to effectively represent nodes. Deep Walk [Perozzi *et al.*, 2014], LINE [Tang *et al.*, 2015], node2vec [Grover & Leskovec, 2016] were early deep-learning-based methods targeting the node classification task by learning latent representations from truncated random walks. However, these models fail to share parameters between nodes causing learning to be inefficient. [Kipf &

Welling, 2016a] introduced Graph Convolutional Networks (GCN) which use an efficient layer-wise propagation rule by approximating the first-order of spectral graph convolution. By limiting the spectral convolution to the first-order, GCNs not only lightened computation cost of the operation but also alleviated the over-fitting problem that previous spectral methods had. Commonly, GCN model is formed from stacked convolutional layers; the number of which decides how many hops of neighbors to consider for the convolution operation. [Gilmer *et al.*, 2017] presented Message Passing Neural Networks (MPNN) as a general form of spatial convolution operation and treated GCN as specific kind of a message passing process. In MPNNs, information between nodes is delivered directly by edges without visiting any spectral domains.

2.2 Link Prediction

Links in graph-structured data appeal important interactions between nodes that may not be represented solely by node attributes. To this end, link prediction is an important task in order to effectively utilize graph data. In the link prediction task, we aim to predict whether an edge exists between two nodes from the same graph. Predicting the existence of links between nodes is useful for many applications including recommendation systems and graph completion tasks.

On a high level, link prediction methods can be separated into similarity-based and learning-based methods. Similarity-based link prediction is the simplest way to predict the link by assigning a similarity score S_{xy} between two nodes x, y . Common Neighbors [Marchette & Priebe, 2008], Salton Index, Jaccard Index [Leydesdorff, 2008], Sorensen Index, Hub Promoted Index, Hub Depressed Index [Zhou *et al.*, 2009], Leicht-Holme-Newman Index (LHN1) [Le-

icht *et al.*, 2006] each define similarity scores by using the overlapping set of neighboring nodes between two nodes. While the former neighborhood-based methods focus on somewhat local similarity, other similarity-based methods consider global similarity by taking into account higher order links. Katz Index [Katz, 1953] considers the ensemble of all paths between two nodes summed with quadratic weight according to the distance of each path. SimRank [Jeh & Widom, 2002] also uses random walk process but measures how soon two random walkers from the two different nodes meet. However, when a cold node is introduced, the required structural information needed for previous similarity-based approaches is missing, thus making them impossible to be applied in cold start situations [Wang *et al.*, 2016].

On the other hand, learning-based methods formulate link prediction as a binary classification problem given feature embedding vectors of two nodes; if an edge exists between the two nodes a positive label is given and vice versa. [Kipf & Welling, 2016b] introduced the graph auto-encoder (GAE), an unsupervised learning technique, which aims to reconstruct the original adjacency matrix. As a variant of GAE, variational graph auto-encoders (VGAE) additionally train a separate GCN module resulting in the logarithm of the variant matrix, $\log(\sigma^2)$. Additional Kullback-Leibler divergence loss between the distribution of encoder and decoder is added to the original reconstruction loss. [Pan *et al.*, 2019] further improved upon GAEs by applying adversarial training to regularize the latent node embedding matrix Z to match a prior distribution of the original node feature matrix X . A simple MLP-based discriminator D was used for adversarial learning between the real distribution of $G = \{X, A\}$ and the latent node matrix Z . Unlike similarity-based methods, learning-based methods can be applied to cold start problems since they make use of node feature information along with topological structure information.

2.3 Cold Start Problem

The cold start problem is a common problem in real-world graph-structured data where new nodes may be added to an existing graph. Most work tackling the cold start problem focuses on solving the recommendation problem in a cold start setting. Recommendation systems are widely used in the real world applications such as friend recommendation, movie recommendation and purchase recommendation. In the recommendation problem, the task is to predict recommendation links from a heterogeneous graph containing two types of graphs; a user-user social graph and user-item rating graph. Users are involved in both types of graphs to connect the two types of graphs into a single heterogeneous graph. The collaborative filtering (CF) method is one of the traditional methods where a rating is predicted based on similarities between users or items. In short, the CF method makes recommendations based on ratings given by other users in the social graph. [Sarwar *et al.*, 2001] introduced an item-based CF method, which computes item-item similarities, and proved it outperforms other user-based collaborative filtering recommendation methods. Efforts to make use of CF for recommendation systems in the cold start scenario have been made. [Lika *et al.*, 2014] approached the cold user problem by exploiting CF followed by hard-clustering of users based on their attributes. Once the cluster of the new user has been decided, CF is only applied to members of the same cluster. [Berg *et al.*, 2017] formulated the matrix completion task of recommendation systems as a link prediction problem by viewing the user-item graph as a bipartite interaction graph. However, they instead used a warm start setting where at least one rating for a user is given which is not precisely a cold start assumption.

2.4 Multi-task Learning Strategy

Multi-task learning is a learning strategy to jointly train models with related multiple tasks in order to improve the generalization performance of tasks [Zhang & Yang, 2017]. Multi-task learning is largely inspired by human learning, where human can easily generalize experiences in one task to other similar tasks. For example, when one has learnt how to play tennis, they can easily generalize the experience when they try to learn ping-pong which is a similar task. As human learning, multi-task learning aims to train generalized model by sharing parameters to solve multiple tasks.

Multi-task learning can be distinguished by how parameters are shared to solve each tasks. Hard sharing multi-task learning, which is depicted in Figure 2.1, use exactly same parameters to solve different kinds of tasks. On the other hand soft sharing multi-task learning models use distinct parameters for each tasks that parameters are shared and updated by specific rules.

The generalizing power of multi-task learning mainly comes from task-specific noises that the data has [Ruder, 2017]. As different tasks hold different noise patterns, the model trained with multiple tasks can learn even general representation, which is similar to an advantage of a data augmentation. In addition, different tasks train the model to attention on different features, allowing the model to explore more features before converging attention to an easy solution. Finally, adding an auxiliary task has effect of a regularizer that gives inductive bias that alleviates overfitting issue of the model. In this thesis, we focus on hard sharing multi-task learning strategy by tackling two different tasks on graph learning, node classification and link prediction, with one model.

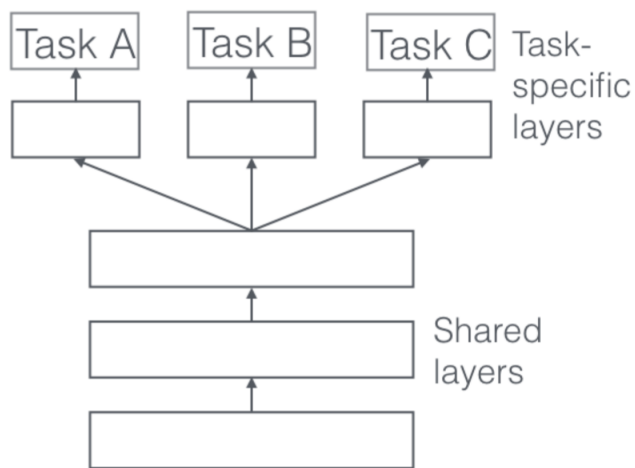


Figure 2.1 Hard parameter sharing for multi-task learning. Same parameters are being trained in order to solve three tasks. Reprinted from [Zhang & Yang, 2017].

Chapter 3

Method

3.1 Problem Definition

A graph G is denoted as a pair $(\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, \dots, v_N\}$ the set of nodes (vertices), and $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ the set of edges. Each node v_i is associated with a feature vector $x_i \in \mathbb{R}^F$. To make notation more compact, the set of node feature vectors of graph G is denoted as a matrix $X = [x_1, x_2, \dots, x_N]^\top \in \mathbb{R}^{N \times F}$. Additionally, a graph has an N -by- N adjacency matrix A where $A_{i,j}$ represents the existence of an edge between v_i and v_j and a degree matrix D , a diagonal matrix which contains information about the degree of each node.

In the cold start setting, we will call those nodes that are newly introduced without any link from the existing graph *cold nodes* and let them be denoted by v_i^c . In the same manner, existing nodes in the graph are denoted as v_j^e . Then, X_e only contains feature vectors of existing nodes $x_0^e, x_1^e, x_2^e, \dots, x_{N_e}^e$ and X_c contains feature vectors of cold nodes $x_0^c, x_1^c, x_2^c, \dots, x_{N_c}^c$, where N_c and N_e are the number of existing nodes and cold nodes, respectively.

In our problem setting, we assume that new nodes are added to the existing graph and need to find possible links between new nodes and existing nodes followed by a node classification to figure out the hidden attributes of new nodes. Our problem definition is as follows: Given existing graph, $G = (V_e, A)$, and cold nodes X_c , find out all possible links between cold nodes X_c and existing nodes X_e , then predict the class of cold nodes X_c .

3.2 ColdExpand: Semi-Supervised Graph Learning in Cold Start

Here we present our method, the ColdExpand, that compensates the lack of topological information from the cold nodes by adding an additional task of semi-supervised node classification to a deep autoencoding principle in the stage of edge reconstruction.

3.2.1 Link Prediction of Cold Nodes

Predicting possible edges of cold nodes is an important sub-goal for expanding a semi-supervised learning but is by itself still a difficult task because the only clue the new node provides is its own node features. As mentioned in Section 2.2, traditional similarity-based link prediction methods are not applicable in the cold start setting since no neighboring nodes of the cold node are known. On the other hand, learning-based link prediction algorithms [Passino *et al.*, 2019, Kipf & Welling, 2016b, Pan *et al.*, 2019] can be applied in the cold start setting by computing similarity scores between two embedded node feature vectors followed by a binary classification layer. Still, while embeddings of the existing nodes are conditioned on both the node feature and the features of neighboring nodes, embeddings of the cold nodes refer only to their own node features

without any help from the topological structure of the graph resulting in isolated learning.

$$x_i^e = \text{GNN}_{\text{encoder}}(x_i^e, N(x_i^e)) \quad (3.1)$$

$$x_i^c = \text{GNN}_{\text{encoder}}(x_i^c, \{\emptyset\}) \quad (3.2)$$

when $N(x_i)$ is a set of neighboring nodes of node x_i and weights of the $\text{GNN}_{\text{encoder}}$ are shared between the two. This hinders traditional GNN encoders from generalizing their training to the cold nodes. In order to make a common goal that can be accomplished by both existing nodes x^e and cold nodes x^c , we train the GCN Encoder with an additional node classification goal as given by the following equation.

$$L_{\text{total}} = L_{\text{link}} + L_{\text{cls}} \quad (3.3)$$

3.2.2 Node Classification of Cold Nodes

Figure 3.1 illustrates the overall structure of our ColdExpand model. Our model is composed of two parts; first, the GCN module for link prediction on the cold nodes and, secondly, the GCN module for classification on the cold nodes. We denote the first GCN module as GCN Encoder and the second GCN module as GCN Classifier. In the GCN Encoder, graph G and a cold node v^c are given as input. We use two basic GCN layers as suggested in [Kipf & Welling, 2016a] for an encoder to get the embedded node vector Z . In order to predict any possible links of the cold node v^c , we feed this embedded node vector Z to an inner product decoder which is similar to previous link prediction methods [Passino *et al.*, 2019, Kipf & Welling, 2016b, Pan *et al.*, 2019]. Along with original edges, predicted links are then fed to the GCN Classifier for node classification.

One GCN layer can be written as follows:

$$M(X, A, \Theta) = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} X \Theta \quad (3.4)$$

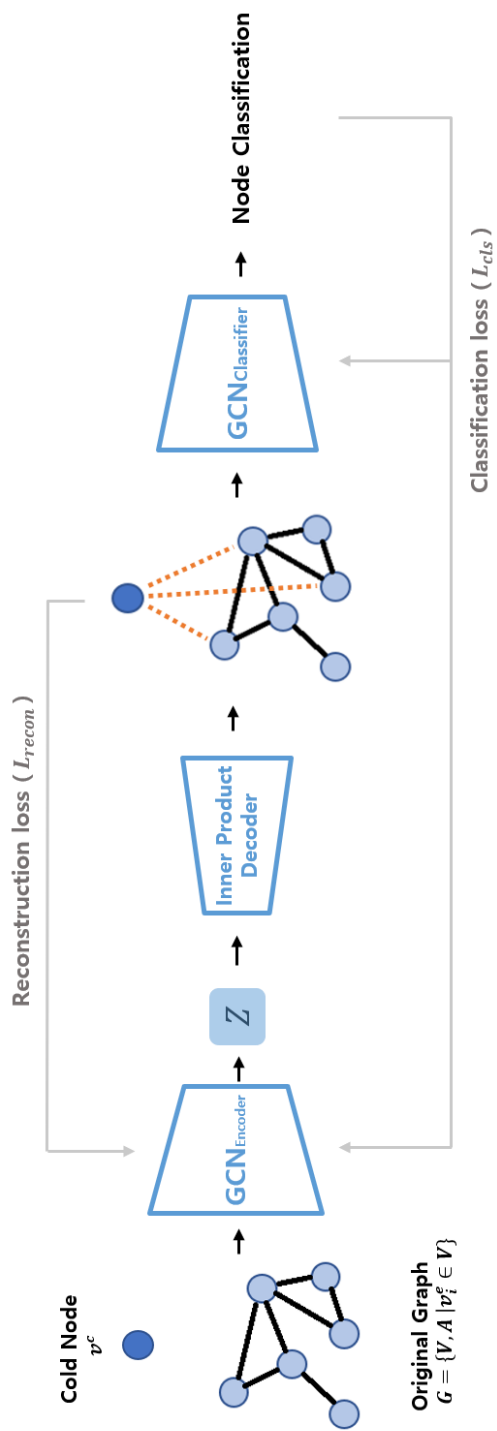


Figure 3.1 Computational graph for the ColdExpand model

where \tilde{A} is an adjacency matrix of graph G with additional self connections and \tilde{D} is the degree matrix of \tilde{A} . Θ is a weight matrix for GCN module. Using the basic 2-layer GCN module, the embedding node matrix for all existing nodes X_e and cold nodes X_c is represented as follows:

$$q(Z_e|(X_e, A_{ee})) = M(\text{ReLU}(M(X_e, A_{ee}, \Theta_1), \Theta_2)) \quad (3.5)$$

$$q(Z_c|(X_c, A_{ee})) = \text{ReLU}(X_c \Theta_1) \Theta_2 \quad (3.6)$$

where Θ_1 is a weight matrix for the first-layer GCN and Θ_2 for the second.

$$p(A_{ij}^{ec} = 1 | z_i^e, z_j^c) = \sigma(z_i^e z_j^{c\top}) \quad (3.7)$$

The inner product decoder computes the sigmoid of the inner product between two embedded node vectors. A_{ec} indicates possible edges between the existing nodes and the cold nodes. The probability of existence of a link between existing nodes x_i^e and cold nodes x_j^c is defined as follows:

Once the probabilities of links have been computed, new hard links for the cold nodes can be added.

$$A_{ec} = \begin{cases} 1, & \text{if } p(A_{ij}^{ec} = 1 | z_i^e, z_j^c) > \xi \\ 0, & \text{otherwise} \end{cases} \quad (3.8)$$

where ξ is a hyperparameter which is a threshold for hard links. During the experiment ξ was set to 0.5.

$$\hat{A} = \begin{pmatrix} A_{ee} & A_{ec} \\ A_{ce} & A_{cc} \end{pmatrix} \quad (3.9)$$

A_{cc} can be obtained in a same manner. Then the new \hat{A} is fed to the GCN Classifier for node classification on the cold nodes. In order to maintain a continuous gradient along the model, we use the convolution operation with edge weights.

Weights of all additional edges A_{ec}, A_{cc} are determined by the probability score of the edge, $p(A_{ij}^{ec} = 1)$ while all the other weights for existing edges are fixed to 1.

$$\hat{Y} = \text{softmax}(\text{GCN}_{\text{Classifier}}(X, \hat{A})) \quad (3.10)$$

we only use $\hat{Y}_c \in \hat{Y}$ for class prediction on cold nodes. For optimization, we use two losses for multi-task learning. The first loss is a link reconstruction loss from the link prediction part of the model and the second loss is node classification loss from GCN Classifier.

$$L_{\text{recon}} = \mathbb{E}_{q(Z|(X,A))} [\log p(A_{ec}|Z) + \log p(A_{cc}|Z)] \quad (3.11)$$

$$L_{\text{cls}} = - \sum_{\hat{y}^c \in \hat{Y}_c} y^c \log \hat{y}^c \quad (3.12)$$

$$L_{\text{total}} = L_{\text{recon}} + \lambda L_{\text{cls}} \quad (3.13)$$

Finally, the total loss is represented as a weighted sum of the two losses. λ is a hyperparameter for training.

Chapter 4

Experiment

4.1 Dataset

In order to evaluate the performance of graph learning in the cold start setting, we first select 10% of randomly shuffled nodes as a test set of cold nodes. Then we create positive edge samples of the set of cold nodes by parsing all links in and out of the cold nodes and every other non-links become negative samples. Whole positive and negative edge samples are used as test sets for the cold start link prediction task and are hidden in the training stage along with the class

Name	Nodes	Edges	Classes	Cold Nodes
Cora	3,327	4,732	6	332
CiteSeer	2,708	5,429	7	270
PubMed	19,717	44,338	3	1971

Table 4.1 Attributes of each Dataset used in the experiment

labels of cold nodes. This process is done in 10-fold cross-validation on all nodes in the dataset. In similar fashion, 5% of the nodes were masked as validation set and another 10% of the nodes as training set. During training we choose the same amount of negative edge samples as positive edge samples. At last, in the training period edges of 20% of nodes were masked. For semi-supervised node classification only 10% of the labels of the nodes were given.

4.2 Baseline

Since there are not many prior works for cold start graph learning tasks, we created some baselines to prove the benefits of our model.

4.2.1 Expanded Semi-supervised Link Prediction

We used learning-based link prediction algorithms GAE, VGAE [Kipf & Welling, 2016b] and ARGAE [Pan *et al.*, 2019] to compare the performance of link generation for cold nodes. For fair comparison we have restricted the number of edges used in training to 10%. Then each link prediction method was trained with the ColdExpand method by adding an additional task of node classification with loss function as given in Equation 3.13.

4.2.2 Expanded Semi-supervised Node Classification

The simplest baseline for semi-supervised node classification is a linear model that only considers node attributes of the graph. The number of labels that the model can see was restricted for fair comparison.

$$E = \text{MLP}(X_e), \quad \hat{y} = \text{softmax}(E) \quad (4.1)$$

The other baselines are vanilla GNNs which are trained on the existing graph and used to predict the labels of the cold nodes. In this setting, the same number

of labels for existing nodes are provided during training.

$$E = \text{GNN}(X_e, A), \quad \hat{y} = \text{softmax}(E) \quad (4.2)$$

There are several training strategies for our ColdExpand model. In the ColdExpand-Freeze model, first part of the model, the GCN Encoder, is pre-trained with a multi-task goal under fixed λ . Once pre-training is completed, weights of the GCN Encoder are frozen and training on the GCN Classifier takes place. Unlike the ColdExpand-Freeze model, the ColdExpand-Unfreeze model updates the weights of the GCN Encoder along with the GCN Classifier. For the ColdExpand models, we choose the best setting among several choices of λ . To demonstrate the advantage of adding an additional task, we also tested with ColdExpand-CLS which shares the same model architecture but only uses L_{cls} for optimization. Each test was done with various convolutional GNN methods including GCN, GraphSAGE [Hamilton *et al.*, 2017], and GAT [Veličković *et al.*, 2017].

4.3 Results

4.3.1 Results of Link Prediction on Cold Nodes

Table 4.2 illustrates the results of expanded semi-supervised link prediction on cold nodes in *area under the ROC curve* (AUC) scores for each benchmark and dataset. Training with our ColdExpand strategy, ColdExpand encoders outperform most learning-based algorithms in reconstructing the links of the cold nodes. Especially in the Cora and CiteSeer datasets which have enough number of classes, our model improves the performance (at most 24%) by a large margin. In PubMed, where there are only 3 classes of labels, our model achieves 5.7% higher AUC score. With the additional training goal, our model definitely outperforms traditional link prediction methods on predicting the

Algorithm	Cora	CiteSeer	PubMed
GAE	0.625 ± 0.017	0.678 ± 0.029	0.809 ± 0.007
GAE (ColdExpand)	0.813 ± 0.021	0.854 ± 0.019	0.907 ± 0.006
VGAE	0.655 ± 0.020	0.651 ± 0.016	0.779 ± 0.030
VGAE (ColdExpand)	0.643 ± 0.034	0.676 ± 0.058	0.836 ± 0.031
ARGA	0.639 ± 0.018	0.679 ± 0.021	0.858 ± 0.016
ARGA (ColdExpand)	0.784 ± 0.024	0.824 ± 0.024	0.881 ± 0.006

Table 4.2 Results of link prediction on cold nodes in area under the ROC curve (AUC) scores. Bold score indicated the best AUC score. In every methods, adding our learning strategy generally increased performance.

possible links of the cold nodes and training is more beneficial when there are enough number of classes.

4.3.2 Results of Semi-supervised Node Classification on Cold Nodes

Table 4.3 shows experimental results on the expanded semi-supervised node classification task in comparison to various other models. In order to test how inductive learning methods perform in the cold start setting, we tested with vanilla GCN, GraphSAGE, GAT methods by using the weights learned from non-cold start settings to classify cold nodes. On every dataset, our ColdExpand method performed better than traditional inductive GNN methods in the cold start setting. This is due to the fundamental limitation that inductive learning methods have that they exclude cold nodes during training but rather test each node independently. In contrast, our ColdExpand model uses a multi-task strategy to incorporate cold nodes into the existing graph and furthermore include

Algorithm	Cora	CiteSeer	PubMed
Linear Function	0.644 ± 0.025	0.650 ± 0.021	0.838 ± 0.011
GCN			
Vanilla GCN	0.690 ± 0.023	0.656 ± 0.024	0.836 ± 0.006
ColdExpand-CLS	0.445 ± 0.153	0.609 ± 0.082	0.768 ± 0.028
ColdExpand-Freeze	0.760 ± 0.015	0.688 ± 0.026	0.852 ± 0.007
ColdExpand-Unfreeze	0.792 ± 0.029	0.710 ± 0.020	0.835 ± 0.010
GraphSAGE			
Vanilla GraphSAGE	0.545 ± 0.058	0.596 ± 0.071	0.801 ± 0.011
ColdExpand-CLS	0.444 ± 0.148	0.575 ± 0.082	0.771 ± 0.017
ColdExpand-Freeze	0.746 ± 0.032	0.683 ± 0.023	0.817 ± 0.009
ColdExpand-Unfreeze	0.791 ± 0.029	0.711 ± 0.018	0.816 ± 0.009
GAT			
Vanilla GAT	0.534 ± 0.040	0.630 ± 0.024	0.796 ± 0.016
ColdExpand-CLS	0.490 ± 0.126	0.571 ± 0.122	0.764 ± 0.022
ColdExpand-Freeze	0.638 ± 0.049	0.634 ± 0.030	0.821 ± 0.008
ColdExpand-Unfreeze	0.686 ± 0.028	0.673 ± 0.021	0.817 ± 0.011

Table 4.3 Experimental results on node classification task in comparison to various other models. Bold score indicated the best accuracy score. Our model outperformed existing GNN methods in every dataset.

Algorithm	Cora	CiteSeer	PubMed
Edges per node	1.42	2.00	2.25
ColdExpand-CLS	5.96	6.30	8.23
ColdExpand-Unfreeze	5.12	4.63	6.73
Decreased edges (%)	14.1%	26.5%	18.3%

Table 4.4 Average number of generated edges for each cold node by two different training strategies. Edges per node represents ground truth number of edges per each node. By using additional link prediction loss, ColdExpand-Unfreeze decreased at most 26.5% of new edges compared to ColdExpand-CLS which is closer to the ground truth amount.

them in training. The linear function model which does not consider topological structure of a graph, generally performed poorly. However for PubMed, which has only three classes of labels, the linear function model showed performance close to our model. On other dataset, our ColdExpand model outperformed linear and vanilla GNN models by a big gap in performance: Cora (15%), CiteSeer (8%).

Another notable result from the experiment was number of additional edges each GCN Encoder generates. In case of ColdExpand-CLS where classification loss was solely used for optimization, the number of the additional edges tends to explode, generating much more edges between cold nodes and existing nodes. On the other hand, other ColdExpand models generated relatively fewer edges (Table 4.4) using the link prediction loss as a regularizer for the GCN Encoder.

Results show that in general, ColdExpand-Unfreeze model works better than ColdExpand-Freeze model. While traditional link prediction methods focus on reconstructing 1-hop links, [Klicpera *et al.*, 2019] found that diffusion of graph,

considering neighbors from multiple hops, improves semi-supervised learning without over-smoothing problem. This indicates that one-hop link prediction may not be the best solution for classifying the nodes. By training GCN Encoder during the stage of node classification we can find such links that can improve node classification that traditional link prediction methods could not find. In contrast, PubMed is a large graph with many nodes which degrades the effect of noisy connections.

4.3.3 Qualitative Results for Node Classification in Cold Start Environment

In order to demonstrate the benefit of our model we further visualized node embeddings using t-SNE [Maaten & Hinton, 2008] for each of the following settings: GCN in an ordinary setting, GCN in the cold start setting, and ColdExpand in the cold start setting. t-SNE is a technique for dimension reduction while preserving the high-dimensional distance between data points. Figure 4.1 shows the results of our experiments. When a traditional inductive GCN model was applied in an ordinary setting, the GCN model successfully mapped nodes from the same label to a close distance. However, once the GCN model is trained in the cold start setting, the data points did not form good clusters but rather got scattered across a wider range. On the other hand, our ColdExpand model formed clear clusters similar to those seen in the GCN mapping in the ordinary setting.

4.3.4 ColdExpand also Improves Conventional Semi-supervised Node Classification

While above results show that our ColdExpand model improves link prediction and node classification of cold nodes by a large margin, in this subsection we

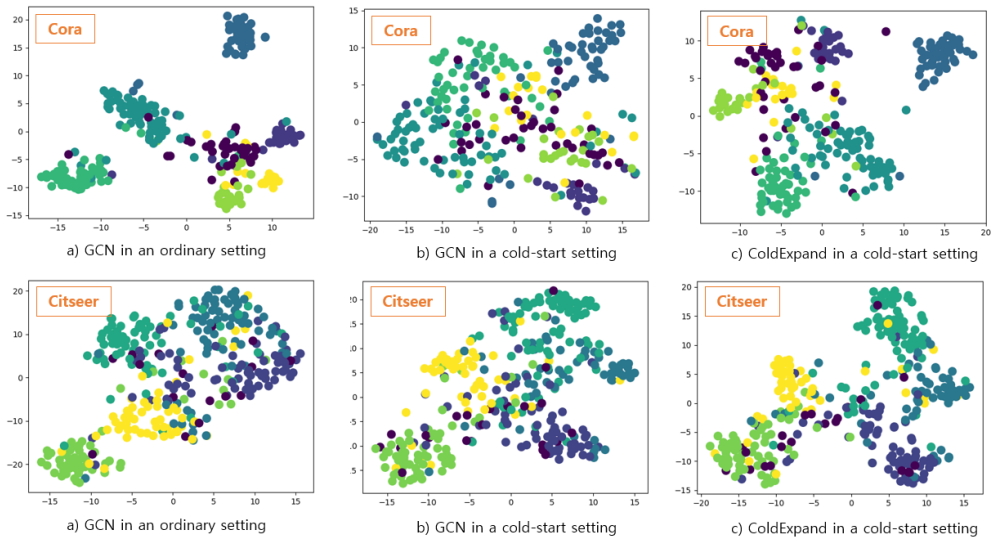


Figure 4.1 2D visualization of node embedding vectors from each setting using t-SNE algorithm. Different colors indicate different class labels. t-SNE mappings in the top row are from the Cora dataset while mappings in the bottom row are from Citseer.

% of cold nodes	GCN	ColdExpand
50%	0.730 ± 0.024	0.738 ± 0.015
30%	0.756 ± 0.014	0.780 ± 0.006
10%	0.767 ± 0.025	0.806 ± 0.012

Table 4.5 Node classification results of non-cold nodes in Cora dataset. ColdExpand model remarkably enhances traditional GCN model. % of cold nodes indicates a fraction of randomly disconnected nodes.

experimentally prove that our model can even improve semi-supervised learning on non-cold nodes in a setting where cold nodes are added to the original graph, which is very likely to happen in the real world. For this experiment we randomly disconnected a fraction of nodes from the graph and assumed these were the newly introduced nodes to the original graph. In the experiment, k-fold validation was done to ensure that every nodes are considered.

Table 4.5 shows node classification results of non-cold nodes where cold nodes are disconnected from the graph and act as nodes that are newly introduced to the graph. Using the ColdExpand model, we can not only improve the learning on cold nodes but also successfully utilize data from unseen nodes to improve semi-supervised learning on existing nodes. Our method is, to the best of our knowledge, the only method that takes advantage of unseen cold nodes to improve learning of the existing nodes while previous methods only focus on solving this task in a static graph.

4.3.5 Ablation Study

Figure 4.2 illustrates trend of the results on cold start link prediction with varying hyperparameter, λ , which decides the amount of additional classifica-

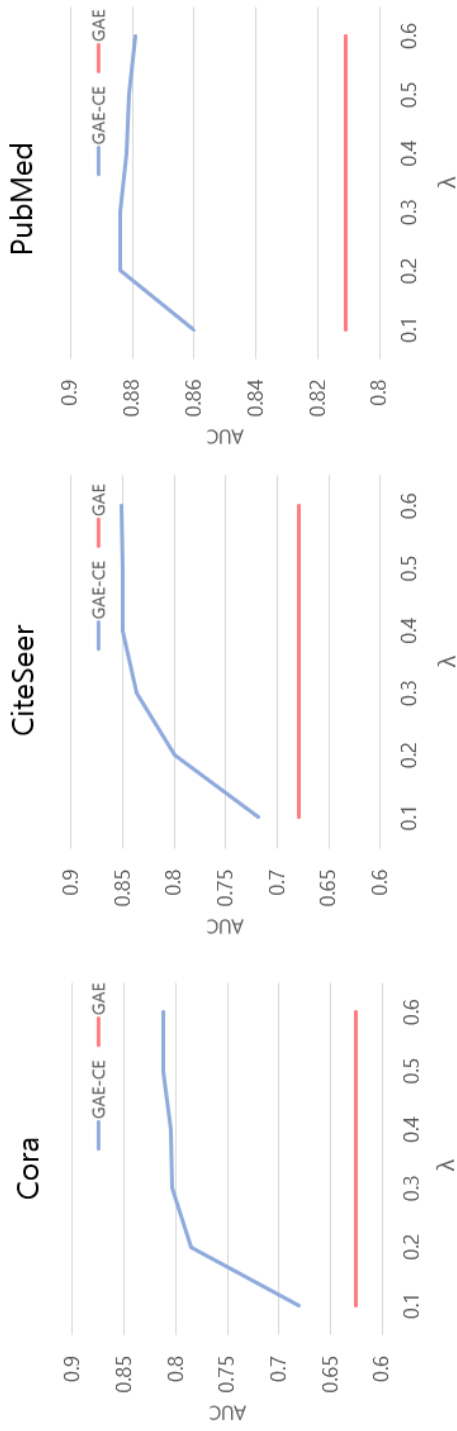


Figure 4.2 Link prediction results for different λ . GAE-CE represents results using ColdExpand strategy. We added results of GAE which is invariant to λ for comparison.

tion loss to be considered. Each dataset has different property resulting in a different optimal λ . From the trend we can easily figure out that adding an additional task benefits cold start link prediction. This approach was more effective when there were enough distinct classes to map nodes to.

Chapter 5

Discussion

In this thesis, we draw attention to a new task Expanded Semi-Supervised Learning which is a task to extend semi-supervised learning even to cold nodes. Even though the cold start environment is an extreme situation, most real-world applications must face it. We proposed the ColdExpand method which uses multi-task strategy to overcome the lack of topological information from cold nodes.

Our new task can be applied in many categories of machine learning. For example, community detection is a task of discovering overlapping groups of nodes, also referred to as communities, that share similar properties from the network graph. While most works on community detection only focused on static social networks [Leskovec & Mcauley, 2012, Chen *et al.*, 2017, Li *et al.*, 2019, Jia *et al.*, 2019], it is critical that learning can be propagated even to new users. Our method can be used to categorize and adapt a new user to an existing community. Our task can also be applied in discovering unknown relations between medicines as new drugs are introduced.

Solving the cold start problem in graphs is an important stepping stone for incremental learning. Incremental learning [Joshi & Kulkarni, 2012] is a learning policy that extends its existing model as new samples are dynamically added. Incremental learning allows long-term perceptual learning that resembles synaptic plasticity of our brain which is based on Hebbian learning rule. One of the key benefits from our model is that ColdExpand is an incremental method.

Graphs that grow incrementally can be used to broaden the knowledge in cognitive models that rely on topological structure such as Adaptive Resonance Theory [Carpenter & Grossberg, 2010] or the Neural Gas model [Fritzke, 1995]. In these models, new concepts are treated as new nodes to be connected to existing knowledge structure in order to build long-term perceptual concepts. However, these models assume each new samples independent to one another, failing to utilize rich relation information between the samples. Considering that most of sensory inputs that human learn from are multi-modal, it is important for cognitive models to be aware of relation between samples: co-occurring, inhibiting, additive and etc. On the other hand, our model is a graph model that can fully utilize relation information between samples while incrementally adding new samples to the existing graph. We leave this for a future work.

Bibliography

- [Berg *et al.*, 2017] Berg, Rianne van den, Kipf, Thomas N, & Welling, Max. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.
- [Bruna *et al.*, 2013] Bruna, Joan, Zaremba, Wojciech, Szlam, Arthur, & LeCun, Yann. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.
- [Carpenter & Grossberg, 2010] Carpenter, Gail A, & Grossberg, Stephen. 2010. *Adaptive resonance theory*.
- [Chen *et al.*, 2005] Chen, Hsinchun, Li, Xin, & Huang, Zan. 2005. Link prediction approach to collaborative filtering. *Pages 141–142 of: Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*. IEEE.
- [Chen *et al.*, 2017] Chen, Zhengdao, Li, Xiang, & Bruna, Joan. 2017. Supervised community detection with line graph neural networks. *arXiv preprint arXiv:1705.08415*.
- [Fritzke, 1995] Fritzke, Bernd. 1995. A growing neural gas network learns topologies. *Pages 625–632 of: Advances in neural information processing systems*.

- [Gilmer *et al.*, 2017] Gilmer, Justin, Schoenholz, Samuel S, Riley, Patrick F, Vinyals, Oriol, & Dahl, George E. 2017. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*.
- [Gope & Jain, 2017] Gope, Jyotirmoy, & Jain, Sanjay Kumar. 2017. A survey on solving cold start problem in recommender systems. *Pages 133–138 of: 2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE.
- [Grover & Leskovec, 2016] Grover, Aditya, & Leskovec, Jure. 2016. node2vec: Scalable feature learning for networks. *Pages 855–864 of: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*.
- [Hamilton *et al.*, 2017] Hamilton, Will, Ying, Zhitao, & Leskovec, Jure. 2017. Inductive representation learning on large graphs. *Pages 1024–1034 of: Advances in neural information processing systems*.
- [Jeh & Widom, 2002] Jeh, Glen, & Widom, Jennifer. 2002. SimRank: a measure of structural-context similarity. *Pages 538–543 of: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*.
- [Jia *et al.*, 2019] Jia, Yuting, Zhang, Qinqin, Zhang, Weinan, & Wang, Xinbing. 2019. CommunityGAN: Community detection with generative adversarial nets. *Pages 784–794 of: The World Wide Web Conference*.
- [Joshi & Kulkarni, 2012] Joshi, Prachi, & Kulkarni, Parag. 2012. Incremental learning: Areas and methods-a survey. *International Journal of Data Mining & Knowledge Management Process*, **2**(5), 43.

- [Katz, 1953] Katz, Leo. 1953. A new status index derived from sociometric analysis. *Psychometrika*, **18**(1), 39–43.
- [Kazemi & Poole, 2018] Kazemi, Seyed Mehran, & Poole, David. 2018. Simple embedding for link prediction in knowledge graphs. *Pages 4284–4295 of: Advances in neural information processing systems*.
- [Kipf & Welling, 2016a] Kipf, Thomas N, & Welling, Max. 2016a. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- [Kipf & Welling, 2016b] Kipf, Thomas N, & Welling, Max. 2016b. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.
- [Klicpera *et al.*, 2019] Klicpera, Johannes, Weissenberger, Stefan, & Günnemann, Stephan. 2019. Diffusion improves graph learning. *Pages 13354–13366 of: Advances in Neural Information Processing Systems*.
- [Leicht *et al.*, 2006] Leicht, Elizabeth A, Holme, Petter, & Newman, Mark EJ. 2006. Vertex similarity in networks. *Physical Review E*, **73**(2), 026120.
- [Leskovec & Mcauley, 2012] Leskovec, Jure, & Mcauley, Julian J. 2012. Learning to discover social circles in ego networks. *Pages 539–547 of: Advances in neural information processing systems*.
- [Leydesdorff, 2008] Leydesdorff, Loet. 2008. On the normalization and visualization of author co-citation data: Salton’s Cosine versus the Jaccard index. *Journal of the American Society for Information Science and Technology*, **59**(1), 77–85.
- [Li *et al.*, 2019] Li, Pei-Zhen, Huang, Ling, Wang, Chang-Dong, & Lai, Jian-Huang. 2019. EdMot: An edge enhancement approach for motif-aware com-

- munity detection. *Pages 479–487 of: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*
- [Li & Chen, 2013] Li, Xin, & Chen, Hsinchun. 2013. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems*, **54**(2), 880–890.
- [Lika *et al.*, 2014] Lika, Blerina, Kolomvatsos, Kostas, & Hadjiefthymiades, Stathes. 2014. Facing the cold start problem in recommender systems. *Expert Systems with Applications*, **41**(4), 2065–2073.
- [Maaten & Hinton, 2008] Maaten, Laurens van der, & Hinton, Geoffrey. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, **9**(Nov), 2579–2605.
- [Marchette & Priebe, 2008] Marchette, David J, & Priebe, Carey E. 2008. Predicting unobserved links in incompletely observed networks. *Computational Statistics & Data Analysis*, **52**(3), 1373–1386.
- [Pan *et al.*, 2019] Pan, Shirui, Hu, Ruiqi, Fung, Sai-fu, Long, Guodong, Jiang, Jing, & Zhang, Chengqi. 2019. Learning graph embedding with adversarial training methods. *IEEE Transactions on Cybernetics*, **50**(6), 2475–2487.
- [Passino *et al.*, 2019] Passino, Francesco Sanna, Bertiger, Anna S, Neil, Joshua C, & Heard, Nicholas A. 2019. Link prediction in dynamic networks using random dot product graphs. *arXiv preprint arXiv:1912.10419*.
- [Perozzi *et al.*, 2014] Perozzi, Bryan, Al-Rfou, Rami, & Skiena, Steven. 2014. Deepwalk: Online learning of social representations. *Pages 701–710 of: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.*

- [Ruder, 2017] Ruder, Sebastian. 2017. *An Overview of Multi-Task Learning in Deep Neural Networks*.
- [Sarwar *et al.*, 2001] Sarwar, Badrul, Karypis, George, Konstan, Joseph, & Riedl, John. 2001. Item-based collaborative filtering recommendation algorithms. *Pages 285–295 of: Proceedings of the 10th international conference on World Wide Web*.
- [Tang *et al.*, 2015] Tang, Jian, Qu, Meng, Wang, Mingzhe, Zhang, Ming, Yan, Jun, & Mei, Qiaozhu. 2015. Line: Large-scale information network embedding. *Pages 1067–1077 of: Proceedings of the 24th international conference on world wide web*.
- [Van Engelen & Hoos, 2020] Van Engelen, Jesper E, & Hoos, Holger H. 2020. A survey on semi-supervised learning. *Machine Learning*, **109**(2), 373–440.
- [Veličković *et al.*, 2017] Veličković, Petar, Cucurull, Guillem, Casanova, Arantxa, Romero, Adriana, Lio, Pietro, & Bengio, Yoshua. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- [Veličković *et al.*, 2018] Veličković, Petar, Fedus, William, Hamilton, William L, Liò, Pietro, Bengio, Yoshua, & Hjelm, R Devon. 2018. Deep Graph Infomax. *In: International Conference on Learning Representations*.
- [Wang *et al.*, 2016] Wang, Zhiqiang, Liang, Jiye, Li, Ru, & Qian, Yuhua. 2016. An approach to cold-start link prediction: Establishing connections between non-topological and topological information. *IEEE Transactions on Knowledge and Data Engineering*, **28**(11), 2857–2870.

- [Zhang & Chen, 2018] Zhang, Muhan, & Chen, Yixin. 2018. Link prediction based on graph neural networks. *Pages 5165–5175 of: Advances in Neural Information Processing Systems*.
- [Zhang & Yang, 2017] Zhang, Yu, & Yang, Qiang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*.
- [Zhou *et al.*, 2004] Zhou, Dengyong, Bousquet, Olivier, Lal, Thomas N, Weston, Jason, & Schölkopf, Bernhard. 2004. Learning with local and global consistency. *Pages 321–328 of: Advances in neural information processing systems*.
- [Zhou *et al.*, 2009] Zhou, Tao, Lü, Linyuan, & Zhang, Yi-Cheng. 2009. Predicting missing links via local information. *The European Physical Journal B*, **71**(4), 623–630.

초록

대다수의 실제하는 그래프 데이터는 시간에 따라 노드와 그 연결이 변화하고 필연적으로 콜드스타트 문제를 직면하게 된다. 콜드스타트 문제를 해결하기 위해서는 새로 발생한 콜드 노드들이 기존 그래프에 포함되기 위해 어떻게 초기 정보를 얻을 수 있는지가 매우 중요하다. 본 논문에서는 그래프 학습에서의 콜드스타트 문제를 제기하고 새롭고 현실적인 학습 세팅인 "확장된 준지도학습"을 제안한다. 확장된 준지도학습에서는 기존의 준지도 학습을 새로 소개되어 기존 그래프와의 연결성이 없는 콜드 노드들에도 적용한다. 새로운 학습 세팅을 위해 논문에서는 콜드 노드들의 분류를 위해 링크 예측을 함께 멀티테스크 목적으로 갖는 ColdExpand 모델을 제시한다. 또한 실험적으로 우리 모델이 추가적인 목적을 모델에 더해줌으로써 콜드 노드들의 링크 예측과 분류 문제에서 과거 모델들을 크게 상회하는 결과가 나옴을 증명한다.

주요어: 그래프 신경망, 콜드 스타트, 멀티테스크 학습, 준지도학습

학번: 2019-24053