Towards an Adaptive Skip-gram Model for Network Representation Learning

I-Chung Hsieh[1] and Cheng-Te Li[1]

[1]Department of Statistics, National Cheng Kung University

Abstract

The random walk process on network data is a widely-used approach for network representation learning. However, we argue that the sampling of node sequences and the subsampling for the Skip-gram's contexts have two drawbacks. One is less possible to precisely find the most correlated context nodes for every central node. The other is not easily controlled due to several hyperparameters. Such two drawbacks lead to higher training cost and lower accuracy due to abundant and irrelevant samples. To solve these problems, we compute the adaptive probability of random walk based on personalized PageRank, and propose an *Adaptive Skip-gram* (ASK) model without random walk process and negative sampling. We utilize k-most important neighbors for positive samples selection, and attach their corresponding PPR probability into the objective function. We demonstrate the improvement of our ASK model for network representation learning in tasks of link prediction and node classification. The results achieve more effective performance and efficient learning time.

1 Introduction

Network data is getting much attention due to modern issues like social media analytics, disease infection, and knowledge database. *Graph representation learning* (GRL) is an essential task to distill latent features from network data. While a network consists of a collection of links between nodes in a non-Euclidean space, the common purpose of GRL is to convert the highly complex network structure to a low-dimensional and explicit vector for each node, which is termed *node embedding*. Eventually, the embedding vectors can be used for downstream network analysis tasks, such as link prediction, node classification, and community detection.

To represent nodes in the context of network structure, the typical approach is matrix-based and edge-based, such as matrix factorization [9] and LINE [10]. While the matrix-based approach is costly in terms of computational efficiency and the edge-based model is shallow resulting in less effect, random walk-based methods along with neural network learning are popular and effective in recent years. DeepWalk [8] adopts the random walk mechanism and the Skip-gram model to efficiently learn node embeddings. The main idea comes from language model in word2vec [5]. Based on the random surfer that walks through highly correlated local neighbors surrounded by each target node, and Skip-gram model is able to truncate a context with inter-correlated words and updates node embeddings. node2vec [3] presents a biased random walk controlled by the hyperparameters of depth-first and breadth-first search. GENE [1] considers the group labels from the random walk's neighbors to preserve more information in node embeddings. DDRW [4] jointly optimizes the classification objective and the objective of random-walk-based embedding entities for better node classification. Extended studies further aim at learning node embeddings in attributed networks, in which ANRL [11] and DANE [2] are random walk-based approaches that also need the Skip-gram model. In short, one research direction of GRL is incorporating the Skip-gram model with random walk, which is widely validated being useful.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

In this work, we aim to revisit the Skip-gram model with random walk, and show how to improve its representation capability for network data. We argue that a random walk-based sampling-based method is easily influenced by random noise and several hyperparameters such as number of walks per node, walk length, and context size. These factors lead to the requirement of more learning samples, tedious hyperparameter tuning, and most importantly, the selection of irrelevant context nodes for every central node. It is because that the truncated context with the fixed length is not capable to depict the topological correlation (e.g., proximity) between the central node and each context node. Besides, the sampling frequency distribution of nodes occurring in the target's context would be less precise as the number of the samples is not enough. If we try to sample more samples to improve accuracy, we have to create the additional cost during training model. Therefore, we need a more precise mechanism to select representative neighbors for every node. This would be achieved by the estimation of adaptive probability in random walk process, along with some incorporation into the Skip-gram model.

To deal with the aforementioned issues, we leverage personalized PageRank [7] (PPR) that represents the convergent probability from root (central node) to any other nodes along a randomly sampled path. We can consider such a probability as the degree of correlation between two nodes as well as the exact node frequency in sampled node sequences. To be specific, by combining the PPR probability and the random walk process, we can derive the adaptive random walk probability indicating the structural correlation between two nodes so that we can accordingly select the most significant context nodes for every central node. Eventually by incorporating PPR into Skip-gram model, we develop the *Adaptive Skip-gram* (ASK) model.

The contribution of this work is three-fold. First, we simplify the complex random walk process by the probability of personalized PageRank. The hyperparameters of the original random walk in Skip-gram model can be combined as one. Second, technically, we improve the Skip-gram model via the estimated probability by proposed Adaptive Skip-gram (ASK) model, which emphasizes and exploits the correlation between nodes. Our model would precisely learn the correlation, and does not require the negative sampling that could lead to misleading embeddings and increase computational cost. Third, the experiments conducted on three different datasets in GRL tasks exhibit the improvement of our Adaptive Skip-gram model in link prediction and node classification. We also suggest an approximated version of the Adaptive Skip-gram model that can be used to achieve efficient but similar performance in the limited environment.

2 The Proposed Methods

2.1 Personalized PageRank (PPR)

Given a network $\boldsymbol{G} = (V, \boldsymbol{A})$, where V is the node set with n nodes ($|V| = n$), $\boldsymbol{A}$ is the adjacency matrix. A personalized PageRank value can be seen as the probability from a certain root r to another node v via a random walk-like process. The probability updating equation of personalized PageRank (PPR) is given by $\boldsymbol{\pi}_r^{(n)} = (1-\alpha)\boldsymbol{H}\boldsymbol{\pi}_r^{(n-1)} + \alpha \boldsymbol{e}_r$, where $\boldsymbol{\pi}_r^{(n)}$ is the probability vector from root r to each node at n-th step, $\boldsymbol{H} = \boldsymbol{D}^{-1}\boldsymbol{A}$ is the normalized adjacency matrix based on $\boldsymbol{A}$ and the degree matrix $\boldsymbol{D}$. In addition, $\alpha \in [0, 1]$ is the restart probability, and $\boldsymbol{e}_r$ is the one-hot encoding for the root. After some reformulation, the PPR matrix $\boldsymbol{\Pi}$ can be described as $\boldsymbol{\Pi} = \alpha(I - (1-\alpha)H)^{-1}$, where $\boldsymbol{\Pi}_{ij}$ means the probability of going to the node j from the root i. Note that we will use "root", "central node" and "target" interchangeably throughout this work.

2.2 Adaptive Skip-gram Model

Typical network representation learning methods with the Skip-gram model and random walk, such as node2vec [3] and DeepWalk [8], have three common phases. It contains sampling node sequences by random walk, generating contexts, and the Skip-gram model. The second is composing contexts of every node by setting central nodes and neighboring nodes from left to right in the derived node sequences. The third is applying the Skip-gram model. We argue such a process cannot precisely extract significant contexts for each node. It is because the random walk is not personally performed to generate the contexts for a central node. That said, the contexts, sampled via random walk, may be correlated with the central node. To be more specific, for nodes with high proximity scores to each other in a densely-connected community, they may not be each other's context. Repeated independent sampling via random walk from any nodes lead to such kind of outcome.

We aim at exploiting the probability values derived from personalized PageRank (PPR) to generate the contexts of every node. Since PPR values reflects the proximity degree from a root node to any other nodes in the network, we propose to leverage PPR for generating more representative

contexts so that the Skip-gram model can be constructed to produce better node embeddings. We will generate representative contexts by selecting top-k neighbors that possess the highest proximity values to the root/central node. In addition, we also want to simplify the process by allowing only one hyperparameter, rather than three typical hyperparameters, including context size, number of walks, and length of walk. The context size (i.e., number of contexts) can be regarded as the demand of the number of contexts to explain the central node. It should be proportional to the density and size of the central node's neighborhood. Hence, we make the parameter k play a role representing the maximum needed context size for learning a central node's embedding.

To estimate k, we need to figure out the occurrence frequency of every node in all random walk generated sequences. We think PPR can also be considered as the probability of sampling a node of any generated infinite-length sequence from the root. The summation of the scaled probability from all nodes to any node j can be simply regarded as the node frequency in all sequences, given by $\boldsymbol{f}_j = \sum_{i=1}^{n}(\boldsymbol{\Pi}_{ij})/n$. Given the average context size a_e as a hyperparameter used to obtain k, the total number of contexts for all nodes would be $a_e \times n$. Then the expected context size for each node can be derived as a vector: $a_e \times n \times \boldsymbol{f}_j = a_e \sum_{i=1}^{n}(\boldsymbol{\Pi}_{ij})$. We choose k to be the maximum expected context size for each node, given by $max_j(a_e \sum_{i=1}^{n} \boldsymbol{\Pi}_{ij})$. The next step is to attach the subsampling mechanism into the derivation of k. The subsampling in the original Skip-gram model utilizes the discarding probability $1 - (t_0(\boldsymbol{f}_w)^{-1} + \sqrt{t_0}(\boldsymbol{f}_w)^{-0.5})$ in [6], where t_0 is a chosen threshold (typically 10^{-5}), and $\boldsymbol{f}_w$ is frequency vector of each word in all sentences. We have obtained the node frequency vector $\boldsymbol{f}_j = (\sum_{i=1}^{n} \boldsymbol{\Pi}_{ij})/n$. Therefore, the subsampling probability would be $\boldsymbol{p}_{sub} = t_0(\boldsymbol{f}_j)^{-1} + \sqrt{t_0}(\boldsymbol{f}_j)^{-0.5}$, which smooths the sampling probability of high frequency nodes. As a result, the maximum expected context size with subsampling is given by $k = max_j(a_e \boldsymbol{f}_j \odot \boldsymbol{p}_{sub})$, where $\odot$ is Hadamard product. Such selection of k-most significant context nodes, along with PPR, simplifies the context generation and its hyperparameters.

We incorporate the Skip-gram model with the derived expected context size $a_e \boldsymbol{f}_c \odot \boldsymbol{p}_{sub}$. Consider the target node t and its k-most significant context nodes, we reconstruct the Skip-gram model to model the importance of each of its neighbors through PPR. Recall the original objective function $\sum_{c \in context(t)} log(\sigma(\boldsymbol{v}_t^T \boldsymbol{v}_c))$ for a pair of target t and its context node set $context(t)$, where $\boldsymbol{v}_t$ is the embedding for node t, and σ is the logit function. We replace original context nodes with nodes possessing k highest values in the subsampling PPR value matrix, given by $\{\boldsymbol{\Pi}_{sub}\}_{t*} = \{a_e \boldsymbol{Diag}(\boldsymbol{f}_c \odot \boldsymbol{p}_{sub})\boldsymbol{\Pi}\}_{t*}$, where $\boldsymbol{Diag}(\boldsymbol{v})$ is a diagonal matrix with diagonal entries equal to a vector $\boldsymbol{v}$. In other words, the values in PPR matrix is used in the objective function to point out which are significant neighbors. Eventually the objective of the modified Skip-gram model is given by: $\sum_{c \in \boldsymbol{T}_k(t)} log(\sigma(v_t^T v_c))\{\boldsymbol{\Pi}_{sub}\}_{tc}$, where $\boldsymbol{T}_k(t)$ is the set of k-most significant nodes of the target t.

In short, our model is learned by k context nodes of each central node. The proposed PPR-enhanced objective not only emphasizes the importance of context nodes without additional cost, but alleviates the problem of choosing irrelevant neighbors as contexts. Thus less correlated nodes in terms of proximity could be pushed away from one another in the learned embedding space. To some extent, such an effect is originally generated through *negative sampling*, and as a by-product in our model. Therefore, we choose not to perform negative sampling in our model.

2.3 An Approximated Approach for PPR

Since the derivation of PPR matrix requires $O(n^3)$ time complexity, our adaptive Skip-gram model may be less efficient when the network is large scale. Hence, we aim to provide an efficient alternative for the estimation of PPR matrix. Consider the inverse part of PPR matrix $(I - (1-\alpha)\boldsymbol{H})^{-1} = \boldsymbol{P}^{-1}$. The normalized matrix with bounded row sum (i.e., $\sum_j (1-\alpha)\boldsymbol{H}_{ij} < 1$) satisfies $||(1-\alpha)\boldsymbol{H}|| < 1$. Therefore, $\boldsymbol{P}$ can be approximated by the convergent sum of Neumann series $\lim_{m \to \infty} \sum_i^m ((1-\alpha)\boldsymbol{H})^i$. Given a small m, the complexity of the approximated PPR matrix would be decreased a lot due to the sparsity of $\boldsymbol{H}$. Besides, $((1-\alpha)\boldsymbol{H})^i$ can be regarded as the i-order proximity. Therefore, the approximated PPR matrix with a small m is capable to cover most of information for modeling.

3 Experiments

We conduct experiments to evaluate the effectiveness of our adaptive Skip-gram model for network representation learning. Three publicly available network datasets, Cora, Citeseer and Pubmed [1] are employed. The data sizes in (#nodes, #edges) are $(2708, 10556)$, $(3312, 9196)$, $(19717, 88651)$,

[1] Datasets available via `https://linqs.soe.ucsc.edu/data`

Table 1: AUC scores and time cost (seconds) for link prediction.

	Cora	Citeseer	Pubmed
SK	0.8902±0.0093 (123.00)	0.9135±0.0080 (147.30)	0.9340±0.0030 (264.01)
ASK	**0.9262**±0.0053 (19.53)	**0.9387**±0.0065 (22.04)	**0.9399**±0.0023 (242.94)
AASK (5)	0.8965±0.0091 (**18.18**)	0.9015±0.0101 (**21.09**)	0.9276±0.0027 (**162.36**)
AASK (10)	0.9110±0.0073 (19.73)	0.9168±0.0053 (21.44)	0.9360±0.0015 (242.46)
AASK (20)	0.9196±0.0035 (22.83)	0.9310±0.0094 (22.70)	0.9395±0.0015 (433.79)

Table 2: #(Positive pair) and the detailed time cost for link prediction.

	Cora	PT(s)	TT(s)	Citeseer	PT(s)	TT(s)	Pubmed	PT(s)	TT(s)
SK	3.6E+05	8.68	114.33	4.8E+05	8.88	138.39	7.0E+06	63.79	200.22
ASK	3.8E+05	1.32	18.21	3.4E+05	1.62	20.42	2.4E+06	114.52	128.42
AASK (5)	**3.5E+05**	**0.88**	**17.30**	**3.1E+05**	**0.91**	**20.18**	**2.1E+06**	**33.20**	129.15
AASK (10)	3.7E+05	1.91	17.82	3.2E+05	1.20	20.24	2.2E+06	114.97	**127.49**
AASK (20)	3.8E+05	4.64	18.18	3.3E+05	2.25	20.45	2.3E+06	305.21	128.58

respectively. We randomly choose 70%, 10%, and 20% edges as the training, validation, and testing sets. We also ensure the network is connected. The tasks include link prediction and node classification. We compare the performance for the original Skip-gram model (SK) with biased random walk [3], our Adaptive Skip-gram model (ASK), and PPR-Approximated Adaptive Skip-gram model (AASK(m)), where the order m of the Neumann series is given by three different sizes $\{5, 10, 20\}$. After obtaining the node embeddings, we use Hadamard product to derive the embedding vectors of node pairs. Then, we utilize logistic regression as the classifier and the area under the ROC curve (i.e., AUC score) as the evaluation metric. Due to page limit, the settings of hyperparameters and the results of node classification are provided in the supplementary material.

The results of link prediction are shown in Table 1 and Table 2. Table 1 shows AUC scores and time cost in seconds. Table 2 exhibits the number of training pairs without negative samples, and the detailed time in processing Time (PT) and training Time (TT). PT is the time cost of random walk process or PPR computation, and TT records the time from the first epoch to the epoch where the loss is convergent.

In Table 1, the results show both of ASK and AASK with higher m lead to better performance on AUC scores than SK. We think it is because we consider PPR to select representative contexts. Regarding the AASK, the time cost would increase dramatically and surpass than ASK because the iteration matrix is getting non-sparse. It suggests that AASK with $m = 5$ or 10 can be more appropriate than ASK when the cost of time is under the restriction.

In Table 2, it clearly demonstrates that random walk efficiently captures the network structure, especially for larger networks (i.e., PT on Punmed). Such results imply that the performance of random-walk sampling model is highly depended on the number of repetitive sampling. Besides, it also affects the time cost of the following training step. Instead, ASK utilizes the PPR probability weighting in the objective so that the learning volume of each epoch can be reduced.

4 Discussion

We design a more efficient and effect Skip-gram model ASK that requires no random walk for network representation learning. ASK overcomes the problems of the decision of multiple hyperparameters and non-efficient training for the original Skip-gram model. Since the hyperparameters, such as number of walks, and walk length, can influence on the performance, we derive the adaptive probability based on PPR, which is equivalent to the random walk process, to avoid the complex sampling process. Besides, the Adaptive Skip-gram model via the estimated probability of k-most significant nodes would precisely make the highly-correlated nodes close, and therefore the objective function can quickly achieve the convergence without negative sampling and even have better performance. We also consider an approximated method as a light version of Adaptive Skip-gram model with using small m, which has an efficient performance when the running environment is limited. The proposed Adaptive Skip-gram model can be seamlessly used for random walk Skip-gram based network representation learning models, such as node2vec and DeepWalk so that the efficiency and the effectiveness can get boosted.

References

[1] Jifan Chen, Qi Zhang, and Xuanjing Huang. Incorporate group information to enhance network embedding. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1901–1904. ACM, 2016.

[2] Hongchang Gao and Heng Huang. Deep attributed network embedding. In *IJCAI*, pages 3364–3370, 2018.

[3] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[4] Juzheng Li, Jun Zhu, and Bo Zhang. Discriminative deep random walk for network classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1004–1013, 2016.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[7] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[8] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[9] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 459–467, 2018.

[10] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[11] Zhen Zhang, Hongxia Yang, Jiajun Bu, Sheng Zhou, Pinggang Yu, Jianwei Zhang, Martin Ester, and Can Wang. Anrl: Attributed network representation learning via deep neural networks. In *IJCAI*, pages 3155–3161, 2018.

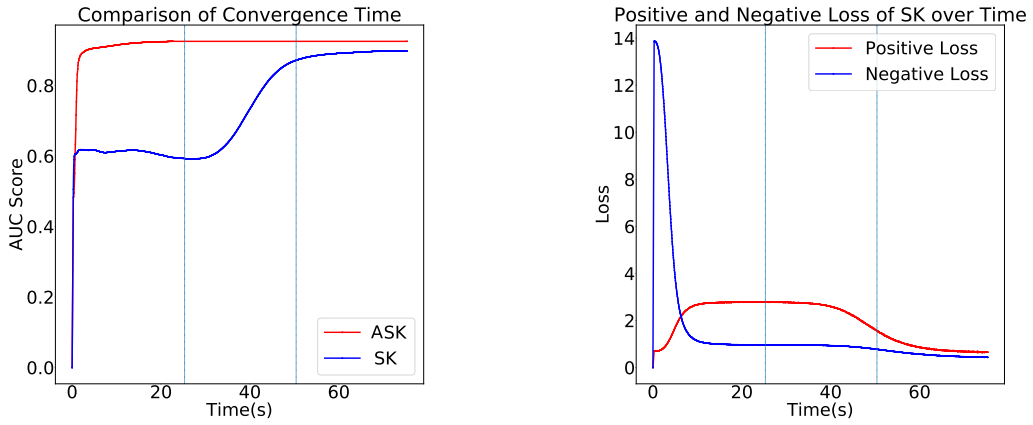
A Hyperparameters of SK and ASK for Link Prediction

The dimensionality of node embedding vector is set 128 for all methods, and all of models are trained by Adam optimizer with learning rate $= 0.001$. For the setting of SK, we set length window size $= 5$, the number of repeating walks $= 1$, and the walk length $= 80$ for random walk process. The number of negative samples is 20 for Cora and Citeseer and 5 for Pubmed.

For the settings of ASK and AASK, we set the default expected average context size $a_e = 25$, and the restart probability of PPR is set as $\alpha = 0.05$ for Cora and Citeseer and 0.07 for Pubmed.

B Convergence Analysis for SK and ASK

We analyze the convergence of SK and ASK. We also discuss the disadvantages of SK that our ASK can overcome. In Figure B.1, the testing AUC scores for link prediction on Cora data, and the loss of ASK and SK are displayed in (a) and (b), respectively. The vertical lines in the figures indicate the timestamps of the epoch of SK at 25.3 (sec) and 50.4 (sec) as the beginning of the 2-nd epoch and the 3-rd epoch. In Figure B.1a, we can clearly observe that the convergent time of ASK is less than one epoch of SK but SK would not start growing until the 2-nd epoch. We think that SK needs to balance the effect between the positive loss and negative one, as first shown in Figure B.1b. In the 1-st epoch, the model makes the negative loss decrease, but the positive loss is retained at the same level, and then focus on reducing the positive loss in the next epochs. In other words, since the correlated nodes are still be far away from each other, the accuracy would not be raised at the beginning. Though negative sampling help estrange the non-correlated nodes, it still has a trade-off in delaying the training efficiency. Our ASK utilizes a more precise selection of positive samples, and therefore avoiding the undesired effect of negative sampling.



(a) AUC scores vs. run time

(b) Loss of SK over time

Figure B.1: Convergence Analysis of SK and ASK.

C Label Classification of SK and ASK .

We also conduct node label classification task for SK and ASK. The number of labels for Cora, Citeseer and Pubmed are 7, 6 and 4, respectively. We first learn node embeddings from the network, and then employ one-vs-rest logistic regression classifier with L2 regularization on randomly select training and testing samples. The percentage of training set is varied from 10% to 90%. We utilize Micro-F1 and Macro-F1 as the evaluation metrics. Higher scores indicate better performance. The results of scores and time cost are shown in Figure C.2 and Table C.3. According to the scores, ASK has a slight improvement in accuracy for small networks, and the performance of ASK and SK on Pubmed are close because the sampling distribution for larger networks would be more well-approximating. It can be also apparently found that the run time of our ASK is significantly less than SK. Such results again prove the efficiency of ASK. In details, during the training time, the

time cost of ASK and SK are dropped. We think classification is the uncomplicated version of link prediction, which only needs to realize the relationship between nodes and rare labels. Therefore, the model can recognize the labels by learning the shallow structure. Especially, our PPR scores offer more significant candidates, so the time cost is clearly decrease.

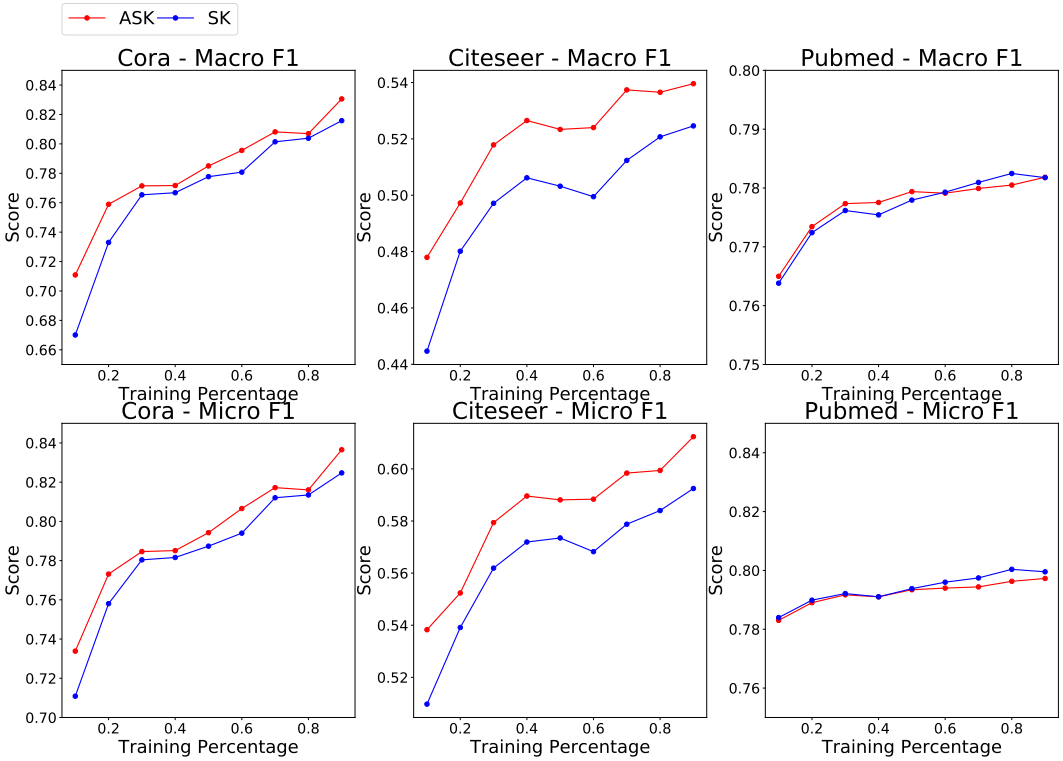


Figure C.2: Results of node classification for SK and ASK.

Table C.3: Total time cost in seconds, and the detailed time cost for node classification.

	Cora	PT(s)	TT(s)	Citeseer	PT(s)	TT(s)	Pubmed	PT(s)	TT(s)
SK	107.06	8.23	98.83	130.1	9.46	120.64	230.32	71.38	158.94
ASK	4.78	1.24	3.53	5.57	1.57	4	143.42	120.77	22.65