DEEP GRAPH INFOMAX

Petar Veličković*
Department of Computer Science and Technology
University of Cambridge
petar.velickovic@cst.cam.ac.uk

William Fedus
Montréal Institute for Learning Algorithms
Google Brain
liamfedus@google.com

William L. Hamilton
Montréal Institute for Learning Algorithms
Stanford University
wleif@stanford.edu

Pietro Liò
Department of Computer Science and Technology
University of Cambridge
pietro.lio@cst.cam.ac.uk

Yoshua Bengio†
Montréal Institute for Learning Algorithms
University of Montreal
yoshua.bengio@mila.quebec

R Devon Hjelm
Microsoft Research
Montréal Institute for Learning Algorithms
devon.hjelm@microsoft.com

ICLR 2019

Presenter: Jack Lanchantin

Deep Graph Infomax (DGI)

- General approach for **learning node representations** within graph-structured data in an unsupervised manner.
- DGI relies on maximizing mutual information between nodes (or groups of nodes) and a high-level summary of the graph.

Unsupervised Node Representation Learning

- **Key idea:** train an encoder so that nodes that are “close” in the input graph are also “close” in the representation space.
  - **Previous Works:** define “close” by random walks out from a chosen node
- **This Paper:** node learning based on mutual information, rather than random walks.

Background: Contrastive Methods

- Using a scoring function, train the encoder to increase the score on positive examples and decrease the score on negative samples.
  - E.g. Collobert & Weston 2008, Mikolov et. al. 2013 (Word2Vec)

Graph Based Unsupervised Learning

- Given
  - Set of node features $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, where N is # of nodes, $\mathbf{x}_i \in \mathbb{R}^F$
  - Binary adjacency matrix, $\mathbf{A} \in R^{NxN}$
- Objective
  - Learn an encoder, *E* such that $E(\mathbf{X},\mathbf{A}) = \mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N\}$, $\mathbf{h}_i \in \mathbb{R}^{F'}$

Local-Global Mutual Information Maximization

- DGI attempts to find node (i.e., local, or patch) representations that capture the global information content of the entire graph by maximizing local mutual information
- This allows for discovering and preserving similarities on the patch-level—for example, distant nodes with similar structural roles

Local-Global Mutual Information Maximization

- Graph-level summary vectors $\mathbf{s} = R(E(\mathbf{X}, \mathbf{A}))$
  - Readout function, R: $\mathbb{R}^{NxF} \rightarrow \mathbb{R}^{F}$
  - R used to summarize the obtained patch representations into a graph-level representation

- Discriminator D: $\mathbb{R}^{F} \rightarrow \mathbb{R}$, such that $D(\mathbf{h}_i, \mathbf{s})$ represents the probability scores assigned to this patch-summary pair (should be higher for patches contained within the summary)
  - Proxy for maximizing the local mutual information

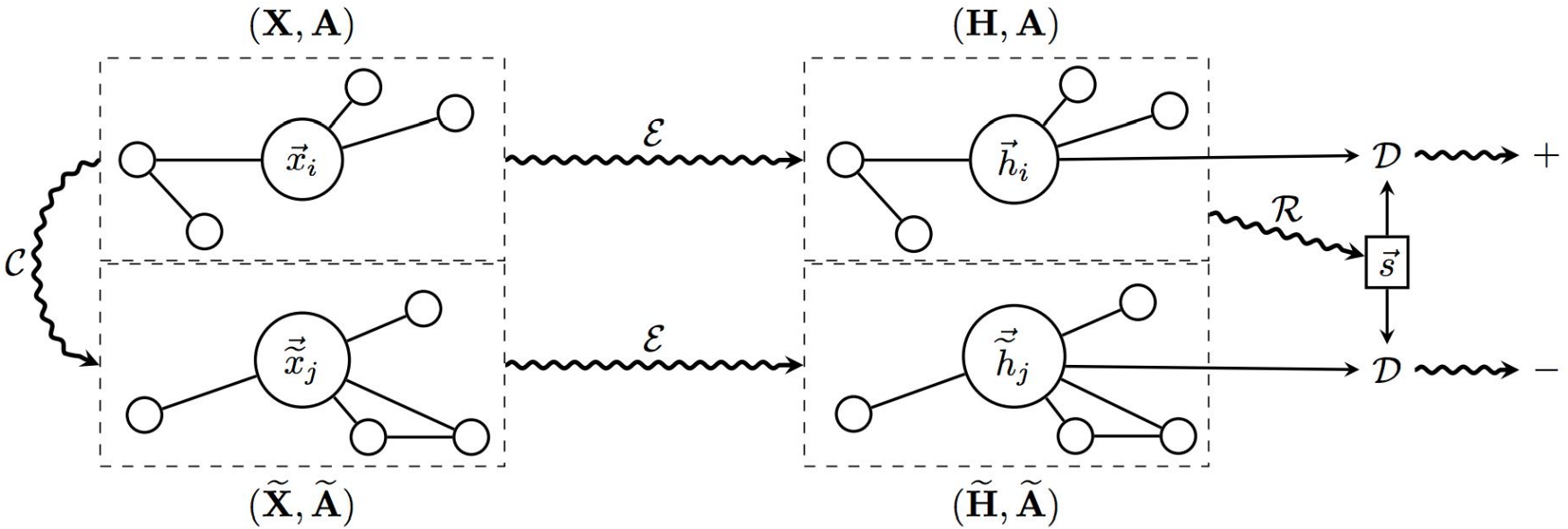
Sampling Negative Patches for MI Maximization

- Negative samples are provided by patch representations $\widetilde{h}_j$ of an alternative graph, $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{A}})$.
- Alternative graph is created from a corrupted original graph using corruption function *C:*
  - $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{A}}) = C(\mathbf{X}, \mathbf{A})$

Maximizing MI for DGI (standard binary cross entropy)

$$\mathcal{L} = \frac{1}{N+M}\left(\sum_{i=1}^{N}\mathbb{E}_{(\mathbf{X},\mathbf{A})}\left[\log\mathcal{D}\left(\vec{h}_i,\vec{s}\right)\right]+\sum_{j=1}^{M}\mathbb{E}_{(\widetilde{\mathbf{X}},\widetilde{\mathbf{A}})}\left[\log\left(1-\mathcal{D}\left(\vec{\widetilde{h}}_j,\vec{s}\right)\right)\right]\right.$$

Local-Global Mutual Information Maximization



Experiment 1: Transductive

Transductive learning. We utilize three standard citation network benchmark datasets—Cora, Citeseer and Pubmed (Sen et al., 2008)—and closely follow the transductive experimental setup of Yang et al. (2016). In all of these datasets, nodes correspond to documents and edges to (undirected) citations. Node features correspond to elements of a bag-of-words representation of a document. Each node has a class label. We allow for only 20 nodes per class to be used for training—however, honouring the transductive setup, the unsupervised learning algorithm has access to all of the nodes' feature vectors. The predictive power of the learned representations is evaluated on 1000 test nodes.

Experiment 1: Transductive

For the transductive tasks, we report the mean classification accuracy (with standard deviation) on the test nodes of our method after 50 runs of training (followed by logistic regression), and reuse the metrics already reported in Kipf & Welling (2016a) for the performance of DeepWalk and GCN, as well as Label Propagation (LP) (Zhu et al., 2003) and Planetoid (Yang et al., 2016)—a representative fully supervised random walk method. Specially, we provide results for training the logistic regression on raw input features, as well as DeepWalk with the input features concatenated.

Experiment 1: Transductive

Transductive (Classification Accuracy)

Available data	Method	Cora	Citeseer	Pubmed
X	Raw features	$47.9 \pm 0.4\%$	$49.3 \pm 0.2\%$	$69.1 \pm 0.3\%$
A, Y	LP (Zhu et al., 2003)	68.0%	45.3%	63.0%
A	DeepWalk (Perozzi et al., 2014)	67.2%	43.2%	65.3%
X, A	DeepWalk + features	$70.7 \pm 0.6\%$	$51.4 \pm 0.5\%$	$74.3 \pm 0.9\%$
X, A	Random-Init (ours)	$69.3 \pm 1.4\%$	$61.9 \pm 1.6\%$	$69.6 \pm 1.9\%$
X, A	**DGI** (ours)	$\mathbf{82.3} \pm 0.6\%$	$\mathbf{71.8} \pm 0.7\%$	$\mathbf{76.8} \pm 0.6\%$
X, A, Y	GCN (Kipf & Welling, 2016a)	81.5%	70.3%	79.0%
X, A, Y	Planetoid (Yang et al., 2016)	75.7%	64.7%	77.2%

Experiment 2: Inductive

Inductive learning on large graphs. We use a large graph dataset (231,443 nodes and 11,606,919 edges) of Reddit posts created during September 2014 (derived and preprocessed as in Hamilton et al. (2017a)). The objective is to predict the posts' community (*"subreddit"*), based on the GloVe embeddings of their content and comments (Pennington et al., 2014), as well as metrics such as score or number of comments. Posts are linked together in the graph if the same user has commented on both. Reusing the inductive setup of Hamilton et al. (2017a), posts made in the first 20 days of the month are used for training, while the remaining posts are used for validation or testing and are *invisible* to the training algorithm.

Experiment 2: Inductive

For the inductive tasks, we report the micro-averaged F_1 score on the (unseen) test nodes, averaged after 50 runs of training, and reuse the metrics already reported in Hamilton et al. (2017a) for the other techniques. Specifically, as our setup is unsupervised, we compare against the unsupervised GraphSAGE approaches. We also provide supervised results for two related architectures—FastGCN (Chen et al., 2018) and Avg. pooling (Zhang et al., 2018).

Experiment 2: Inductive

Inductive (F1 Scores)

Available data	Method	Reddit	PPI
X	Raw features	0.585	0.422
A	DeepWalk (Perozzi et al., 2014)	0.324	—
X, A	DeepWalk + features	0.691	—
X, A	GraphSAGE-GCN (Hamilton et al., 2017a)	0.908	0.465
X, A	GraphSAGE-mean (Hamilton et al., 2017a)	0.897	0.486
X, A	GraphSAGE-LSTM (Hamilton et al., 2017a)	0.907	0.482
X, A	GraphSAGE-pool (Hamilton et al., 2017a)	0.892	0.502
X, A	Random-Init (ours)	0.933 ± 0.001	0.626 ± 0.002
X, A	**DGI** (ours)	$\mathbf{0.940} \pm 0.001$	$\mathbf{0.638} \pm 0.002$
X, A, Y	FastGCN (Chen et al., 2018)	0.937	—
X, A, Y	Avg. pooling (Zhang et al., 2018)	0.958 ± 0.001	0.969 ± 0.002

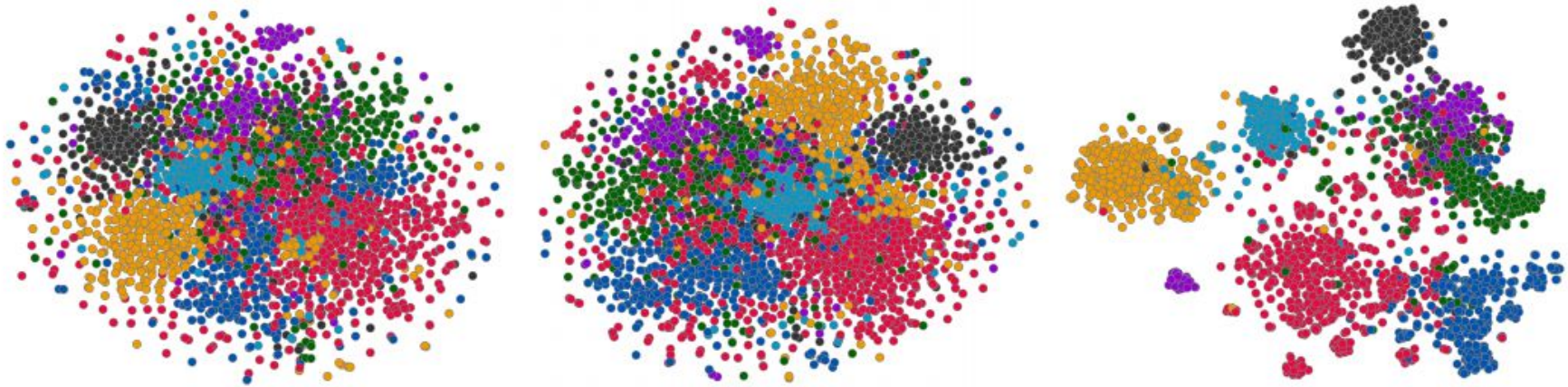


Figure 3: t-SNE embeddings of the nodes in the Cora dataset from the raw features (**left**), features from a randomly initialized DGI model (**middle**), and a learned DGI model (**right**). The clusters of the learned DGI model's embeddings are clearly defined, with a Silhouette score of 0.234.

Background: Mutual Information

- Measures the information that X and Y share: It measures how much knowing one of these variables reduces uncertainty about the other
  - E.g. if X and Y are independent, then knowing X does not give any information about Y and vice versa, so I(X;Y) = 0.

$$H(X) = \mathbb{E}_X[I(x)] = -\sum_{x \in \mathbb{X}} p(x) \log p(x).$$

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right)$$

$$\mathcal{R}(\mathbf{H}) = \sigma\left(\frac{1}{N}\sum_{i=1}^{N}\vec{h}_i\right)$$

$$\mathcal{D}(\vec{h}_i, \vec{s}) = \sigma\left(\vec{h}_i^{T}\mathbf{W}\vec{s}\right)$$