Multitask Graph Autoencoder

Phi Vu Tran
Strategic Innovation Group
San Diego, CA USA

Workshop on Relational Representation Learning, NIPS 2018, Montréal, Canada

Presenter: Arshdeep Sekhon
https://qdata.github.io/deep2Read

Overview

- Link Prediction and Node Classification(LPNC)
- input graph is a partially observed set of edges and nodes (or vertices), and the learning task is to predict the labels for edges and nodes

Overview

- Link Prediction and Node Classification(LPNC)
- input graph is a partially observed set of edges and nodes (or vertices), and the learning task is to predict the labels for edges and nodes
- **Multi-Task Graph Autoencoder (MTGAE)**: a shared representation of latent node embeddings from local graph topology and available explicit node features for LPNC
- trained end-to-end for the joint, simultaneous multi-task learning of unsupervised link prediction and semi-supervised node classification in a single stage

Notation

- input: graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of $N = |\mathcal{V}|$ nodes.
- adjacency matrix $\mathsf{A} \in \mathbb{R}^{N \times N}$
- For a partially observed graph, $\mathsf{A} \in \{1, 0, \textsc{unk}\}^{N \times N}$
- Optionally a matrix of available node features, i.e. side information $\mathsf{X} \in \mathbb{R}^{N \times F}$.

Problem Formulation: Link Prediction

- The aim of the MTGAE model $h(\mathrm{A}, \mathrm{X})$ is to learn a set of low-dimensional latent variables for the nodes $\mathrm{Z} \in \mathbb{R}^{N \times D}$
- that can produce an approximate reconstruction output $\hat{\mathrm{A}}$ such that the empirical error between A and $\hat{\mathrm{A}}$ is minimized, thereby preserving the global graph structure.

Method:Unsupervised Link Prediction

- $a_i \in \mathbb{R}^N$ be an *adjacency vector* of A that contains the local neighborhood of the *i*th node.

$$
\begin{aligned}
\text{Encoder} \qquad & z_i = g(a_i) = \mathrm{ReLU}\left(W \cdot \mathrm{ReLU}\left(Va_i + b^{(1)}\right) + b^{(2)}\right). \\
\text{Decoder} \qquad & \hat{a}_i = f(z_i) = V^T \cdot \mathrm{ReLU}\left(W^T z_i + b^{(3)}\right) + b^{(4)}. \\
\text{Autoencoder} \qquad & \hat{a}_i = h(a_i) = f(g(a_i)).
\end{aligned}
$$

- a_i is highly sparse, with up to 80 percent of the edges missing at random in some of our experiments, and the dense reconstructed output $\hat{a}_i$ contains the predictions for the missing edges.

Unsupervised Link prediction

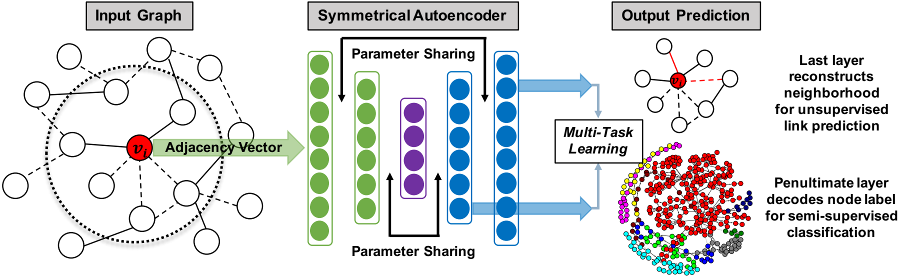

Figure: Schematic depiction of the Multi-Task Graph Autoencoder (MTGAE) architecture. *Left*: A partially observed graph with positive links (solid lines) and negative links (dashed lines) between two nodes; pairs of nodes not yet connected have unknown status links.

Method: Unsupervised Link prediction

- node features $\mathrm{X} \in \mathbb{R}^{N \times F}$ is available, then concatenate (A, X) to obtain an *augmented* adjacency matrix $\bar{\mathrm{A}} \in \mathbb{R}^{N \times (N+F)}$
- enable a shared representation of both graph and node features by way of the tied parameters $\{\mathrm{W}, \mathrm{V}\}$.
- During inference, the model takes as input an adjacency vector a_i and computes its reconstructed output $\hat{\mathrm{a}}_i = h(\mathrm{a}_i)$ for unsupervised link prediction.

$$\mathcal{L}_{\mathrm{BCE}} = -\mathrm{a}_i \log\left(\sigma\left(\hat{\mathrm{a}}_i\right)\right) \cdot \zeta - \left(1 - \mathrm{a}_i\right) \log\left(1 - \sigma\left(\hat{\mathrm{a}}_i\right)\right),$$

$$\mathcal{L}_{\mathrm{MBCE}} = \frac{\sum_i \mathrm{m}_i \odot \mathcal{L}_{\mathrm{BCE}}}{\sum_i \mathrm{m}_i}.$$

- $\zeta = 1 - \frac{\#\text{ positive links}}{\#\text{ negative links}}$

Semi-Supervised Node Classification: Node classification

- For a given augmented adjacency vector $\bar{\mathrm{a}}_i$, the model learns the corresponding node embedding z_i to obtain an optimal reconstruction.
- $\hat{\mathrm{y}}_i = \mathrm{softmax}(\tilde{\mathrm{z}}_i) = \frac{1}{\mathcal{Z}} \exp(\tilde{\mathrm{z}}_i)$, where $\mathcal{Z} = \sum_i \exp(\tilde{\mathrm{z}}_i)$ and $\tilde{\mathrm{z}}_i = \mathrm{U} \cdot \mathrm{ReLU}\left(\mathrm{W}^\mathsf{T}\mathrm{z}_i + \mathrm{b}^{(3)}\right) + \mathrm{b}^{(5)}$.

Multi-Task Learning

- in knowledge base completion and network analysis, the input graph has an incomplete set of edges and a small fraction of labeled nodes.
- combines the masked categorical cross-entropy loss for semi-supervised node classification with the MBCE loss for unsupervised link prediction:

$$\mathcal{L}_{\text{MULTI-TASK}} = \overbrace{-\mathrm{MASK}_i \sum_{c \in C} y_{ic} \log(\hat{y}_{ic})}^{\text{semi-supervised classification}} + \mathcal{L}_{\text{MBCE}},$$

Complexity

- $\mathcal{O}((N+F)DI)$, where N is the number of nodes, F is the dimensionality of node features, D is the size of the hidden layer, and I is the number of iterations. In practice, F, $D \ll N$, and I are independent of N. Thus, the overall complexity of MTGAE is $\mathcal{O}(N)$, linear in the number of nodes.

Datasets

Dataset	Nodes	Average Degree	$\lvert O^+\rvert : \lvert O^-\rvert$ Ratio	Node Features	Node Classes	Label Rate
Pubmed	19,717	4.5	1 : 4384	500	3	0.003
Citeseer	3,327	2.8	1 : 1198	3,703	6	0.036
Cora	2,708	3.9	1 : 694	1,433	7	0.052
Arxiv-GRQC	5,242	5.5	1 : 947	–	–	–
BlogCatalog	10,312	64.8	1 : 158	–	–	–

Baseline	Evaluation Task	Metric
SDNE [14]	Reconstruction	Precision@k
VGAE [8]	Link Prediction	AUC, AP
GCN [7]	Node Classification	Accuracy

Figure: Summary of datasets (*left*) and baselines (*right*) used in empirical evaluation. The notation $\lvert O^+\rvert : \lvert O^-\rvert$ denotes the ratio of positive to negative edges and is a measure of class imbalance. Label rate is the number of nodes labeled for training divided by the total number of nodes.

Results: reconstruction task

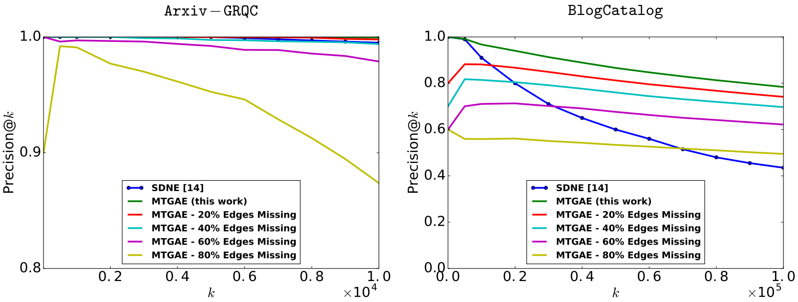

Figure: Comparison of precision@k performance between our MTGAE model and the related autoencoder-based SDNE model for the reconstruction task on the `Arxiv-GRQC` and `BlogCatalog` network datasets. The parameter k indicates the total number of retrieved edges.

Results: LPNC

The model takes as input an incomplete graph with 10 percent of the positive edges, and the same number of negative edges, missing at random and all available node features to simultaneously predict labels for the nodes and missing edges.

Method	Cora	Citeseer	Pubmed
Link Prediction			
MTGAE	**0.946**	**0.949**	0.944
VGAE [8]	0.920	0.914	**0.965**
Node Classification			
MTGAE	0.790	**0.718**	**0.804**
GCN [7]	**0.815**	0.703	0.790
Planetoid [15]	0.757	0.647	0.772

Figure: Link prediction performance is reported as the combined average of AUC and AP scores. Accuracy is used for node classification performance.

Conclusion

- Both link prediction and node classification task
- shown to work for only undirected non typed edges