Project Report : E0-270 Machine Learning

-By Abhishek Kumar, Anup Patel, Pragati Kumar singh and Shivam Chauhan

Adversarially Regularized Graph Autoencoder for Graph Embedding

- **Original Paper**: https://arxiv.org/pdf/1802.04407.pdf

  &#;-by Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao and Chengqi Zhang

Content

Introduction

- Graph representation has always been a good way to represent any relationship among the entities. Graphs are essential tools to capture and model complicated relationships among data.
- Graph embedding converts graph data into a low dimensional, compact, and continuous feature space. The key idea is to preserve the topological structure, vertex content, and other side information.
- In general graph embeddings are used to solve complex analytical tasks scuh as classification, clustering, and link prediction.
- Our target during graph embedding is to preserve the topological structure, vertex content. Our theme is not only to minimize the reconstruction error of topological structure but also to enforce that the latent embedding being learnt matches a prior distribution.

Related Works

- Our Graph embedding model is an exploratory work which is inspired by - Adversarially Regularized Graph Auto encoder for Graph Embedding by Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao and Chengqi Zhang.
- There were a lot of models/algorithms proposed for learning graph embeddings but most of the existing embedding algorithms typically focus on preserving the topological structure or minimizing the reconstruction errors of graph data, but they have mostly ignored the data distribution of the latent codes from the graphs,which often results in inferior embedding in real world graph data.
- Some earlier graph embeddings models are: DeeepWalk, node2vec, Hope, TADW etc.
- Our Model is motivated by the generative adversarial network (GAN) [Goodfellow et al.]

Datasets

- **Cora**

  The Cora dataset consists of 2708 scientific publications classified into one of seven classes. The citation network consists of 5429 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1433 unique words. The README file in the dataset provides more details.

- **Citeseer**

  The CiteSeer dataset contains 1504 machine learning documents with 2892 author references to 165 author entities. For this dataset, the only attribute information available is author name. The full last name is always given, and in some cases the author's full first name and middle name are given and other times only the initials are given.

- **Pubmed**

  The Pubmed Diabetes dataset consists of 19717 scientific publications from PubMed database pertaining to diabetes classified into one of three classes. The citation network consists of 44338 links. Each publication in the dataset is described by a TF/IDF weighted word vector from a dictionary which consists of 500 unique words. The README file in the dataset provides more details.

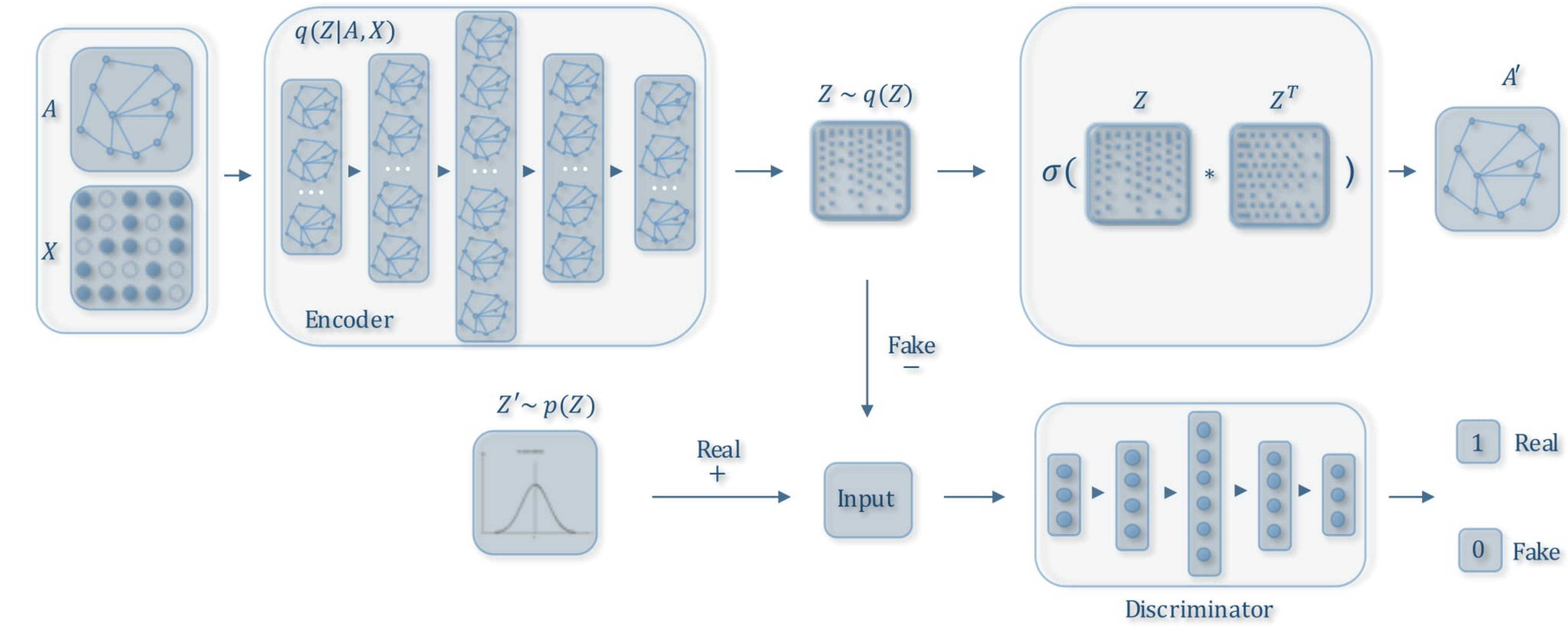
- The Cora dataset consists of Machine Learning papers. These papers are classified into one of the following seven classes:
  - Case Based
  - Genetic Algorithms
  - Neural Networks
  - Probabilistic Methods
  - Reinforcement Learning
  - Rule Learning
  - Theory

- The papers were selected in a way such that in the final corpus every paper cites or is cited by at least one other paper. There are 2708 papers in the whole corpus.

- After stemming and removing stop words we were left with a vocabulary of size 1433 unique words. All words with document frequency less than 10 were removed.

- Similarly the other two datasets were also preprocessed before learning the embeddings.

Model



$q(Z|A,X)$
A
X
Encoder
$Z \sim q(Z)$
Z
Z^T
$\sigma($
$*$
$)$
A'
Fake
$-$
$Z' \sim p(Z)$
Real
$+$
Input
1
Real
0
Fake
Discriminator

Model

- **Baseline Architecture:** The baseline model consists of two modules: (a) Graph auto encoder and (b) Adversarial Regularization.
- **Graph Convolutional encoder Model :** The baseline model uses a variant of the graph convolutional network (GCN) as a graph encoder. It extends the operation of convolution to graph data in the spectral domain, and learns a layer wise transformation by a spectral convolution function F(Z, A | W). It consists of two hidden layer Z_1 and Z_2 . Relu ( ) and linear activation functions are used for the first and second layers.
- Decoder Model : It is used to reconstruct the graph data. Baseline model reconstruct only the graph structure A and ignores the content information X .
- The objective function is to minimize the reconstruction error of the graph data.

Model

- **Adversarial Model**
- The key idea of Baseline model is to enforce latent representation Z to match a prior distribution.
- The adversarial model is built on a standard multilayer perceptron (MLP) where the output layer only has one dimension with a sigmoid function.
- The adversarial model acts as a discriminator to distinguish whether a latent code is from the prior p_z (positive) or from graph en-code G(X,A) (negative).
- The objective function of Adversarial Model is to minimize the cross-entropy cost for training the binary classifier.

Model

- Spectral convolution is calculated as:

$$\boxed{\mathbf{Z}^{(l+1)} = f(\mathbf{Z}^{(l)}, \mathbf{A}|\mathbf{W}^{(l)})}$$

$$f(\mathbf{Z}^{(l)}, \mathbf{A}|\mathbf{W}^{(l)}) = \phi(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{Z}^{(l)}\mathbf{W}^{(l)}),$$

- The Graph Encoder is constructed as follows:

$$Z^{(1)} = f_{Relu}(X, A|W^{(0)});$$

$$Z^{(2)} = f_{linear}(Z^{(1)}, A \mid W^{(1)})$$

Model

- Optimization objective of Encoder : $\mathbf{L_0 = E_{q(Z|(X,A))} [\log p(\hat{A}|Z)]}$
- Optimization objective of Adversarial Model :

$$-\frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_z}\log\mathcal{D}(\mathbf{Z}) - \frac{1}{2}\mathbb{E}_{\mathbf{X}}\log(1-\mathcal{D}(\mathcal{G}(\mathbf{X},\mathbf{A}))),$$

Evaluation and Metrics

- We have used AP (Average Precision) , F1, NMI, Accuracy, and Precision to evaluate our model.
- Evaluation metrics for Link Prediction :

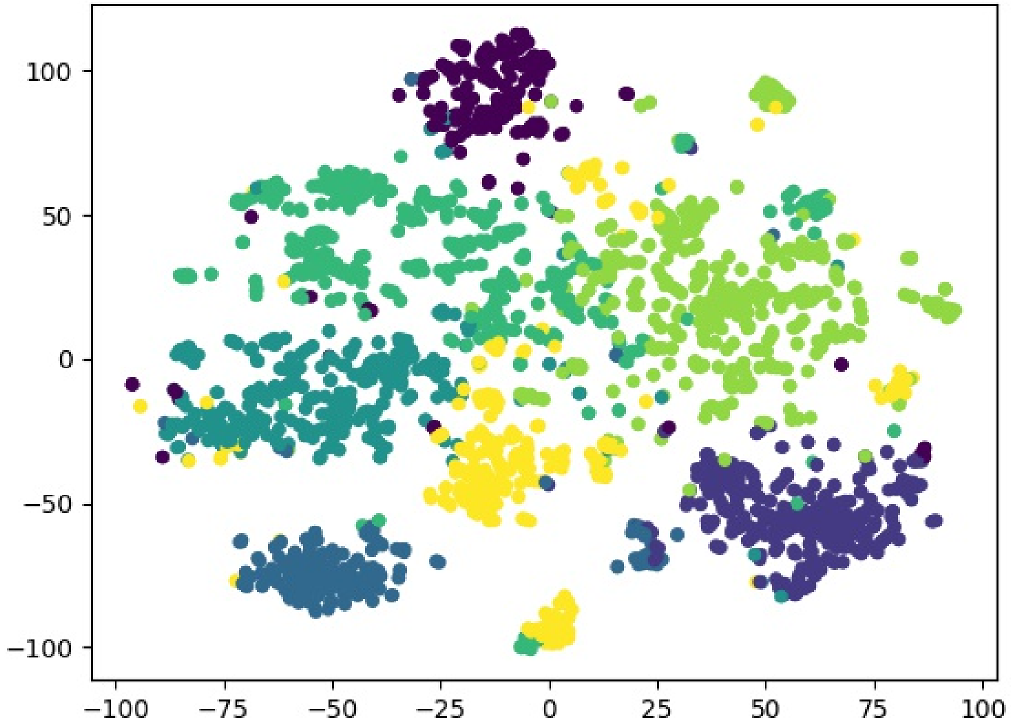
Model	Cora (AP Score)	Citeseer	Pubmed
ARGA	93.8	93.2	96.8

Evaluation

- Node Clustering :

Cora	Accuracy	NMI	Precision	
ARGA	0.668	0.489	0.680	

- Graph embedding visualization on 2d space



Conclusion

- Most existing graph embedding algorithms are unregularized methods that ignore the data distributions of the latent representation and suffer from inferior embedding in real world graph data.
- Experiment results demonstrated that the proposed algorithms ARGA outperform baselines in link prediction, node clustering, and graph visualization tasks.

Thank you