Labs
Machine Learning Course
Fall 2025

Westlake University
Department of Artificial Intelligence, SOE
Tao Lin
https://github.com/LINs-lab/course_machine_learning

Problem Set Lab 12, Dec 25, 2025 (Graph Embeddings)

Goals. The goals of this exercise are to:

- Understand the core components and principles of the Graph Encoder-Decoder Model (GraphEDM).
- Explain the mechanisms behind shallow unsupervised graph embedding methods like DeepWalk.
- Describe the functioning of spatial GNNs, particularly GraphSAGE, for inductive learning.
- Understand how deep unsupervised models like (V)GAE learn graph embeddings.
- Compare and contrast different graph embedding approaches for a specific downstream task.

Submission instructions:

- Please submit a PDF file to canvas.

Problem 1 (Label Propagation):

Consider the undirected graph $G = (V, E)$ with $V = \{1, 2, 3, 4\}$. The unweighted adjacency matrix A is given by:

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Nodes 1 and 4 are labeled: $y_1 = +1$ and $y_4 = -1$. Nodes 2 and 3 are unlabeled. We initialize the soft labels for unlabeled nodes as $f_2^{(0)} = 0$ and $f_3^{(0)} = 0$. The labels for labeled nodes are fixed: $f_1^{(t)} = y_1 = +1$ and $f_4^{(t)} = y_4 = -1$ for all iterations t.

1. Construct the row-normalized transition matrix P. Recall $P_{ij} = \frac{W_{ij}}{\sum_k W_{ik}}$, where W is the (weighted) adjacency matrix. Here, use A as W.
2. Perform one iteration of label propagation for the unlabeled nodes $i \in \{2, 3\}$:

$$f_i^{(t+1)} = \sum_j P_{ij} f_j^{(t)}$$

   Compute $f_2^{(1)}$ and $f_3^{(1)}$.

Problem 2 (Shallow Embeddings: Matrix Factorization):

Consider a graph with $N = 3$ nodes. We have learned low-dimensional embeddings $Z \in \mathbb{R}^{N \times L}$ for these nodes using a shallow embedding method. Let $L = 2$ and the learned embedding matrix be:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

We are using an outer product-based decoder to reconstruct the similarity matrix $\hat{W} = ZZ^T$. The original (ground truth) similarity matrix (e.g., adjacency matrix) is:

$$W = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

1. Compute the reconstructed similarity matrix $\hat{W} = ZZ^T$.
2. Compute the reconstruction loss $L_{G,RECON} = ||W - \hat{W}||_F^2$ (squared Frobenius norm of the difference).

Problem 3 (Graph Convolutional Networks (GCN)):

A common form for a GCN layer is:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$$

where $\tilde{A} = A + I_N$ is the adjacency matrix with self-loops, $\tilde{D}$ is the diagonal degree matrix of $\tilde{A}$ (i.e., $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$), $H^{(l)}$ is the matrix of node features/embeddings at layer l, $W^{(l)}$ is the learnable weight matrix for layer l, and σ is an activation function.

Let A be the adjacency matrix from Problem 1:

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Let $H^{(0)} = X$ be the initial node feature matrix.

1. Compute $\tilde{A} = A + I_N$.
2. Compute the degree matrix $\tilde{D}$ for $\tilde{A}$.
3. Explain the role of the term $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ in the GCN layer. What does multiplying by $H^{(l)}W^{(l)}$ achieve?

Problem 4 (DeepWalk Intuition):

DeepWalk is a shallow graph embedding method inspired by Word2Vec from Natural Language Processing.

1. Briefly explain how DeepWalk generates "sentences" from a graph.
2. How does DeepWalk then use these "sentences" to learn node embeddings, drawing an analogy to Word2Vec? What is the objective function typically optimizing?

Problem 5 (Graph Autoencoder (GAE)):

A Graph Autoencoder (GAE) uses a GCN as an encoder and a simple inner product decoder for link prediction. The encoder is typically a two-layer GCN:

$$Z = \text{GCN}(X, A) = \tilde{A}\text{ReLU}(\tilde{A}XW^{(0)})W^{(1)}$$

(Here, $\tilde{A}$ is used as shorthand for the normalized adjacency matrix $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ for simplicity in the formula structure, though different normalization schemes can be used). The decoder reconstructs the adjacency matrix $\hat{A}$ from the embeddings Z:

$$\hat{A}_{ij} = \sigma(Z_i Z_j^T)$$

where Z_i is the embedding of node i and σ is the sigmoid function. The reconstruction loss is often the binary cross-entropy:

$$L_{G,RECON} = -\sum_{i,j}(A_{ij}\log\hat{A}_{ij} + (1 - A_{ij})\log(1 - \hat{A}_{ij}))$$

1. What is the primary purpose of the GCN encoder in the GAE framework? What information does it aim to capture in the embeddings Z?
2. Explain why an inner product decoder $(Z_i Z_j^T)$ followed by a sigmoid is a suitable choice for reconstructing the adjacency matrix.
3. What does the binary cross-entropy loss penalize in this context?