Permutation Invariant Representations for Graph Convolutional Networks

Naveed Haghani¹, Maneesh Singh², Radu Balan¹
¹University of Maryland, ²Verisk Analytics

Introduction

We address the problem of graph classification and graph-wide regression using Graph Neural Networks (GNNs). GNNs utilize a graph’s adjacency matrix to learn a set of latent node embeddings. Permuting the order of the nodes in the adjacency leads to a different ordering of the rows in a latent embedding matrix. Therefore, a network that produces a single graph-wide estimate over these embeddings may not be consistent over permutations of the node ordering.

We introduce a permutation invariant mapping that takes a set of node embeddings produced by a GNN and produces an embedding for the entire graph that is invariant to any permutation of the nodes.

Rather than learn all possible permutations of every graph in a training set, we use a permutation invariant mapping such that any node ordering of a particular graph provides an identical result.

Problem

Given a set of node embeddings \( X \in \mathbb{R}^{n \times d} \) (where each row corresponds to the latent feature set for a given node), develop a stable and injective mapping that produces a representation \( \phi(X) \in \mathbb{R}^m \) that is invariant to row permutations of the input \( X \).

Take the equivalence relation – on \( \mathbb{R}^{n \times d} \) such that \( V, V' \in \mathbb{R}^{n \times d} \), \( \exists \pi \in S_n \), \( V' = \pi V \).

Find \( \phi: \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^m \) such that we have:

1. Permutation invariance: if \( V \rightarrow V' \), then \( \phi(V) = \phi(V') \)
2. Injectivity modulo permutations: if \( \phi(V) = \phi(V') \), then \( V \sim V' \)
3. Upper Lipschitz: if \( \phi(V) - \phi(V') \) is bounded by \( \| V - V' \|_2 \)
4. Lower Lipschitz: \( \min_{\pi \in S_n} \| \phi(V) - \phi(V\pi) \|_2 \leq \| \phi(V) - \phi(V') \|_2 \)

Theorem:

\[ \| \phi(V) - \phi(V') \|_2 \leq \| V - V' \|_2 \] for any \( V, V' \in \mathbb{R}^{n \times d} \). Here, \( \| \cdot \|_2 \) is the Euclidean norm.

*Ordering Approach*

1. Introduce redundancy into the embeddings by concatenating additional columns out of linear combinations of the rows of \( X \):

\[ X = \phi(X) \cdot \Pi ] \]

\( \Pi \) is a permutation matrix.

2. Order each column in descending order:

\[ \lambda(x) = (x_{n_1}, x_{n_2}, \ldots, x_{n_d}) \]

where \( n_1 \leq n_2 \leq \ldots \leq n_d \) and \( X' = \lambda(X) \).

*Kernel Approach*

1. Take a set of \( m \) kernel vectors:

\[ k = 1, \ldots, m \]

\[ a_i \in \mathbb{R}^d \] for \( i = 1, \ldots, m \).

2. Take the following kernel scheme:

\[ v_i: \mathbb{R}^d \rightarrow \mathbb{R} \]

\[ v_i(a_i, x) = \exp \left( -\frac{\| x - a_i \|^2}{\sigma_i^2} \right) \]

\[ \sigma_i^2 = E_i (\| \xi_i \|^2) \]

3. Let \( x^j \in \mathbb{R}^d \) represent the \( j \)th row of \( X \) (transposed to a column vector), then we produce \( \phi(X) \in \mathbb{R}^m \)

\[ \phi(x^j) = \sum_{i=1}^{m} v_i(a_i, x^j) \]

Problem consists of \( m = (1 + d - 3/2)n \).

Experiments & Data

Experiments were run on the qm9 molecular dataset. The data consists of 134 thousand chemical compounds along with 13 computational derived quantum chemical properties for each compound. We employ our network to perform regression over these values, specifically we look here at the electron energy gap \( \Delta (eV) \).

Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Loss Holdout, ( d = 50 )</th>
<th>Loss Holdout Permuted, ( d = 50 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Connected Network</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td><em>Kernel Approach</em></td>
<td>0.5</td>
<td>0.6</td>
</tr>
</tbody>
</table>

References

8. Li, Y.; Turbow, O.; Brodskiy, M.; and Zemel, R. 2016. Coded Graph Sequence Neural Networks.