Machine Learning on Publicly Available Semantic Graph Data for Interpretable QSAR Modeling

Joseph D. Romano*, Yun Hao*, and Jason H. Moore†
Institute for Biomedical Informatics, University of Pennsylvania,
Philadelphia, Pennsylvania 19104, United States
†*E-mail: jhmoore@upenn.edu*
* *These authors contributed equally.*

This is a placeholder for the real abstract. Please write each sentence on its own line in the LaTeX source file for easier tracking in Git.

Keywords: Toxicology; Graph machine learning; QSAR; Artificial Intelligence.

1. Introduction

Testing.

2. Methods

2.1. *Data sources*

The data used in this study come from a new data resource for computational toxicology, named ComptoxAI. ComptoxAI includes a large graph database containing many entity and relationship types that pertain to translational mechanisms of toxicity, all of which are sourced from third-party public databases (including PubChem, Drugbank, the US EPA's Computational Toxicology Dashboard, NCBI Gene, and many others). We extracted the subgraph from ComptoxAI's graph database defined as all nodes representing chemicals, genes, and toxicological assays, as well as the complete set of edges linking nodes of those types. A metagraph describing the node and edge types in the subgraph is shown in [FIGURE XXX].

2.2. *Heterogeneous graph neural network*

We constructed a heterogeneous graph neural network (GNN) architecture for the graph ML experiments. Since our approach uses multiple entity types (e.g., chemicals, genes, assays) in the same graph—each with possibly different sets of node features, and linked by multiple semantically distinct edge types—the architecture extends the common GCN model to learn separate message passing functions for each edge type. We use a message passing strategy similar to GraphSAGE,[1] where signals propagated from adjacent nodes are aggregated using their arithmetic mean:

$$\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W} \cdot \mathrm{MEAN}\left(\{\mathbf{h}_v^{k-1}\} \cup \{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\}\right)\right) \tag{1}$$

where $\mathbf{h}_v^k$ is an encoded representation of the input vector $\mathbf{x}$ of node v at layer k of the network, $\mathbf{W}$ is a weight matrix, and $\mathcal{N}(v)$ is the neighborhood of all nodes adjacent to v. Each layer in the network ‘pulls’ information from an increasingly wider radius around each node v. σ is an activation applied to the output of each layer; we use the leaky ReLU function:

$$\sigma(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases} \tag{2}$$

2.2.1. *Node prediction model*

2.2.2. *Edge prediction model*

2.3. *Baseline QSAR models*

3. Results

4. Discussion

5. Conclusions

6. Code availability

All source code pertaining to this study is available on GitHub at `https://github.com/JDRomano2/psb-gnn`, and in a ‘frozen’ version on FigShare at [XXX].

References

1. W. L. Hamilton, R. Ying and J. Leskovec, Inductive representation learning on large graphs, 1025 (2017).