Applying Graph Neural Network to Chemical Enantioselective Reactions

Haoyu Fan, Rye Zhang, Yiwei Zhang, Yuxuan Peng

1. Abstract

Predicting enantioselectivity of a chemical reaction (the ratio of R and S product formed) is critical to chemistry research. In this project, we applied GNN models to a dataset of catalytical reaction dataset. We discussed test set sampling techniques for small dataset we had, and, after hyperparameter tuning and investigate of different graph convolution layers, we achieved a mean absolute error of 4.87% in predicting the ratio of enantioselectivity.

2. Introduction

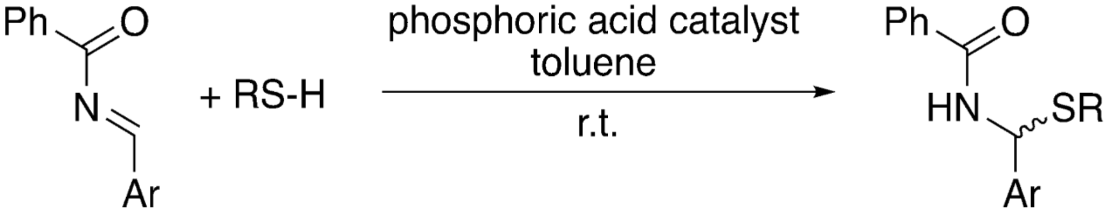
Enantioselectivity, the degree to which one enantiomer of a chiral product (R or S product) is preferentially produced in a chemical reaction, is an important factor in organic chemistry research, because sometimes only one enantiomer is biologically active or synthetically valuable. In enantioselective reactions, the ratio of R and S products is usually determined by the structure of the catalyst, so that the mechanism to form one enantiomer is favored. Being able to predict such selectivity is very helpful for the discovery of new reactions.

Traditionally, the discovery of useful ligands to a new reaction is based on the domain knowledge of chemists, who usually need to start from a potentially good structure and exhaustively search for improvement that gives the desired result. Being able to predict enantioselectivity based on molecular structure will greatly boost the efficiency of this research. In this project, we would like to use GNN, a model naturally suitable for molecules, to predict the enantioselectivity, given a dataset of a specific type of reaction.

3. Dataset

3.1 The source and format of dataset

The nature of chemical reaction setup determines that it is usually difficult to collect large dataset. However, because of the development of new instruments like high-throughput equipment, it is now possible to generate dataset at size of 1000~10000 reactions, even though that is still small compared to NLP or CV. In this project, we used a data set collected by Scott Denmark group of the following reaction[1]:



In this reaction, there are three molecules that are variables: the two on the left-hand side of the arrow, and the catalyst. The experimental results for enantioselectivity were rigorously collected for ~1000 reactions (5 * 5 * 40, by varying the Ar group, R group, and catalyst structures themselves).

3.2 Problem formulation

We have mapped the structure of the molecules in the dataset into graphs via a chemical language system called SMILES (Simplified Molecular-Input Line-entry System), or we one-hot encoded the molecules, which will be discussed in the following sessions.

The target of regression is a number of ratio of two possible products determined by an energy difference. However, we treated the problem as a binary classification with probabilities, because the proportion of two products add up to 1.

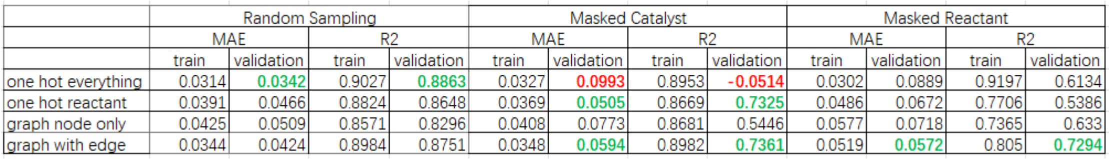
The node and edge features corresponding to the nature of the atoms and bonds were generated using the code from Coley et al,[2] based on some chemical facts of the molecules.

4. Results

4.1 Baseline and data representation

The graph models investigated in the project have the same basic structure: the graph representation of molecules underwent graph convolution layers, global meaning pooling, a fully connected layer and finally sigmoid to map the result into a binary probability. To find the appropriate baseline model as well as to account for the fact there are limited numbers of molecules in the dataset—5 for each reactant and 43 catalysts, we tried to one-hot encode the chemicals and feed it into an MLP besides investigating the graph neural networks. We also researched the model where only the reactants are one-hot encoded, and the catalysts are put in graphs. The one-hot information was concatenated to the vectors after graph global pooling before fully connected layer.

The above-mentioned ways of representing the molecules were compared using crudely tuned models. The convolution layers used for node-information only model is GCNConv in PyTorch, and GINEConv of the one with edge information. The results were summarized in Table 1. under “random sampling”, meaning that the train/test split was the conventional random approach. Each result was an average of 5 runs. The metrics used were MAE (mean absolute error) and the R2 of predicted vs true values of the test set.



	Random Sampling				Masked Catalyst				Masked Reactant			
	MAE		R2		MAE		R2		MAE		R2	
	train	validation	train	validation	train	validation	train	validation	train	validation	train	validation
one hot everything	0.0314	0.0342	0.9027	0.8863	0.0327	0.0993	0.8953	-0.0514	0.0302	0.0889	0.9197	0.6134
one hot reactant	0.0391	0.0466	0.8824	0.8648	0.0369	0.0505	0.8669	0.7325	0.0486	0.0672	0.7706	0.5386
graph node only	0.0425	0.0509	0.8571	0.8296	0.0408	0.0773	0.8681	0.5446	0.0577	0.0718	0.7365	0.633
graph with edge	0.0344	0.0424	0.8984	0.8751	0.0348	0.0594	0.8982	0.7361	0.0519	0.0572	0.805	0.7294

Table 1. Comparison of different input representations

The one-hot encoding approach behaved better than the graph neural networks in the random sampling of test set. The reason is most probably the small size of the dataset, such that the high expressive power of graph neural nets was not needed, and the difficulty in training GNN leads to worse performance than one-hot encoding and MLP. However, it is intuitive that one-hot encoding can not generalize to new molecules. Therefore, such results inspired us to experiment test set sampling other than random split of the whole dataset, which will be discussed in the next section.

4.2 Test set sampling

Since the dataset is combinatory, meaning every combination of 5 reactants A, 5 reactants B, and 43 catalysts produce one reaction result, the test set generated by random sampling may include molecules already seen by the model in the training set. Even though this is not meaningless, as sometimes chemists do need to generalize known reaction conditions to new combination of reactants, it is nevertheless more desirable to find a model with more generalizability. Therefore, we masked out some catalysts or reactants from the training set and tested the models' performance on the reactions related to these masked molecules in the test set. The results were summarized in Table 1., and the validation loss curves were plotted in Figure 1. The one hot-encoding approach behaved much worse than the graph models in this setting of test set sampling, proving the better generalizability of the latter models.

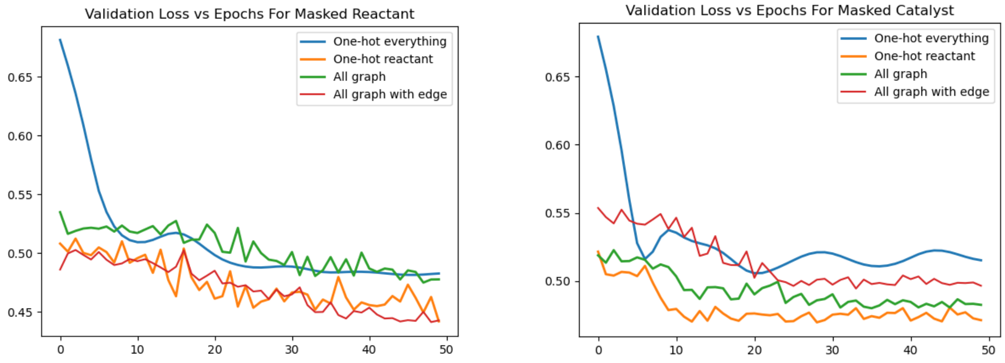


Figure 1. Validation loss of masked reactant and masked catalyst test set.

4.3 Hyperparameter tuning

Based on our knowledge of previous sections, we have performed hyperparameter tuning on the graph model where all molecules are treated as unconnected component of the single graph with edge information included. The model tuned used GINEConv as convolution layer. The results were shown in Table 2., and some representative curves were selected and plotted in Figure 2. The model with more layers did not necessarily perform better than one with less layers, probably due to small dataset.

LEARNING_RATE	DROP_OUT	NUM_LAYER	Test_MAE
0.001	0	2	0.0793
0.001	0	5	0.0704
0.001	0.1	2	0.0671
0.001	0.1	5	**0.0590**
0.001	0.5	2	0.1017
0.001	0.5	5	0.0902
0.0075	0	2	0.0787
0.0075	0	5	0.0656
0.0075	0.1	2	0.0763
0.0075	0.1	5	0.0635
0.0075	0.5	2	0.0669
0.0075	0.5	5	0.0811

Table 2. Hyperparameter Tuning.

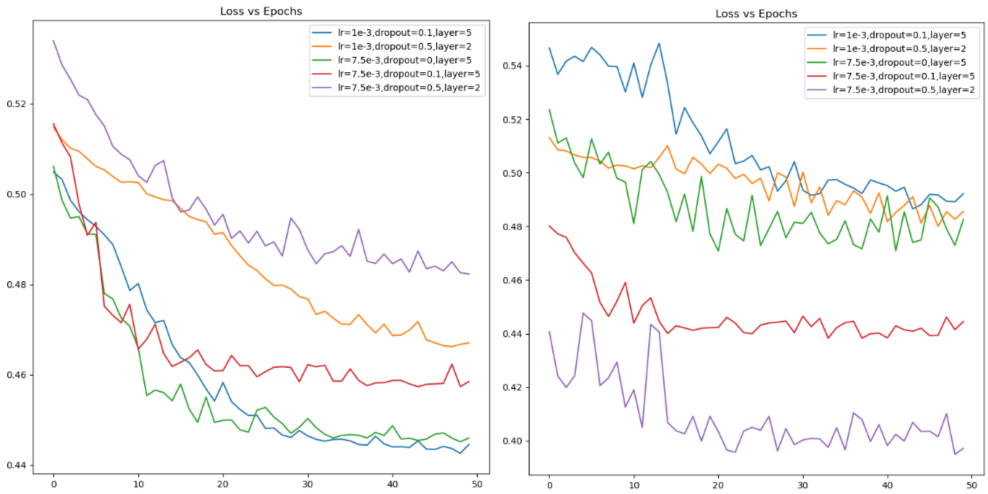


Figure 2. Training and validation loss for hyperparameter tuning, left being training loss.

4.4 Investigation of different convolution layer

We also researched the performance of the model using different graph convolution layers implemented in PyTorch. Specifically, we compared the model using

ChebConv,[3] fast localized spectral filtering, and TransformerConv[4] that brings in attention. The result was shown in Table 3. and Figure 3. It is noticeable that these two layers render the loss to converge faster than the normal edge convolution in GINEConv. The Chebychev convolution did give the least overfitting.

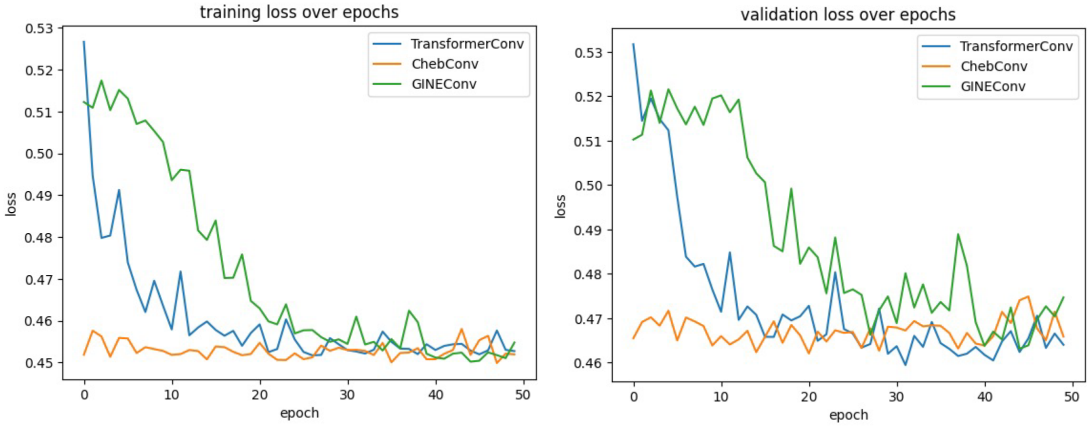


Figure 3. Comparison of training curves convolution layers.

conv	train MAE	train R2	test MAE	test R2
TransformerConv	0.0409	0.8779	0.0524	0.7678
ChebConv	0.0437	0.8631	0.0487	0.7997
GINEConv	0.0388	0.8878	0.0531	0.7536

Table 3. Comparison of performance of convolution layers.

5. Conclusion

Based on the discussion above, we have successfully applied graph neural networks to a chemistry problem of predicting enantioselectivity. We have formulated the problem as a classification with binary probability. Different data representations were compared, and to put all molecules in graph gave the best generalizability in masked catalyst and masked reactant sampling. After hyperparameter tuning and research about different convolution layer, we have achieved a best mean absolute error of ratio of 0.0487, meaning our prediction of ratio has an error as low as 4.87%.

6. References

(1) Zahrt, A. F., Henle, J. J., Rose, B. T., Wang, Y., Darrow, W. T., & Denmark, S. E. (2019). Prediction of higher-selectivity catalysts by computer-driven workflow and machine learning. *Science (New York, N.Y.)*, *363*(6424), eaau5631.

(2) Coley, C. W., Jin, W., Rogers, L., Jamison, T. F., Jaakkola, T. S., Green, W. H., ... & Jensen, K. F. (2019). A graph-convolutional neural network model for the prediction of chemical reactivity. Chemical science, 10(2), 370-377.

(3) Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems, 29.

(4) Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., & Sun, Y. (2020). Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*.