MULTI-MODEL ENSEMBLE ON HYPERGRAPH

Xinwei Zhang
School of Software, THUIBCS
Tsinghua University

Yifan Feng
School of Software, THUIBCS
Tsinghua University

Yue Gao
School of Software, THUIBCS
Tsinghua University

ABSTRACT

This paper introduces an ensemble learning approach, HyperFusion, specifically designed for Graph Neural Network (GNN) models. HyperFusion treats the multiple outputs of GNN models as features corresponding to different modalities and employs a hypergraph to efficiently fuse data from these modalities for model integration. Experimental results across multiple datasets demonstrate the effectiveness of the HyperFusion method. This approach is crucial for enhancing model performance, stability, and generalization capabilities.

1 Introduction

In recent years, Graph Neural Networks (GNNs)[1, 2, 3, 4] have witnessed extensive applications across various domains. By modeling the relationships between samples pairwise, GNNs can effectively capture the topological relationships among samples. Leveraging convolution in the spectral domain[2, 3] or message propagation in the spatial domain[5, 6, 7], GNNs can yield features and prediction outcomes of higher quality compared to other non-GNN networks.

Nevertheless, a singular GNN model is susceptible to the risk of overfitting[8]. In contrast, ensemble models can mitigate the risk of overfitting by aggregating predictions from multiple models[9, 10], thereby enhancing the model's generalization ability, reducing model variance, and improving stability and reliability. This approach further contributes to enhanced model performance and reduced error rates. Consequently, researchers have extensively investigated methodologies for integrating GNN models to address these concerns.

The determination of weights for individual sub-models during the model integration process is a crucial research issue, as the adjustment of weights directly impacts the performance of the final integrated model. Appropriate weight allocation can render the performance more robust and enhance the generalization ability of the integrated model. Commonly employed methods include voting and weighted averaging, among others. However, these methods introduce strong priors for weight determination; for instance, the voting method assumes equal weights for all models, while the weighted averaging method predefines the weights for each model. In contrast, adaptive model weight assignment algorithms can dynamically adjust weights based on the task, data, and model prediction results. This adaptive approach can yield a more robust and higher-performing integrated model.

Building upon the aforementioned analysis, we propose the HyperFusion (Multi-Model Ensemble on Hypergraph) method, which treats the output of models as multimodal data and utilizes a hypergraph to fuse and integrate this multimodal data, ultimately yielding the final prediction results. Initially, the term "multimodal data" refers to the requirement in the OGB competition[11] where models are tasked with providing predictions for positive/negative edges (link prediction task) or predictions on the validation/test sets. These predictions can be regarded as multimodal features corresponding to each model. It is important to note that hyperparameter tuning on the test set was not performed; rather, a non-parametric post-processing operation was applied to the model outputs during testing. Subsequently, for the multimodal data associated with each model, hypergraph fusion is employed. As a generalization of a graph, a hypergraph includes hyperedges connecting any number of nodes. This property allows the hypergraph to capture and characterize high-order correlation information among different models[12].

2 Preliminary: Hypergraph

Before delving into the specific method, let us provide a brief overview of relevant concepts related to hypergraphs. A hypergraph is typically represented by a triplet $\mathcal{H} = (V, E, W)$, where V is the set of nodes, $E \subset 2^V$ is the set of hyperedges, and W is a diagonal matrix representing the weights of the hyperedges. The hypergraph's incidence matrix, denoted as H, has elements H_{ij} indicating the association between the i-th node and the j-th hyperedge:

$$H_{ij} = \begin{cases} 1 & \text{if } v_i \in e_j \\ 0 & \text{else} \end{cases}. \tag{1}$$

This representation encapsulates the structural characteristics of hypergraphs, where hyperedges can connect arbitrary subsets of nodes, allowing for the exploration of high-order relationships within the data.

3 Method

HyperFusion first constructs a corresponding sub-hypergraph for each modality and subsequently integrates these sub-hypergraphs to form a multimodal hypergraph. Utilizing message passing on the multimodal hypergraph, the model fuses the multimodal features, ultimately yielding the final prediction results.

3.1 Sub-Hypergraph Construction

Let X_k^m denote the features of the m-th modality of the k-th sub-model. d^m represents the feature distances between different models pairwise on the m-th modality:

$$d_{i,j}^m = \mathrm{DIST}(X_i^m, X_j^m). \tag{2}$$

Equation 2 employs cosine distance in the experiments. For various tasks or datasets, alternative distance metrics, such as Euclidean distance or Mahalanobis distance, can also be considered.

Subsequently, for each node, the selection of nearby nodes to form a hyperedge is performed. There are various methods for this selection, including those based on k-NN (k-nearest neighbors)[13] or ϵ-ball criteria[14]. In the k-NN approach, the distances from the node to all other nodes are sorted, and the closest k nodes are selected. Alternatively, in the ϵ-ball approach, a threshold ϵ is defined, and all nodes within this distance are selected.

Through the aforementioned operations, we obtain the hypergraph corresponding to the m-th modality, denoted as $\mathcal{H}^m = (V^m, E^m, W^m)$.

3.2 Hypergraph Fusion

The purpose of hypergraph fusion is to integrate the sub-hypergraphs corresponding to different modalities, resulting in a multimodal hypergraph. Within the set of sub-hypergraphs $\mathcal{H}^m$ obtained in the previous step, the number of nodes in each hypergraph is consistent, i.e., equal to the number of models. Consequently, a multimodal hypergraph can be constructed based on the sub-hypergraph set, where the node set keeps constant, i.e.:

$$V = V^1 = \cdots = V^m, \tag{3}$$

and the hyperedge set is the union of the hyperedge sets from the sub-hypergraph set:

$$E = \bigcup E^m. \tag{4}$$

The hyperedge weigth matrix is the combination of W^m:

$$W = \mathrm{DIAG}(W^1, \cdots, W^m). \tag{5}$$

The resulting hypergraph integrates the correlation information between multiple modalities from different models, allowing for a more accurate characterization of high-order correlations among different models.

3.3 Message Passing on Hypergraph

The message passing step on the hypergraph integrates the predictions of the models, i.e., the features of a particular modality, to obtain the final integrated prediction results.

We first compute the normalized adjacency matrix based on the hypergraph's incidence matrix:

$$A = D_v^{-\frac{1}{2}} H^\top D_e^{-1} W H D_v^{-\frac{1}{2}}, \tag{6}$$

where D_v represents the degree matrix of nodes, and D_e represents the degree matrix of hyperedges.

Next, we use the normalized adjacency matrix A to fuse the features of the m-th modality, resulting in the integrated prediction results:

$$X' = AX^m, \tag{7}$$

where X^m represents the matrix obtained by stacking the m-th modality features of different models.

4 Experiments

In order to validate the effectiveness of the proposed method, experiments were conducted on the OGB competitions' datasets ogbl-collab[15], ogbl-ddi[16], ogbg-molhiv[17], and ogbg-molpcba[17]. The first two datasets are used for link prediction tasks, while the latter two are employed for graph property classification tasks. The experiments were conducted on an RTX 3080 GPUs. The results on the four datasets are presented in Table 1, Table 2, Table 3, and Table 4. From the results, it can be observed that the proposed HyperFusion method consistently outperforms existing state-of-the-art methods, achieving superior performance.

Table 1: Performance comparison on ogbl-collab dataset.

Method	Hits@50
HyperFusion	0.7129±0.0018
GIDN@YITU[18]	0.7096±0.0055
PLNLP+SIGN	0.7087±0.0033

Table 2: Performance comparison on ogbl-ddi dataset.

Method	Hits@20
HyperFusion	0.9972±0.0004
ELGNN[8]	0.9777±0.0037
GIDN@YITU[18]	0.9542±0.0000

Table 3: Performance comparison on ogbg-molhiv dataset.

Method	ROC-AUC
HyperFusion	0.8474±0.0003
PAS+FPs	0.8420±0.0015
HIG	0.8403±0.0021

5 Conclusion

For ensemble learning of GNN models, this paper proposes the HyperFusion model, which treats the multiple outputs of the models as features corresponding to different modalities. By utilizing a hypergraph to fuse data from these modalities, efficient model integration is achieved. Experimental results on multiple datasets demonstrate the effectiveness of the proposed method.

References

[1] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

Table 4: Performance comparison on ogbg-molpcba dataset.

Method	AP
HyperFusion	0.3204±0.0001
HIG(pre-trained on PCQM4M)	0.3167±0.0034
Graphormer(pre-trained on PCQM4M)[19]	0.3140±0.0032

[2] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3837–3845, 2016.

[3] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*. OpenReview.net, 2017.

[4] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *2nd International Conference on Learning Representations*, 2014.

[5] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1993–2001, 2016.

[6] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, pages 2224–2232, 2015.

[7] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *Proceedings of the 33nd International Conference on Machine Learning*, volume 48, pages 2014–2023, 2016.

[8] Zhen Hao Wong, Ling Yue, and Quanming Yao. Ensemble learning for graph neural networks. *CoRR*, abs/2310.14166, 2023.

[9] Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, volume 1857, pages 1–15, 2000.

[10] Ling Yue, Yongqi Zhang, Quanming Yao, Yong Li, Xian Wu, Ziheng Zhang, Zhenxi Lin, and Yefeng Zheng. Relation-aware ensemble learning for knowledge graph embedding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16620–16631, 2023.

[11] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, 2020.

[12] Qionghai Dai and Yue Gao. *Hypergraph Computation*. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer, 2023.

[13] Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. HGNN^{+}: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3181–3199, 2022.

[14] Yue Gao, Meng Wang, Dacheng Tao, Rongrong Ji, and Qionghai Dai. 3-D object retrieval and recognition with hypergraph analysis. *IEEE Transactions on Image Processing*, 21(9):4290–4303, 2012.

[15] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quant. Sci. Stud.*, 1(1):396–413, 2020.

[16] Emre Guney. Reproducible drug repurposing: When similarity does not suffice. In *Proceedings of the Pacific Symposium*, pages 132–143, 2017.

[17] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: A benchmark for molecular machine learning. *CoRR*, abs/1703.00564, 2017.

[18] Zixiao Wang, Yuluo Guo, Jin Zhao, Yu Zhang, Hui Yu, Xiaofei Liao, Hai Jin, Biao Wang, and Ting Yu. GIDN: A lightweight graph inception diffusion network for high-efficient link prediction. *CoRR*, abs/2210.01301, 2022.

[19] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, pages 28877–28888, 2021.