Isomorphic Neural Network for Graph Representation Learning and Classification

IFM Lab
Information Fusion and Mining Laboratory

Lin Meng, Jiawei Zhang

lin@ifmlab.org, jiawei@ifmlab.org

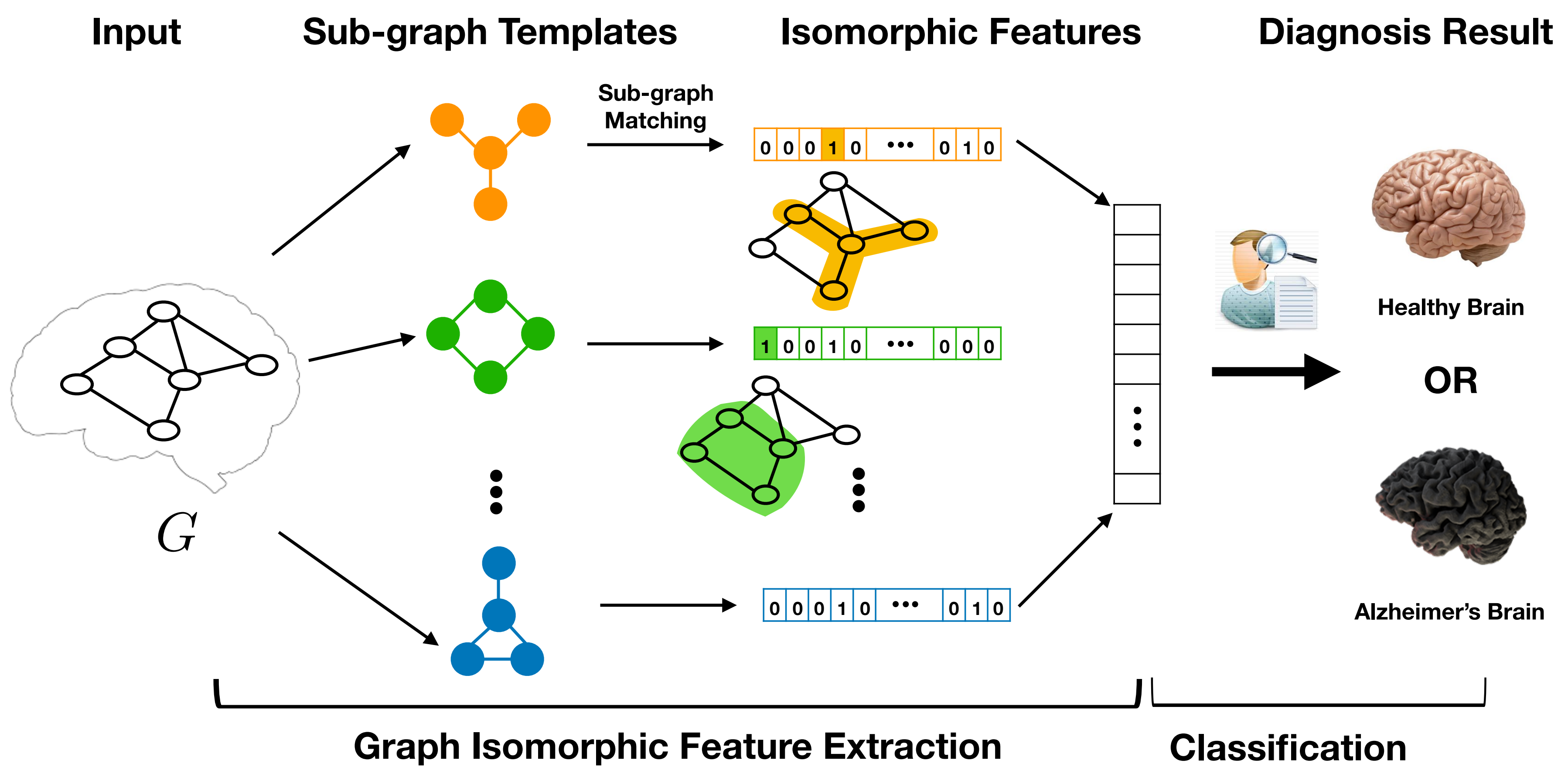
Motivation

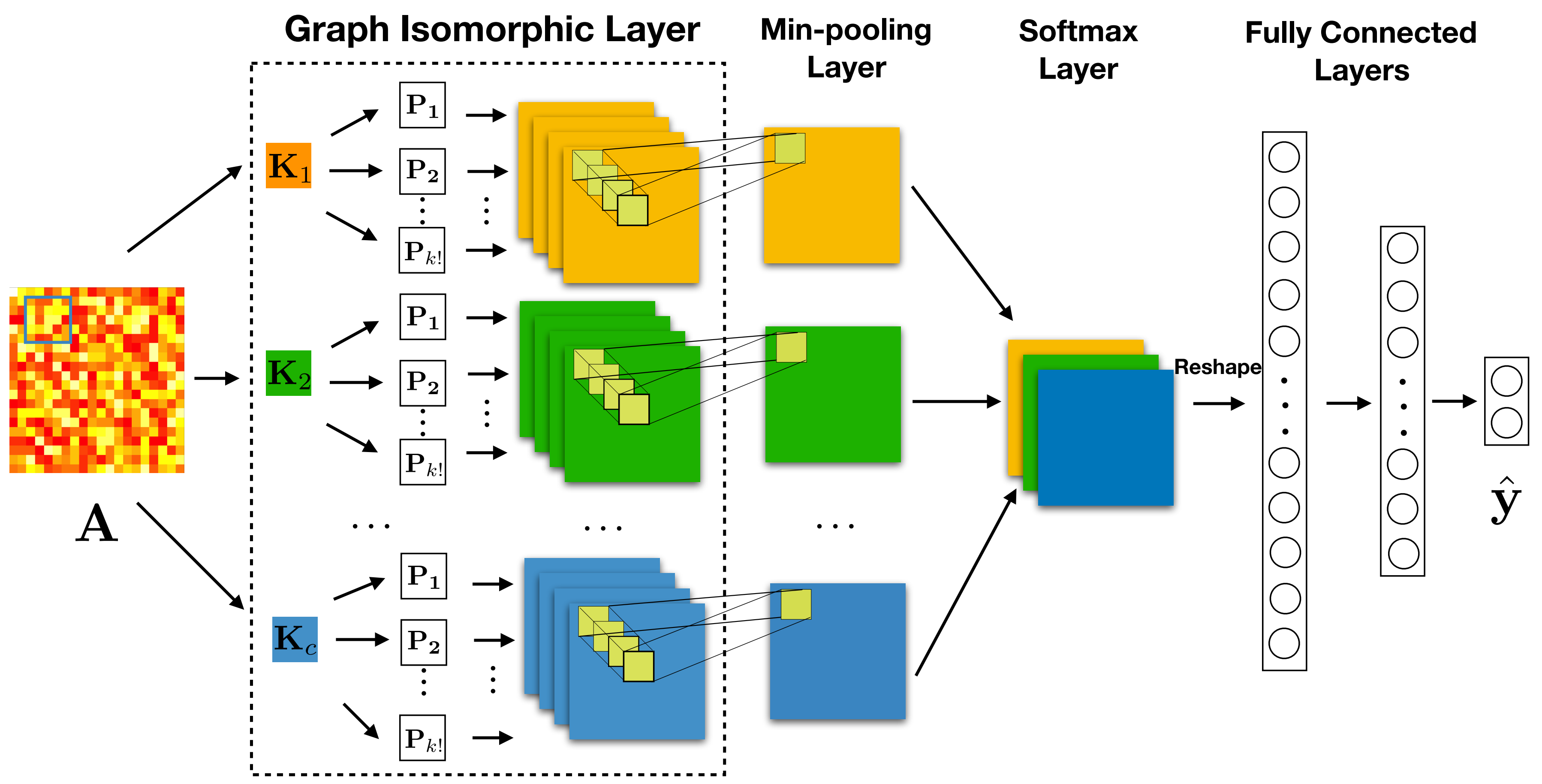
- The great representation power of graph.
- Matrix naturally pose a node-order constraint.
- Existing methods on Subgraph require high cost.
- Many graph representation model cannot maintain the structural information.

Problem Studied: propose a novel neural network to deal with the binary graph classification problem from subgraph perspective.

Proposed Model

- A novel neural network named **IsoNN** consisted of **isomorphic feature extraction component** and **classification component**.





Isomorphic feature extraction component: **automatically** extract isomorphic subgraph features with three layers in order:

1. Isomorphic Layer. It extracts all possible subgraph features under all possible node permutation for subgraphs. For each subgraph $\mathbf{M}_{(s,t)}$, we can get a isomorphic feature vector under all possible node permutations

$$\bar{\mathbf{z}}_{i,(s,t)}(j) = \left\| \mathbf{P}_j \mathbf{K}_i \mathbf{P}_j^\top - \mathbf{M}_{(s,t)} \right\|_F^2, \forall j \in \{1,2,\cdots,k!\}$$

Then, all subgraph for kernel $\mathbf{K}_i$ composes the $\bar{\mathcal{Z}}_i(1:k!,s,t) = \bar{\mathbf{z}}_{i,(s,t)}(1:k!)$.

2. Min-pooling Layer. It finds the best node permutation matrix for each subgraph for each kernel $\mathbf{K}_i$.
3. Softmax Layer

$$\mathbf{Z}_i(s,t) = \min\{\bar{\mathcal{Z}}_i(1:k!,s,t)\}$$

   1. Normalize the learned features to avoid the instability for each kernel variable.

$$\mathcal{Q}(i,:,:) = \hat{\mathbf{Z}}_i \text{ , where } \hat{\mathbf{Z}}_i = softmax(-\mathbf{Z}_i), \ \ \forall i \in \{1,\ldots,c\}$$

Classification Component

We adopt three fully-connected layer as the classifier and use the cross-entropy as the loss function

FC Layers:
$$\begin{cases} \mathbf{d}_1 = \sigma(\mathbf{W}_1\mathbf{q} + \mathbf{b}_1), \\ \mathbf{d}_2 = \sigma(\mathbf{W}_2\mathbf{d}_1 + \mathbf{b}_2), \\ \hat{\mathbf{y}} = \sigma(\mathbf{W}_3\mathbf{d}_2 + \mathbf{b}_3), \end{cases}$$

Objective function:
$$\mathcal{L} = -\sum_{g\in\mathcal{B}}\sum_{j=1}^{2} y_j^g \log \hat{y}_j^g,$$

- $\mathbf{q}$ is the flatten vector of $\mathcal{Q}$, σ denotes the ReLU function.

Experimental Results

Table 1: Classification Results of the Comparison Methods.

Dataset	Metric	Freq	Conf	Ratio	Gtest	HSIC	AE	CNN	SDBN	IsoNN
		Methods								
HIV-fMRI-77	Accuracy	54.3	58.6	54.3	50.0	58.7	46.9	59.3	66.5	**73.4**
	F1	58.2	64.2	62.0	52.5	59.5	35.5	66.3	66.7	**72.2**
HIV-DTI-77	Accuracy	64.6	52.4	59.3	59.3	49.8	62.4	54.3	65.9	**67.5**
	F1	63.9	46.1	57.9	58.5	58.3	0.0	55.7	65.6	**68.3**
BP-fMRI-97	Accuracy	56.8	50.8	54.2	55.2	54.9	53.6	54.6	64.8	**64.9**
	F1	57.6	49.1	53.7	53.9	55.8	69.5	52.8	63.7	**69.7**

- IsoNN outperforms all other baseline methods on these three datasets
- the proposed method achieves a better performance without searching for all possible subgraphs manually compared for those subgraph search based methods (Freq, Conf, Ratio, Gtest, HSIC).
- IsoNN achieves better results than AE (Autoencoder), showing that structural information contributes the classification results.
- IsoNN has simpler architecture than SDBN and performs better than it.

* For HIV-DTI, AE gets 0 in F1 due to the input contains too many zeros.

* The three datasets are benchmarks for the brain image, which have no node label or any extra information other than adjacency matrix.

Conclusion

(1) isomorphic component, where a set of permutation matrices is used to break the randomness order posed by matrix representation for a bunch of templates and one min-pooling layer and one softmax layer are used to get the best isomorphic features.

(2) The experimental results on real-world datasets show the proposed method outperforms all comparison methods, which demonstrate the superiority of our proposed method.