Disentangled Graph Convolutional Networks

Jianxin Ma (https://jianxinma.github.io), Peng Cui (http://cuip.thumedialab.com), Kun Kuang, Xin Wang, Wenwu Zhu
Department of Computer Science and Technology, Tsinghua University, Beijing, China

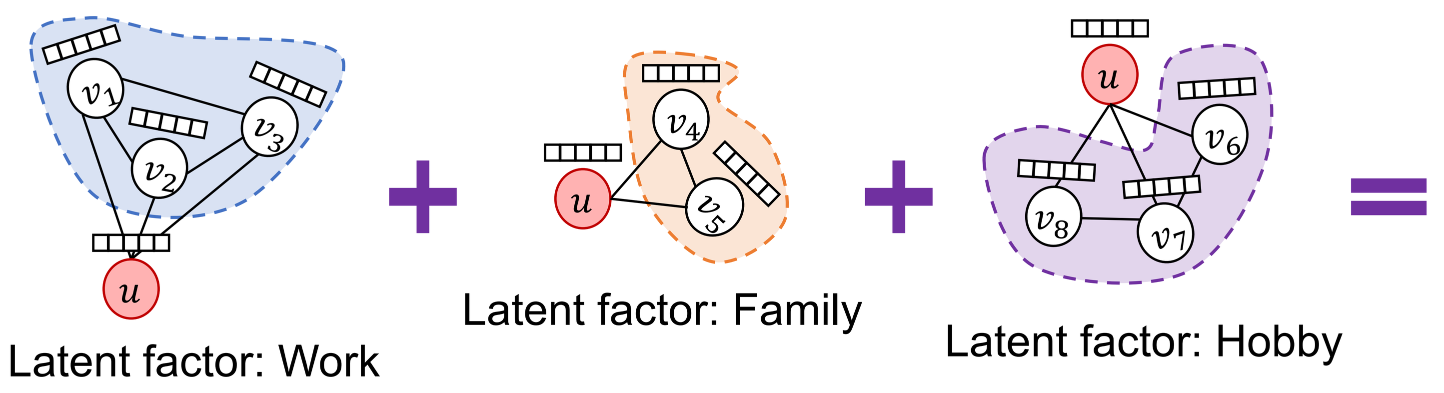
ICML | 2019

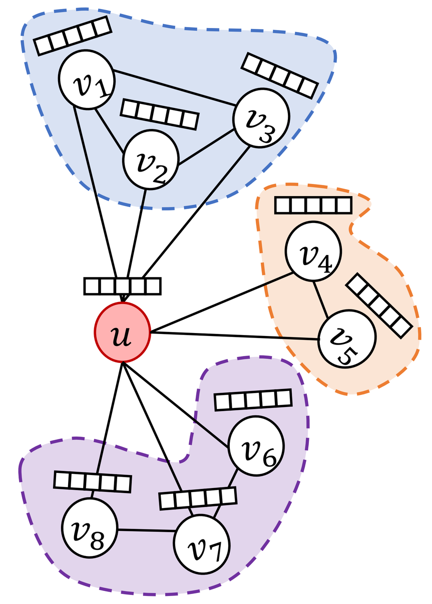
media and network lab

清華大學 Tsinghua University

1 (Motivation) Many Latent Factors behind a Graph

The neighborhood of a node is formed due to *many latent factors*.



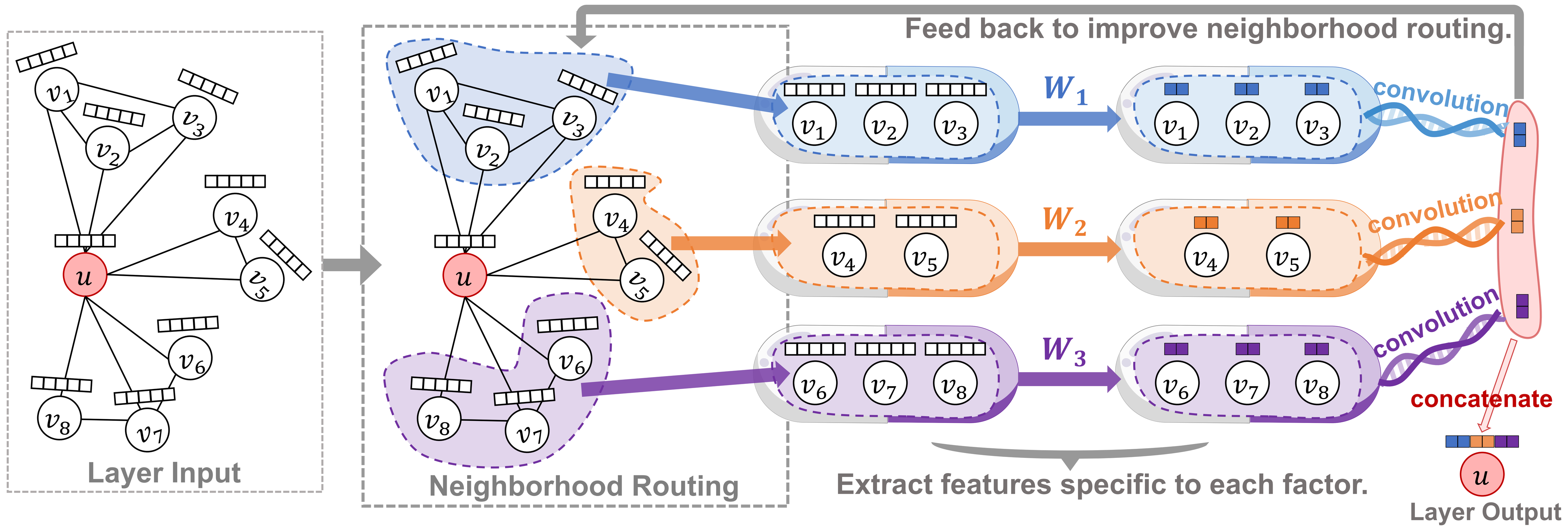


Existing graph convolutional networks convolute the neighborhood of a node *as a whole*, i.e., they do not distinguish between the latent factors. As a result:

- The node representations are *not robust*, and *hardly interpretable*.

2 (Method Overview) Disentangled Graph Convolutional Layers

Our work is related with *disentangled representation learning*, which aims to identify and separate the underlying explanatory factors behind the observed data. [Bengio et al., 2013]



We propose a disentangled graph convolutional layer. Given a neighborhood as input:

- We infer the latent factors, and segment the neighborhood accordingly.
- Each segment, related with an isolated factor, is then convoluted separately.

3 (Implementation Details) Neighborhood Routing

The proposed layer takes in a set of node features $\{\mathbf{x}_u\} \cup \{\mathbf{x}_v : (v,u) \in G\}$, and outputs K representations $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$, where $\mathbf{c}_k$ describes the aspect related with factor k.

Phase I: To extract factor-specific features.

- For node $i \in \{u\} \cup \{v : (v,u) \in G\}$ and factor $k \in \{1,2,\ldots,K\}$, we compute:
  - $\mathbf{z}_{i,k} = \dfrac{\sigma(\mathbf{W}_k^\top \mathbf{x}_i + \mathbf{b}_k)}{\left\|\sigma(\mathbf{W}_k^\top \mathbf{x}_i + \mathbf{b}_k)\right\|_2}$.
- $\mathbf{z}_{i,k}$ stands for node i's aspect k. Each $\mathbf{z}_{i,k}$ may be incomplete or inaccurate. We are hence required to look at the whole picture by convoluting the neighborhood.

Phase II (Neighborhood Routing): To identify the factor that causes the link between node u and a neighbor v.

- Initialize $\mathbf{c}_k \leftarrow \mathbf{z}_{u,k}$ for each factor k.
- Iteratively update $\mathbf{c}_k$ for $T \approx 5$ times:
  - $p_{v,k} \leftarrow \dfrac{\exp(\mathbf{z}_{v,k}^\top \mathbf{c}_k / \tau)}{\sum_{k'} \exp(\mathbf{z}_{v,k'}^\top \mathbf{c}_{k'} / \tau)}$
  - $\mathbf{c}_k \leftarrow \dfrac{\mathbf{z}_{u,k} + \sum_{v:(v,u)\in G} p_{v,k} \mathbf{z}_{v,k}}{\left\|\mathbf{z}_{u,k} + \sum_{v:(v,u)\in G} p_{v,k} \mathbf{z}_{v,k}\right\|_2}$
- $\mathbf{c}_k$ $(k = 1,2,\ldots,K)$ is a representation that describes the neighborhood's aspect k.

Remark:

- It is differentiable, and therefore can be trained in an end-to-end manner.
- It supports real-time processing of new nodes, i.e., inductive learning.

4 Intuitions & Theoretic Analysis

Intuitions: Neighborhood routing is based on two hypotheses.

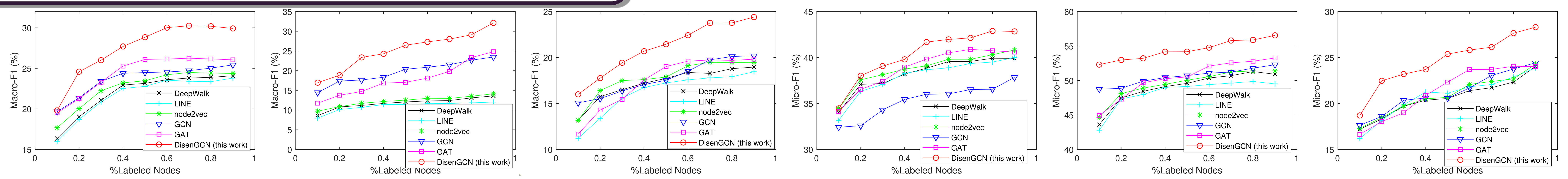
- $p\left(\begin{array}{c}\text{Factor } k \text{ is the one that causes}\\ \text{the links between node } u \text{ and a segment}\end{array}\right) \propto$ The segment contains a large number of nodes that are similar w.r.t. aspect k.
- $p\left(\begin{array}{c}\text{Factor } k \text{ is the one that causes}\\ \text{the link between node } u \text{ and a neighbor } v\end{array}\right) \propto$ Node u and the neighbor v are similar w.r.t. aspect k.

Theory: Neighborhood routing is equivalent to an expectation-maximization (EM) algorithm that performs probabilistic inference under a von Mises-Fisher subspace clustering model.

- It searches for one large cluster in each of the K subspaces, assuming that each neighbor belongs to one cluster.

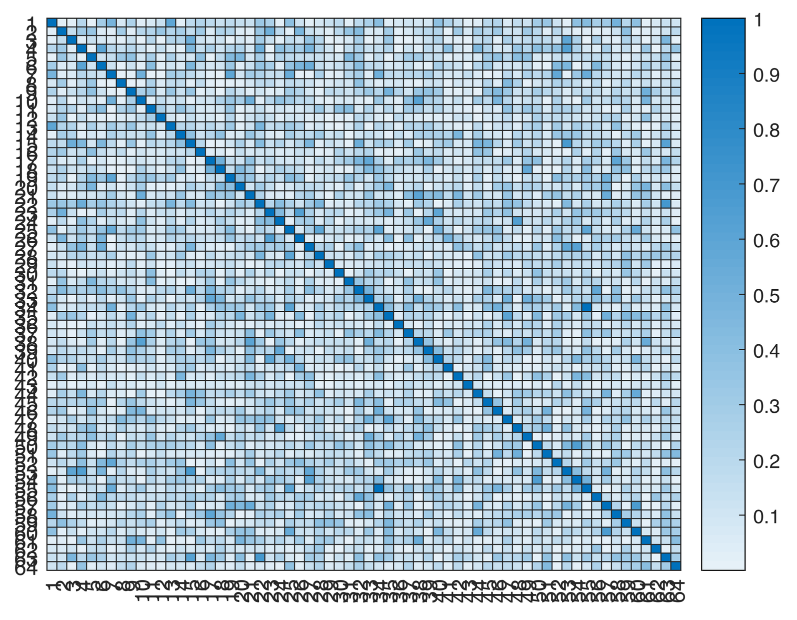
For the center node u,
$\mathbf{z}_{u,k} \sim vMF(\mathbf{c}_k, 1/\tau)$,
$k = 1,2,\ldots,K$.
For $v \in \{v : (v,u) \in G\}$,
$r_v \sim Categorical(\mathbf{1}/K)$,
$\mathbf{z}_{v,r_v} \mid r_v \sim vMF(\mathbf{c}_{r_v}, 1/\tau)$,
and
$\mathbf{z}_{v,k'} \mid r_v \sim vMF(\boldsymbol{\mu}, 0)$,
$k' = 1,2,\ldots,K$, and $k' \neq r_v$.

5 (Experiment) Multilabel Node Classification

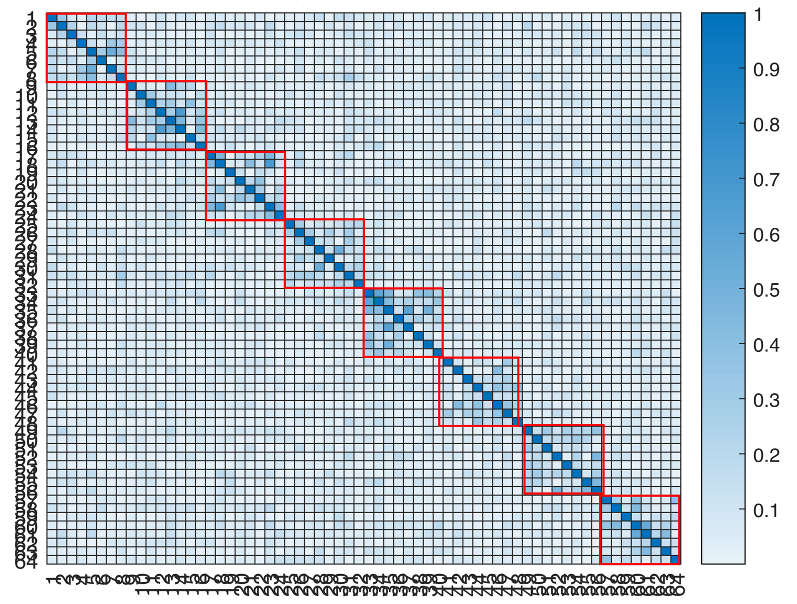


(a) BlogCatalog, Macro-F1. (b) POS, Macro-F1. (c) PPI, Macro-F1. (d) BlogCatalog, Micro-F1. (e) POS, Micro-F1. (f) PPI, Micro-F1.

6 (Experiment) Disentanglement



(a) GCN.



(b) DisenGCN (this work).

The absolute correlations between 64 output neurons, on a graph with eight factors. The eight channels of our model are capturing mutually exclusive information, as Fig. b exhibits eight clear diagonal blocks.