Over-smoothing 防止のための層ごとに重み付けする Summarize-GNN

情報科学専攻 猪口研究室　47020726 矢嶋 悠太

1　はじめに

属性をもつ頂点と辺からなるグラフ構造は、SNS など実世界の様々なデータを表現できる。グラフの各頂点を高精度にクラス分類することは、SNS の各ユーザを適切なコミュニティに分類することに対応する。近年、頂点の高精度なクラス分類のために、グラフを入力として、互いに属性が似ていたり、辺で結ばれている頂点同士が空間上の近い位置にマッピングされるように、各頂点をベクトル表現する、表現学習が注目されている。頂点のベクトル表現が得られると、既存のクラス分類モデルにそのベクトル表現を入力するだけで、頂点のクラス分類が可能になる。頂点のクラス分類精度を向上させる、質の高い頂点の表現を学習するために、近年は深層学習を用いた Graph Neural Network (GNN)[1][2] が注目されている。

GNN は層数を適切に設定することで高い頂点の分類精度が得られる一方、より多層にすると分類精度が低下する。GNN を多層化したときに、学習される頂点の表現の質が低下し、クラス分類精度が落ちる現象を Over-smoothing という。本研究は Over-smoothing を防止することを目的とする。

グラフの定義及び頂点のクラス分類

グラフを $G = (V, E, X)$ とする。ここで V は n 個の頂点集合 $\{1, \ldots, n\}$、E は頂点 v，u 間の辺 e_{vu} の集合である。各頂点 v がもつ属性 $\boldsymbol{x}_v$ はベクトルとし、全頂点の属性の集合を $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ とする。このグラフ G を入力とし、表現学習モデルにより、全頂点のベクトル表現 $H = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n\}$ を出力する。H を入力とし、クラス分類モデルにより、全頂点のクラスラベルの予測を出力する。頂点のクラスラベルの予測精度 (分類精度) を向上させる表現 H 程、質が高いと言える。

2　既存インターフェース：GNN

GNN は、G と層数 L を入力とし、頂点 v の表現 $\boldsymbol{h}_v$ を求める。

$$\boldsymbol{h}_v^l = \mathrm{Conv}(H^{l-1}, N(v)) = \sigma\left(W^l \sum_{u \in N(v)} w_{vu}^l \boldsymbol{h}_u^{l-1}\right) \quad (1)$$

$$\boldsymbol{h}_v = \boldsymbol{h}_v^L \quad (2)$$

頂点 v 自身を含む、v と隣接する頂点の集合 : $\{v\} \cup \{u \mid e_{vu} \in E\}$ を「v の 1 近傍内」と呼び、$N(v)$ で表記する。また、$\boldsymbol{h}_v^0$ は属性 $\boldsymbol{x}_v$、$\sigma(\cdot)$ は活性化関数である。$l-1$ から l 層に進む度、式 (1) の畳み込み Conv により、頂点 v の表現 $\boldsymbol{h}_v^{l-1}$ を $\boldsymbol{h}_v^l$ に更新する。各 l 層の Conv では以下を順に実行する。

1. v の 1 近傍内の各頂点 $u \in N(v)$ の表現を足し合わせる集約
2. 行列 W^l と活性化関数 $\sigma(\cdot)$ による線形及び非線形な変換

よって、Conv を 2 回繰り返して得られる表現 $\boldsymbol{h}_v^2$ には、v の 1 近傍内の $u \in N(v)$ の、更に 1 近傍内の $u' \in N(u)$ の属性 $\boldsymbol{x}_{u'}$ が集約される。つまり、$\boldsymbol{h}_v^2$ には 2 近傍内の頂点の属性が集約される。式 (2) より、Conv を層数 L 回分繰り返して得られる、L 近傍内の頂点の属性が集約された $\boldsymbol{h}_v^L$ を $\boldsymbol{h}_v$ として出力する。

Conv の集約の仕方 (重み w_{vu}^l の具体的な学習手法) により、GCN[1]，GAT[2] など様々な GNN モデルとして具体化される。

3　GNN の課題：Over-smoothing の原因の推察

適切な層数 L_v^* の GNN は質の高い表現を得ることができ、近年最高峰の分類精度を達成している [1][2]。一方、GNN の層数 L を L_v^* より大きくする程、つまり GNN を多層にする程、表現 $\boldsymbol{h}_v (= \boldsymbol{h}_v^L)$ の質が低下し、その後のクラス分類精度も低下する現象を Over-smoothing と呼ぶ。

図 1 の赤と青の破線で示すように、将来的に分類すべき頂点のクラスラベルが全て分かっている、という前提で Over-smoothing の原因を推察する。グラフの中心にある v は 1 近傍内だけ集約すれば、自身と同ラベルの頂点の属性を全て集約できる。一方、グラフの端にある u はより遠くの 4 近傍内を集約してようやく、自身と同ラベルの頂点の属性を全て集約できる。つまり、頂点 v ごとに適切な集約距離 L_v^* は異なる。

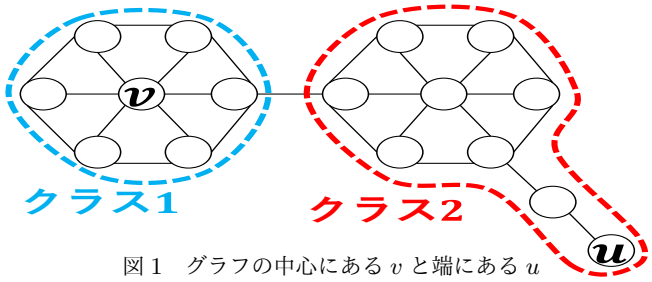

図 1　グラフの中心にある v と端にある u

式 (2) より、L 層 GNN[1][2] は、どの頂点 v についても、L 近傍内の属性が集約された $\boldsymbol{h}_v^L$ を $\boldsymbol{h}_v$ として出力する。つまり、どの頂点 v についても近傍の集約距離を一括して L としている。よって、頂点 v ごとに適切な集約距離 L_v^* を学習することはできず Over-smoothing を引き起こす、と本研究は推察する。

4　提案インターフェース：Summarize-GNN

式 (2) の代わりに、以下の L_v^* を学習する Summarize 関数をもつ、Summarize-GNN というインターフェースを提案する。

$$\boldsymbol{h}_v = \mathrm{Summarize}(\boldsymbol{h}_v^1, \ldots, \boldsymbol{h}_v^L) = \sum_{l=1}^{L} \alpha_v^l \boldsymbol{h}_v^l \quad (3)$$

但し、α_v^l は学習可能パラメータであり、$\sum_{l=1}^{L} \alpha_v^l = 1$ を満たす。

式 (1) の従来の GNN の Conv を L 回繰り返し、表現の系列 $\{\boldsymbol{h}_v^1, \ldots, \boldsymbol{h}_v^L\}$ を得る。次に、式 (3) の Summarize 関数に $\{\boldsymbol{h}_v^1, \ldots, \boldsymbol{h}_v^L\}$ を入力し、各表現 $\boldsymbol{h}_v^l$ をパラメータ α_v^l で重み付けして、最終的な表現 $\boldsymbol{h}_v$ として出力する。尚、α_v^l の具体的な学習手法は 5 節で提案する。

もしパラメータ α_v^l が適切に学習できれば、図 1 のグラフの中心にある頂点 v の場合、$(\alpha_v^1, \alpha_v^2, \alpha_v^3, \alpha_v^4) = (1, 0, 0, 0)$ で、1 近傍内の属性が集約されている表現 $\boldsymbol{h}_v^1$ に大きな重み付けができる。一方、図 1 のグラフの端にある頂点 u の場合、$(\alpha_u^1, \alpha_u^2, \alpha_u^3, \alpha_u^4) = (0, 0, 0, 1)$ で、より遠い 4 近傍内の属性が集約されている表現 $\boldsymbol{h}_u^4$ に大きな重み付けができる。つまり、α_v^l は、各 l が頂点 v にとって適切な集約距離 L_v^* である確率、と捉えられる。また、この α_v^l を学習することは、L_v^* を学習することに対応し、様々な GNN モデルが抱える Over-smoothing の防止が期待できる。

5　提案手法：クラスラベルを用いない α_v^l の学習

どのような条件を満たすとき α_v^l が高くなるか、クラスラベルを用いずに考える。3 節冒頭で述べたように、多層 (L 層) の GNN で得られる表現 $\boldsymbol{h}_v^L$ は質が低い。そこで、$\boldsymbol{h}_v^L$ を反面教師と捉えて、その反面教師と $\boldsymbol{h}_v^l$ 間の類似度が低い、という条件を満たすとき $\boldsymbol{h}_v^l$ は質が高いと考える。つまり、$\boldsymbol{h}_v^L$ と $\boldsymbol{h}_v^l$ 間の類似度が低い、という条件を満たすとき α_v^l が高くなると考える。

この条件が正しいことを説明するため、図 2 左上で示すように、各頂点 v が単一の属性値 $x_v \in \mathbb{R}$ をもつグラフの例を用いる。属性値 x_v の大きさは、図 2 右上の横軸のように色の濃さで表す。また、各頂点の属性値 x_v は v のクラスラベルに応じたガウス分布に従うと仮定する。例えば今回の例の場合、クラスラベルが 1 の頂点の属性値 x_1, x_2, x_3 は $\mathbb{G}(\mu_1, \sigma_1^2)$ に、クラスラベルが 2 の頂点の属性値 x_4, x_5, x_6 は $\mathbb{G}(\mu_2, \sigma_2^2)$ に従う。

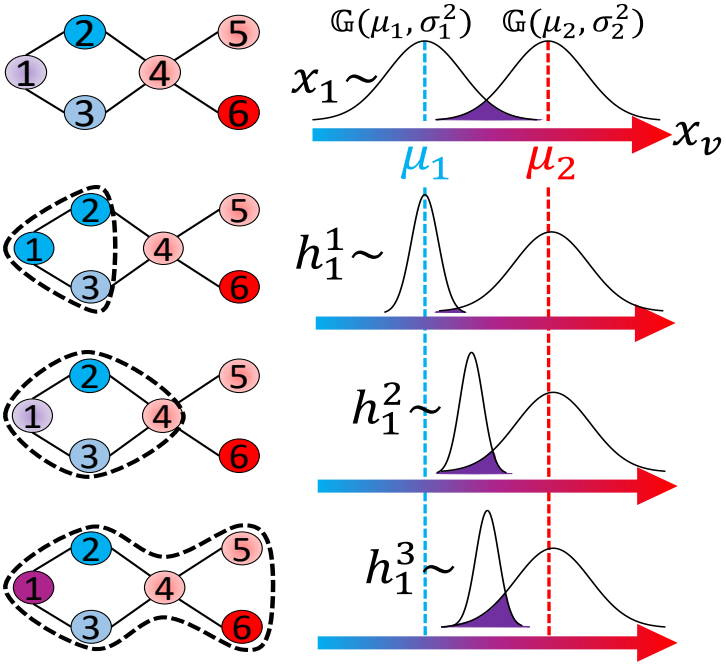

図 2　グラフ G と、集約距離 l により変化する h_1^l の従う分布

頂点のクラスラベルが全て分かっていれば、頂点 1 は 1 近傍内を集約すれば、自身と同じラベルの頂点の属性値 x_2 と x_3 を全て集約できる。よって L_1^* は 1 であり、最適な $(\alpha_1^{1*}, \alpha_1^{2*}, \alpha_1^{3*})$ は $(1, 0, 0)$ である。対して、本研究の考える、クラスラベルが不要な条件に基づき求まる $(\alpha_1^1, \alpha_1^2, \alpha_1^3)$ の期待値について考える。

層数 1 の GNN により、頂点 1 には 1 近傍内の属性が集約、足し合わされる (図 2 左側二段目)。つまり、x_1 と同じガウス分布 $\mathbb{G}(\mu_1, \sigma_1^2)$ に従う属性が足し合わされる。同じ分布に従う属性を足し合わせることで得られる表現値 h_1^1 が従う分布は、元々の属性値 x_1 が従う分布 $\mathbb{G}(\mu_1, \sigma_1^2)$ と比較して、平均は維持したまま分散が小さくなる (図 2 右側一段目と二段目を比較)。これは分布の再生成という統計の性質から導ける。

しかし、1 を超えて、2，3 層と、より GNN を多層にすると、頂点 1 には 2，3 近傍内の属性が集約、足し合わされる (図 2 左側三、四段目)。つまり、x_1 と異なるガウス分布 $\mathbb{G}(\mu_2, \sigma_2^2)$ に従う属性も足し合わされる。すると、同じく分布の再生成という性質から、h_1^2，h_1^3 が従う分布の平均は μ_1 と μ_2 の間にずれる。

以上、GNN の層数 l を 1,2,3 と伸ばしながら、各表現値 h_1^l の従う分布の変化を観察した。l が 1 のとき、図 2 右側四段目の分布から生起する質の低い反面教師 h_1^3 と、二段目の分布から生起する h_1^1 間の類似度の期待値は低い。よって確率 α_1^1 の期待値は高い。一方、l が 2 のとき、h_1^3 と、三段目の分布から生起する h_1^2 間の類似度の期待値は高い。よって確率 α_1^2 の期待値は低い。

以上の例を用いて、本研究が提案する類似度に基づいた条件により、クラスラベルを用いることなく、最適 (α_v^{l*}) に近い α_v^l を求められることを説明した。以下で具体的に求める。

$$\overline{\alpha}_v^l = \mathrm{Attention}(\boldsymbol{h}_v^l, \boldsymbol{h}_v^L) \tag{4}$$

$$\alpha_v^l = \mathrm{Softmax}(\overline{\alpha}_v^l, L) = \frac{\exp(\overline{\alpha}_v^l)}{\sum_{l'=1}^{L} \exp(\overline{\alpha}_v^{l'})} \tag{5}$$

式 (4) の Attention[3] により $\boldsymbol{h}_v^l$ と $\boldsymbol{h}_v^L$ 間の類似度を求め、式 (5) の Softmax で $\sum_{l=1}^{L} \alpha_v^l = 1$ を満たすように正規化する。

6　評価実験

Over-smoothing の防止性能について 3 つの既存モデルと比較する。1 つ目は 2 節で紹介した GNN[1][2] とする。2 つ目は、式 (1) の各 Conv を Skip することが可能な Skip-GNN[4] とする。3 つ目は、本研究と同じ目的の下、メモリ量を要するモデルを用いて α_v^l を学習する Complex Summarize-GNN (CS-GNN)[5] とする。類似度により α_v^l を求める提案モデルは CS-GNN より省メモリなため、Simple Summarize-GNN (SS-GNN) とする。

データセット (グラフ G) は、Reddit, Arxiv, PubMed, PPI, $\mathrm{PPI}_{\mathrm{Ind.}}$ の 5 つを用いる (頂点数 n は数万から数十万)。G と一部の頂点のクラスラベル集合を用いて、各モデルを学習する。次に、残りの頂点のクラスラベル集合の内、学習済みモデルにより正しくラベルを予測できた割合を頂点の分類精度とする。

層数 L は 6 ～ 8 に設定し、多層にした各モデルによる分類精度を表 1 に示す。全データセットにおいて、既存より同等以上の分類精度を多層の SS-GNN は達成した。つまり既存より同等以上の Over-smoothing の防止性能を達成した。また、Reddit と Arxiv では、他モデルよりメモリ量を要する CS-GNN が OOM に陥り実験できなかったが、SS-GNN は実験できた。

表 1　多層な各モデルによる分類精度 (*OOM は Out of Memory)

	Reddit	Arxiv	PubMed	PPI	$\mathrm{PPI}_{\mathrm{Ind.}}$
GNN	79.5	70.9	76.8	81.0	47.9
Skip-GNN	95.1	71.2	86.8	81.5	97.7
CS-GNN	OOM*	OOM*	88.8	80.8	**99.1**
SS-GNN	**96.5**	**72.7**	**89.4**	**82.1**	**99.0**

7　まとめ

GNN の課題である Over-smoothing を防止するべく、頂点 v ごとの適切な集約距離 L_v^* を学習する SS-GNN を提案した。既存モデル GNN，Skip-GNN，CS-GNN と比較して、SS-GNN は同等以上の Over-smoothing 防止性能を持ちつつ、大規模データセットにも適用可能なモデルであることが実験的に示された。

参考文献

[1] T. N. Kipf, M. Welling.: Semi-supervised classification with graph convolutional networks, *arXiv Preprint*, arXiv:1609.02907 (2016).

[2] P. Veličković, et al.: Graph attention networks, *arXiv Preprint*, arXiv:1710.10903 (2017).

[3] A. Vaswani, et al.: Attention is all you need, *NIPS*, pp. 5998–6008 (2017).

[4] G. Li, et al.: Deepergcn: All you need to train deeper gcns, *arXiv Preprint*, arXiv:2006.07739 (2020).

[5] K. Xu, et al.: Representation learning on graphs with jumping knowledge networks, *ICML*, pp. 5453–5462. PMLR (2018).