

クラス間の分散の最大化に基づく グラフノード埋め込み学習の改善

井上 魁人^{1,a)} 方 鐘熙² 黄 健明² 笠井 裕之²

概要: 近年、グラフのノード分類タスクにおいて、Graph Neural Networks (GNN) は著しい精度向上を達成している。しかしながら、グラフノード分類タスクで一般的に損失関数として用いられる交差エントロピーは予測ラベルを正解ラベルに近づけることのみを考慮した関数であり、クラス識別に有効なノード埋め込みが困難である。そこで本稿では、クラス間分散およびクラス内分散に着目し、クラス識別に有効なノード埋め込み手法を提案する。評価実験から、ノード分類精度について提案手法は従来手法と同等かそれ以上の性能を有することを示す。

1. はじめに

現実世界には、論文情報や化学物質 [1], [2], ソーシャルネットワークなど様々なグラフ構造を持つデータが溢れている。本稿ではエッジの向きを考慮しない無向グラフを対象とする。ここでグラフデータを $G = (V, E)$ と表し、構成するノードの数を N とする。また、ノードとエッジを $v_i \in V, (v_i, v_j) \in E$ と表し、ノード特徴量の次元数を D とする。近年、このようなグラフ構造を有するデータに対して、ディープラーニングの考え方をういた Graph Neural Networks (GNN) が注目を浴びている。

GNN は、隣接するノードの特徴量ベクトルと、自身の特徴量ベクトルを組み合わせることで新たな特徴量を生成し、様々なタスクに取り組む手法である。自身のノードの特徴量と隣接するノードの特徴量をどのように集めるかを定める関数を集約関数、そしてその集めた情報をどのように更新していくかを定める関数を更新関数と呼ぶ。この二つの関数をどのような関数で定義するかにより様々な種類の GNN が提案されている。代表的な例として Graph Convolutional Networks (GCN)[3], Graph Attention Neural Network (GAT)[4], [5] などが挙げられる。GNN はグラフ分野のタスクである、グラフ表現タスク、グラフノード分

類タスク、リンク予測タスクにおいて従来手法より著しい精度向上を実現している。

本稿では、特にノードクラス分類タスクに着目する。グラフノード分類タスクにおいて一般的に損失関数として用いられる交差エントロピーは、予測ラベルを正解ラベルに近づけることのみを考慮した関数であり、クラス識別に有効なノード埋め込みが困難であると考えられる。

そこで本稿では、クラス識別に有効なノード埋め込み手法を提案し、ノード分類の精度向上を目指す。具体的には、線形判別分析 (LDA) で基本とする、クラス間分散を最大化しながらクラス内分散を最小化する考え方に着目する。この考え方を、既存の交差エントロピーの損失関数に付加する手法を提案する。最後に、評価実験からノード分類のタスクにおいて、既存手法と比較して提案手法が同等かそれ以上の性能を有することを示す。

2. 関連研究

本稿では、ノード分類の損失関数として使用されている交差エントロピーにクラス間分散とクラス内分散の要素を加える手法を提案し、既存の GNN モデルに対して適用する。そこで、本節では適用対象の GNN モデルについて説明をする。さらに、本稿で対象とする同じ課題の解決を目指すカーネル手法 [6] についても説明をする。

2.1 Graph Convolutional Networks [3]

画像処理の分野において、グリッド構造を有するデータ [7], [8] に対して、学習可能なパラメータをもつローパスフィルタを用いて周辺情報を畳み込むことで効率的に学習をする手法として Convolutional Neural Network

¹ 早稲田大学 基幹理工学部 情報通信学科
Department of Communications and Computer Engineering,
School of Fundamental Science and Engineering, Waseda
University

² 早稲田大学大学院 基幹理工学研究科 情報理工・通信専攻
Department of Computer Science and Communications Engineering,
Graduate School of Fundamental Science and Engineering,
Waseda University

a) inouekt@ruri.waseda.jp

(CNN)[9]が提案されている。しかしながら、現実世界のデータは必ずしもグリッド構造を持つとは限らない。例えば、ソーシャルネットワークや3Dメッシュ、化学物質の構造などはグラフ構造を有する。

このようなグラフ構造データに対して、CNNの考え方とGNNと組み合わせた手法としてGraph Convolutional Network (GCN)[3], [10], [11]が提案されている。具体的には、グラフデータに含まれるノード特徴量ベクトルと、ノードとノードの隣接関係を示した隣接行列から、メッセージパッシングの考え方にに基づき、あるノード特徴量をその周辺のノード特徴量と組み合わせる。その情報に基づいて当該ノードの特徴量を再帰的に更新していく。そして、最終的なノード特徴量を自身のノード特徴量とする手法である。ここで隣接行列を $\mathbf{A} \in \mathbb{R}^{N \times N}$ 、対角次数行列を $\hat{\mathbf{D}} \in \mathbb{R}^{N \times N}$ 、単位行列を $\mathbf{I} \in \mathbb{R}^{N \times N}$ 、パラメータを $\mathbf{W} \in \mathbb{R}^{D \times D}$ 、活性化関数を ρ 、 $\bar{\mathbf{A}} = \hat{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\hat{\mathbf{D}}^{-\frac{1}{2}}$ 、ノード属性情報行列を $\mathbf{H} \in \mathbb{R}^{N \times D}$ 、 l 回畳み込みをした際のノード属性情報行列を $\mathbf{H}^{(l)}$ と定義する。このとき、 \mathbf{H} は以下のように更新される。

$$\mathbf{H}^{(l+1)} = \rho(\bar{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$$

2.2 グラフノード分類にカーネルの要素を加えた研究

グラフ分類のタスク [12]において、カーネルを取り入れる検討が多く行われており、これらは一般的にグラフカーネルと呼ばれている [13], [14]。グラフカーネルは、隣接行列やノード特徴量をもとに類似性を判定する手法である。しかし、ノード分類やリンク予測タスクではこのグラフカーネルを使用した研究は少ない。そのような中、グラフノード分類を対象としたグラフカーネルが提案されている [6]。

具体的には、損失関数として一般的に用いられる交差エントロピーに対して、同じラベル同士のコサイン類似度を取った際に1に、異なるラベル同士のコサイン類似度を取ったときに-1になるような損失関数を加えることで、クラス識別に有効なノード埋め込みを行なっている。

3. 提案手法

本節では、分類のみを目的とした交差エントロピーのみでは、クラス識別に有効なノード埋め込みが困難であるという課題に対して、クラス間分散とクラス内分散に着目することでクラス識別に有効なノード埋め込みを可能にする提案手法について詳しく述べる。

3.1 課題

ノード分類のタスクでは損失関数として、交差エントロピーが一般的に用いられている。ここで、ラベルをone-hotベクトルに変換する関数を $q: \mathbb{R} \rightarrow \mathbb{R}^C$ 、クラスラベルの

数を C 、ノード特徴量を定める関数を $g_{(\theta)}: \mathcal{V} \rightarrow \mathbb{R}^D$ と定義すると、交差エントロピーの損失関数は以下のように示される。

$$\mathcal{L} = - \sum_{v_i \in \mathcal{V}} q(y_i) \log(g_{\theta}(v_i)) \quad (1)$$

この交差エントロピーは分類のみを目的とした関数なため、クラス識別に有効なノード埋め込みが困難である。

3.2 線形判別分析 (LDA)

本稿の提案手法は、LDAの最適化式であるクラス間分散を最大化してクラス内分散を最小化する項を加えたものであることから、本節ではLDAについて詳しく説明する。LDAは、データ $\mathbf{x} \in \mathbb{R}^d$ の集合 \mathcal{X} にて、データ \mathbf{x} の特徴量次元数を d 、データ \mathbf{x} の次元削減後の特徴量次元数を d' 、クラス i に属するデータの集合を \mathcal{X}_i 、その平均を $\boldsymbol{\mu}_i \in \mathbb{R}^d$ 、全体の平均を $\boldsymbol{\mu} \in \mathbb{R}^d$ 、クラス i に属するデータの数を n_i と定義する。このとき、クラス内共分散行列 $\mathbf{S}_W \in \mathbb{R}^{d \times d}$ とクラス間共分散行列 $\mathbf{S}_B \in \mathbb{R}^{d \times d}$ は以下のように定義する。

$$\mathbf{S}_W = \sum_{i=1}^C \sum_{\mathbf{x} \in \mathcal{X}_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^T,$$

$$\mathbf{S}_B = \sum_{i=1}^C n_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T.$$

この \mathbf{S}_W と \mathbf{S}_B を用いて、線形変換行列 $\mathbf{W} \in \mathbb{R}^{d \times d'}$ によって次元削減された後のクラス間分散をより大きく、クラス内分散をより小さくする次元削減手法である。最適化目的関数は以下のように定義される。

$$J(\mathbf{W}) = \frac{\text{tr}(\mathbf{W}^T \mathbf{S}_W \mathbf{W})}{\text{tr}(\mathbf{W}^T \mathbf{S}_B \mathbf{W})}.$$

以上の $J(\mathbf{W})$ を最大化することで \mathbf{W} を求めることができる。

3.3 提案手法

前述したように、グラフノード分類タスクにおいて、損失関数として用いられる交差エントロピーは、予測ラベルを正解ラベルに近づけることのみを考慮した関数であり、クラス識別に有効なノード埋め込みが困難である。これに対して、クラス識別に有効なノード埋め込み手法を提案し、ノード分類の精度向上を目指す。具体的には、線形判別分析 (LDA) で基本とする、クラス間分散を最大化しながらクラス内分散を最小化する考え方に着目する。この考え方を、既存の交差エントロピーの損失関数に付加する手法を提案する。

ノードの特徴量行列を $\mathbf{X} = [g_{\theta}(v_1), \dots, g_{\theta}(v_N)]^T \in \mathbb{R}^{N \times D}$ 、ノードのラベル集合 $\mathbf{y} \in \mathbb{R}^N$ 、それぞれのクラスの平均を $\mathbf{M} \in \mathbb{R}^{C \times D}$ 、 D 次元の全体の平均ベクトルを列方向に C 個並べたものを $\hat{\mathbf{M}} \in \mathbb{R}^{C \times D}$ 、クラスの個数を対

表 1 データセットの詳細

データセット	ノード数	エッジ数	クラス数	特徴量数	ラベル率
Cora	2,708	5,429	7	1,433	0.052
Citeseer	3,327	4,732	6	3,703	0.036
Pubmed	19,717	44,338	3	500	0.003

角成分に持つ行列を $\mathbf{N} \in \mathbb{R}^{C \times C}$, クラスラベルベクトルを one-hot ラベル行列に変換する関数 $Q: \mathbb{R}^N \rightarrow \mathbb{R}^{N \times C}$ と定義すると, 各クラスの平均 $\mathbf{D} \in \mathbb{R}^{N \times D}$ を \mathbf{X} から計算する.

$$\mathbf{D} = \mathbf{X} - Q(\mathbf{y}) \cdot \mathbf{M}.$$

上式より, クラス内分散を以下に定義する.

$$s_W = \text{tr}(\mathbf{D}^T \mathbf{D}). \quad (2)$$

また, 各クラスの平均と全体の平均との差分行列を $\hat{\mathbf{D}} \in \mathbb{R}^{C \times D}$ は以下に計算する.

$$\hat{\mathbf{D}} = \mathbf{M} - \hat{\mathbf{M}}.$$

上式より, クラス間分散は以下のように定義する.

$$s_B = \text{tr}(\hat{\mathbf{D}}^T \hat{\mathbf{N}} \hat{\mathbf{D}}). \quad (3)$$

提案手法の損失関数は既存の交差エントロピー損失関数にクラス間分散をより大きく, クラス内分散をより小さくする要素を加える手法であることから, 式 (1), (2), (3) より以下のように定義する.

$$\mathcal{L} = - \left(\sum_{v_i \in \mathcal{V}} q(y_i) \log(g_\theta(v_i)) \right) + \alpha \frac{s_W}{s_B}. \quad (4)$$

ここで $\alpha > 0$ は交差エントロピーと提案した損失関数の項の比率を表す. また, クラス間分散のみに着目したときの損失関数を以下に定義する.

$$\mathcal{L} = - \left(\sum_{v_i \in \mathcal{V}} q(y_i) \log(g_\theta(v_i)) \right) + \alpha \frac{1}{s_B}. \quad (5)$$

4. 数値実験

提案手法の有効性を示すために, 損失関数を交差エントロピーのみにした手法, 損失関数として交差エントロピーにカーネルの要素を加えた手法を用いてノード分類の精度比較実験を行った.

4.1 データセット

実験で用いる Cora, Citeseer, Pubmed は, 引用ネットワークのデータセット [15] であり, ノードは文章, エッジは引用リンクを示す. ラベル率は, 学習に用いられたラベル付きのノードの数をデータセットのすべてのノード数で割った数である. これらのデータセットは, 特徴量ベクトル

ルとしてそれぞれの文章のスパースな bag-of-words, 文章間のつながりを表す引用リンクが含まれている. この引用リンクを無向エッジとして扱い, この情報をもとに隣接行列 \mathbf{A} を作成する. また, それぞれの文章はクラスラベルを持っている. 学習の際には訓練データとしてクラスごとに 20 個のノードを使う. 特徴量ベクトルはすべて使う.

それぞれのデータセットに含まれるノード数, エッジ数, クラス数を, 表 1 に示す. 訓練データ, 検証データ, テストデータの数を表 2 に示す.

表 2 訓練データ, 検証データ, テストデータ

データセット	訓練	検証	テスト
Cora	140	500	1000
Citeseer	120	500	1000
pubmed	60	500	1000

4.2 実験条件

比較手法として, 既存のモデルである GCN と GAT に対して, 損失関数として交差エントロピーのみ使用した手法, 損失関数として交差エントロピーにカーネルの要素を加えた手法を用いてノード分類精度について評価する. 実験で用いたソースコードは pytorch_geometric の GitHub にて提供されている GCN, GAT のソースコードをベースに作成した [16]. 2 層の GCN を学習に使用し. 訓練データとしてクラスラベルごとに 20 個のラベル付きデータ, 検証データとして 500 個のラベル付きデータ, テストデータとして 1000 個のラベル付きデータを使用するものとする. ADAM を最適化手法として用いる. 学習率は GCN について 0.01, GAT について 0.005 とする. 10 回実験を行いその平均と標準偏差を評価する.

4.3 実験結果

実験結果を表 3 および表 4 に示す. 最も精度が高い数値を太字で示し, 2 番目に高い数値を下線にて示す. なお, 表における {GCN, GAT}-default は損失関数として交差エントロピーのみを使った手法, {GCN, GAT}-Kernel は従来手法である損失関数を交差エントロピーにカーネルの要素を加えた手法, {GCN, GAT}- s_W/s_B は, 式 (4) に示した損失関数として交差エントロピーに α 倍した s_W/s_B を加えた手法 {GCN, GAT}- $1/s_B$ は, 式 (5) に示した損失関数として交差エントロピーに α 倍した $1/s_B$ を加えた手法を表している.

表 3 GCN を用いたノード分類精度 (%)

手法	α	Cora	Citeseer	Pubmed
GCN-default	-	80.66 \pm 0.53	70.96 \pm 0.59	78.77 \pm 0.46
GCN-Kernel	-	79.84 \pm 1.05	69.72 \pm 0.13	78.75 \pm 0.47
GCN- s_W/s_B	0.0001	80.83 \pm 0.61	70.89 \pm 0.50	79.05 \pm 0.47
GCN- $1/s_B$	0.005	81.81 \pm 0.67	70.90 \pm 0.78	79.11 \pm 0.41
GCN- $1/s_B$	0.001	80.77 \pm 0.52	71.10 \pm 1.16	<u>79.10 \pm 0.51</u>

表 4 GAT を用いたノード分類精度 (%)

手法	α	Cora	Citeseer	Pubmed
GAT-default	-	82.80 \pm 0.66	71.50 \pm 0.82	<u>77.62 \pm 0.28</u>
GAT-Kernel	-	82.76 \pm 0.42	71.36 \pm 0.86	77.52 \pm 0.60
GAT- s_W/s_B	0.0001	82.75 \pm 0.38	71.56 \pm 1.12	77.60 \pm 0.60
GAT- $1/s_B$	0.005	82.96 \pm 0.42	<u>71.74 \pm 0.80</u>	77.55 \pm 0.56
GAT- $1/s_B$	0.001	<u>82.91 \pm 0.40</u>	71.92 \pm 0.66	77.66 \pm 0.56

表 3 および表 4 からすべてのデータセットにおいて既存手法の精度を上回ることが分かる。このことから、提案手法は特定のデータセットに対してのみ効果を示すのではなく、汎用性のある手法であることが分かる。また、提案手法の中でも s_W/s_B を損失関数に加える手法よりも $1/s_B$ を用いた手法の方が高い精度を示している。提案した損失関数を加えたとき、GCN と GAT とを比較したとき、GCN のほうが全体的な精度の向上幅が大きいことが分かる。

5. まとめ

本稿では、グラフノード分類タスクにおいて、分類のみを目的とした交差エントロピーだけでは、クラス識別に有効なノード埋め込みが困難であると言う課題に着目した。そこで、クラス間分散とクラス内分散の要素を新たに損失関数に加える手法を提案した。評価実験から、GNN の中でも代表的な手法である GCN と GAT に対して、提案手法を適用することことで、交差エントロピーのみを損失関数としたときと比べ、同等かそれ以上の性能を示すことができた。今後の課題として、今回提案したクラス間分散とクラス内分散に着目することがグラフ分野のグラフ分類タスクやリンク予測タスクにも応用できるのか調査する必要がある。

参考文献

[1] Gilmer, J. and et al., S. S. S.: Neural Message Passing for Quantum Chemistry, *International Conference on Learning Representations*.
 [2] You, J. and et al., B. L.: Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation, *Advances in Neural Information Processing Systems*.
 [3] Kipf, T. N. and Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks, *International Conference on Learning Representations*.
 [4] Veličković, P. and et al., G. C.: Graph Attention Networks, *International Conference on Learning Representations*.
 [5] Vaswani, A. and et al., N. S.: Attention Is All You Need,

International Conference on Learning Representations.
 [6] Tian, Y. and et al., L. Z.: Rethinking Kernel Methods for Node Representation Learning on Graphs, *Advances in Neural Information Processing Systems*.
 [7] Tang, Z. and et al., X. P.: Quantized Densely Connected U-Nets for Efficient Landmark Localization, *European Conference on Computer Vision*.
 [8] Tian, Y. and et al., X. P.: CR-GAN: Learning Complete Representations for Multi-view Generation, *International Joint Conference on Artificial Intelligence*.
 [9] Krizhevsky, A. and et al., I. S.: ImageNet Classification with Deep Convolutional Neural Networks ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems*.
 [10] Xu, K. and et al., C. L.: Representation Learning on Graphs with Jumping Knowledge Networks, *International Conference on Machine Learning*.
 [11] Zhang, L. and et al., H. S.: Graph Node-Feature Convolution for Representation Learning, *arXiv preprint arXiv:1812.00086*.
 [12] Draief, M. and et al., K. K.: KONG: Kernels for ordered-neighborhood graphs, *Advances in Neural Information Processing Systems*.
 [13] Togninalli, M. and et al., E. G.: Wasserstein Weisfeiler-Lehman Graph Kernels, *Advances in Neural Information Processing Systems*.
 [14] Shervashidze, N. and et al., P. S.: Weisfeiler-Lehman Graph Kernels, *Journal of Machine Learning Research*, Vol. 12, pp. 2539–2561 (2011).
 [15] Sen, P. and et al., N.: Collective Classification in Network Data, *AI Magazine*, Vol. 29, No. 3, p. 93 (2008).
 [16] Fey, M. and Lenssen, J. E.: Fast Graph Representation Learning with PyTorch Geometric.