

異なる特徴空間におけるグラフによるメタ学習 Meta-learning from Graphs in Heterogeneous Feature Spaces

莊中¹⁾ 田中 佑典²⁾ 岩田 具治²⁾

Zhong Chuang Yusuke Tanaka Tomoharu Iwata

1 はじめに

グラフにおけるエッジの予測は、機械学習における重要な課題である。例えば、Social Networking Service (SNS) で複数のユーザー間のフォロー関係を予測することや、知識グラフにおける文章の引用関係の補完などの応用例が挙げられる。これらの問題には、グラフニューラルネットワークに基づく手法がよく利用されるが、学習データ数が少ない場合に性能が低くなるという問題がある。メタ学習は、データが少ない状況において有効なアプローチであるが、既存のメタ学習法はグラフのノードが同じ特徴空間を持つことを前提としており、より多様なグラフから学習することができない。

本研究では、ノードの特徴空間が異なる複数のグラフからメタ学習できる手法 Heterogeneous Feature Spaces Graph Meta-Learning (HFGM) を提案し、目的のグラフで既知のエッジが少ない場合でも、エッジ予測の精度を向上させる。例えば、新しい SNS のフォロワー推薦システムを導入する際、他の SNS データを入手が困難であり、HFGM を使用することで公開されている多様なデータを活用し、モデルを効果的に学習させることが可能になる。また、[1] はノードの特徴空間が異なるグラフに適用できるが、データ量に応じて計算量が大幅に増加するため、大規模データを用いた学習が困難である。これに対して、本研究の提案手法はエンコーダ・デコーダタイプのメタ学習であり、計算コストを抑えつつ多様なデータを用いた学習ができる。

2 提案手法

2.1 問題設定

本研究の問題設定を図 1 に示す。メタ学習段階では、ノードの特徴の空間が違う複数の元グラフ $\{\hat{g}_d\}_{d=1}^D$ が与えられ、元グラフから生成した学習グラフ g を用いてモデルを学習する。 g のノード数と特徴数は N, I 、特徴行列と隣接行列は $\mathbf{X} \in \mathbb{R}^{N \times I}$ 、 $\mathbf{A} \in \mathbb{R}^{N \times N}$ 、ノード n の特徴 i は $x_{ni} \in \mathbb{R}$ と表記される。隣接行列はサポートエッジとクエリエッジ $\mathbf{A}_s, \mathbf{A}_q$ に割り当てる。サポートエッジは特徴行列と共にモデルに入力しノード埋め込みを計算する。クエリエッジはモデルの評価に使われる。ここでは、エッジに方向性がないと仮定したが、提案手法は方向性があるグラフにも適用できる。メタテスト段階では、既知のエッジが少ない、すべての学習グラフとは異なるグラフ g^* が与えられる。本研究の目的は、 g^* の未知のエッジを予測することである。

2.2 モデル

HFGM は、グラフの特徴行列と隣接行列を入力として、複数層の特徴毎メッセージ伝搬ネットワークとノードエンコーダを交互に繰り返すことにより、ノ

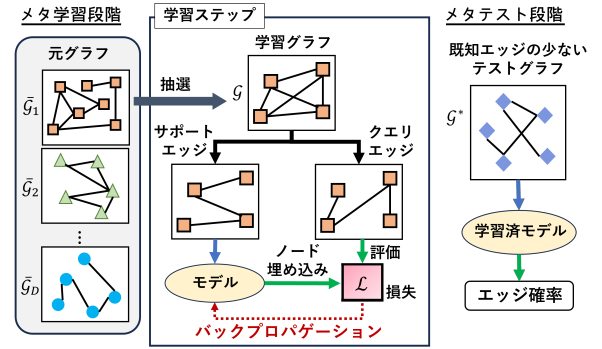


図 1 問題設定

ード埋め込みを獲得する。以下では、モデルの層数を $\ell = 0, 1, \dots, L$ で表す。HFGM の構造を図 2 に示す。すべてのグラフでニューラルネットワークが共有されるため、異なるグラフ間で知識の共有ができる。グラフ毎に最適化されたノード埋め込みの間のユークリッド距離を用いてエッジを予測する。

特徴毎メッセージ伝搬ネットワーク まず、ノード n の特徴 i の埋め込みベクトル \mathbf{v}_{ni}^0 を、隣接ノードの特徴および自身の特徴から計算する。次の層からは、ノード n の特徴 i の埋め込みベクトル \mathbf{v}_{ni}^ℓ は前の層の埋め込みを用いて計算する。

$$\mathbf{v}_{ni}^0 = \frac{1}{|\mathcal{N}_n| + 1} \left(\sum_{m \in \mathcal{N}_n} f_v^0(x_{mi}) + f_v^0(x_{ni}) \right),$$

$$\mathbf{v}_{ni}^\ell = \frac{1}{|\mathcal{N}_n| + 1} \left(\sum_{m \in \mathcal{N}_n} f_v^\ell([\mathbf{v}_{mi}^{\ell-1}, \mathbf{z}_m^{\ell-1}]) + f_v^\ell([\mathbf{v}_{ni}^{\ell-1}, \mathbf{z}_n^{\ell-1}]) \right). \quad (1)$$

ここで、 \mathcal{N}_n は隣接ノード集合、 $|\cdot|$ は要素数、 f_v^0, f_v^ℓ は Feedforward Neural Networks (FNN)、 \mathbf{z}_n^ℓ は第 ℓ 層のノード埋め込み、 $[\cdot]$ は連結関数を表す。特徴毎メッセージ伝搬は特徴空間が異なる場合にも適用でき、複数グラフで共有することが可能である。

ノードエンコーダ ノードエンコーダは特徴毎の埋め込みベクトルを集約し、ノード埋め込み \mathbf{z}_n^ℓ を計算する。

$$\mathbf{z}_n^\ell = \mathbf{g}_z^\ell \left(\frac{1}{I} \sum_{i=1}^I f_z^\ell(\mathbf{v}_{ni}^\ell) \right). \quad (2)$$

ここで、 $f_z^\ell, \mathbf{g}_z^\ell$ は FNN を表す。式 2 は、異なる特徴数に対応することができ、特徴の順番を変えても同じ出力が得られる [2]。

最終層のノード埋め込みはグラフ毎に異なる線形層により変換される。グラフ毎の線形変換行列 $\hat{\mathbf{W}}$ は、エッジがあるノード間の埋め込みのユークリッド距離を最小化しつつ、エッジがないノード間の埋め込みの距離を最大化することで得る。これは一般化固有値問題を解くことで計算できる。

エッジ予測 ノード n, m の間のエッジ確率 \hat{a}_{nm} は、グラフ毎に最適化されたノード埋め込み $\hat{\mathbf{W}}\mathbf{z}_n^\ell$ を用い

1) 奈良先端科学技術大学院大学. Nara Institute of Science and Technology.

2) NTT コミュニケーション科学基礎研究所. NTT Communication Science Laboratories.

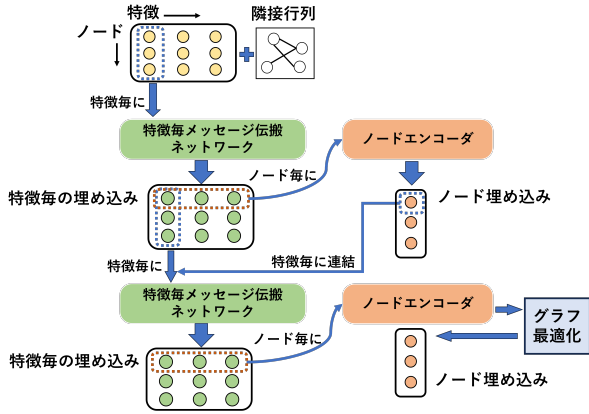


図2 提案モデル

て、ノード埋め込みのユークリッド距離を用いて予測する。

$$\hat{a}_{nm}(\mathbf{X}, \mathbf{A}, \Phi) = \sigma(-\|\hat{\mathbf{W}}\mathbf{z}_n^L - \hat{\mathbf{W}}\mathbf{z}_m^L\|^2) \quad (3)$$

ここで、 Φ は FNN のパラメーター、 σ はシグモイド関数を表す。

メタ学習 期待 AUC を最大化するように下式によりモデルを学習する。

$$\hat{\Phi} = \underset{\Phi}{\operatorname{argmin}} \mathbb{E}_{d \sim D} [\mathbb{E}_{g \sim \hat{g}_d} [\mathbb{E}_{\mathbf{A}^q \sim g} [J(\mathbf{A}^q | \mathbf{X}, \mathbf{A}^s; \Phi)]]] \quad (4)$$

ここで、 J は損失関数、 $\mathbb{E}_{d \sim D}$ はすべての元グラフの期待値、 $\mathbb{E}_{g \sim \hat{g}_d}$ は元グラフから生成した学習グラフの期待値、 $\mathbb{E}_{\mathbf{A}^q \sim g}$ は学習グラフにおけるクエリエッジの期待値を表し、モンテカルロサンプリングにより計算される。

損失関数 J はスムーズした AUC により計算される。

$$J(\mathbf{A}^q | \mathbf{X}, \mathbf{A}^s; \Phi) = -\frac{1}{\|\mathbf{A}^{q+}\|} \frac{1}{\|\mathbf{A}^{q-}\|} \sum_{\substack{i, j \in \mathbf{A}^{q+} \\ i', j' \in \mathbf{A}^{q-}}} \sigma(\|\hat{\mathbf{W}}\mathbf{z}_i^L - \hat{\mathbf{W}}\mathbf{z}_j^L\|^2 - \|\hat{\mathbf{W}}\mathbf{z}_{i'}^L - \hat{\mathbf{W}}\mathbf{z}_{j'}^L\|^2) \quad (5)$$

ここで、 \mathbf{A}^{q+} はクエリエッジ、 \mathbf{A}^{q-} はネガティブサンプルとして、エッジがないノードから \mathbf{A}^{q+} と同じ数で抽選したノードの集合である。式 5 を最小化することで、エッジがあるノード埋め込み間のユークリッド距離を最小化しつつ、エッジがないノード埋め込み間のユークリッド距離を最大化することができる。

3 評価実験

3.1 設定

本研究では、SNS や文献参照ネットワークなどの 14 の実世界グラフを使用して評価実験を行った。これらのグラフの特徴は、0 と 1 の間に正規化した。サブグラフの生成は、元グラフから 500 ノードと 70 特徴を一樣にサンプリングし、孤立ノードを削除した後、残りのエッジを半分ずつサポートエッジとクエリエッジに割り当てることを行った。学習段階では、8 種類の元グラフを使用し、1 バッチあたり 50 の学習グラフを生成した。検証とテスト段階では、それぞれ 2 種類および 4 種類の元グラフから 100 グラフを生成した。

表 1 テストグラフに対する平均 AUC(± 標準偏差)

モデル	AUC スコア
HFGM	0.650±0.021
DSGCN	0.584±0.071
MGCN	0.415±0.032
NN	0.558±0.024
GCN	0.456±0.051
GAT	0.476±0.049

メタ学習の比較手法としては、Deep Set [2] と Graph Convolutional Networks (GCN) を併用したモデル (DSGCN)、および MAML [3] を用いて GCN を最適化するモデル (MGCN) を使用した。ただし、MGCN は異なる特徴に対応できないため、特徴なしグラフにより学習した。Neural Networks (NN)、GCN、Graph Attention networks (GAT) [4] はメタ学習でない比較手法として、テストグラフ毎にモデルを学習した。これらの比較手法は、埋め込みの内積を用いてエッジを予測し、交差エントロピーを用いて学習した。

HFGM の層数 L は 2 に設定される。HFGM およびすべての比較手法が使う FNN は 3 層構造、中間層と出力層は 32 チャンネルとした。活性化関数は $ReLU(x) = \max(0, x)$ を用いた。最適化法としては Adam を用い、学習率は 10^{-3} 、ドロップアウトは 10^{-1} 。HFGM と DSGCN はそれぞれ 50,000 エポック、MGCN と非メタ学習手法は 500 エポックで学習した。

3.2 結果

学習・検証・テストグラフを生成する元グラフをランダムに変更し、10 回の実験を行った。実験の結果は平均 AUC を表 1 に示す。提案法である HFGM は、最も高い精度を達成した。特徴毎メッセージ伝搬ネットワークとノードエンコーダはノードの特徴の空間が異なるグラフから学習ができ、グラフ毎の最適化は効果的であった。次に高い精度を示したの DSGCN は、異なるノード特徴空間の学習ができるが、グラフ毎の最適化をせず、より高い精度を達成できなかった。MGCN は特徴を利用できないため、低い性能であった。非メタ学習法の NN、GCN、GAT は他のグラフの知識を利用できないため、低い性能であった

4 終わりに

本研究では、異なるノード特徴空間を持つ複数のグラフからメタ学習できるモデルを提案し、目的のグラフはエッジデータが少ないグラフにおいても、既存手法より高い予測精度を達成することを実験において示した。時系列特徴や複数種類のノードを持つグラフへの拡張が今後の発展として考えられる。

参考文献

- [1] Lin, Mingkai, et al. "Multi-domain generalized graph meta learning." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. No. 4. 2023.
- [2] Zaheer, Manzil, et al. "Deep sets." *Advances in Neural Information Processing Systems* 30 (2017).
- [3] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." *International conference on machine learning*. PMLR, 2017.
- [4] Velickovic, Petar, et al. "Graph attention networks." *stat* 1050.20 (2017): 10-48550.