情報処理学会論文誌 Vol.62 No.1 357–368 (Jan. 2021)

非教示なグラフ分散表現のエッジ特徴による改良

陳 宏[1,a)] 古賀 久志[1,b)]

受付日 2020年4月28日, 採録日 2020年10月6日

概要：近年，様々なグラフデータセットの分散表現を学習する手法が提案された．その 1 つである Graph2vec はグラフ分類に有用なグラフ全体の分散表現を非教示で学習可能である．本論文ではまず，Graph2vec には（1）エッジラベルを取り扱えない，（2）ノードラベルと構造情報を同時に分散表現に畳み込むため，構造の類似性を適切に判定できないことがあるという 2 つの課題があることを述べる．本論文では，これら 2 つの課題をライングラフ（edge-to-vertex dual graph）を用いて解決する手法を提案する．提案手法では，Graph2vec では考慮できない元グラフ G のエッジラベルや構造情報をエッジ特徴として表現後，ライングラフ LG のノード特徴に変換する．そして，G の分散表現に LG の分散表現を連結することで，エッジラベルや構造情報が補完された分散表現を生成する．この提案手法を GL2vec（Graph and Line graph to vector）と名付ける．実験により，GL2vec が多くのベンチマークデータセットに対してグラフ分類性能を Graph2vec より改善できることを示す．さらに，グラフとライングラフの分散表現を連結するという GL2vec のアプローチは，任意の非教示型の分散表現学習手法にも適用できる．実際に Graph2vec 以外の分散表現学習手法でも有効性を確認した．

キーワード：ライングラフ，グラフの分散表現，グラフベースパターン認識

Unsupervised Graph Embedding Improved by Edge Features

Hong Chen[1,a)] Hisashi Koga[1,b)]

Received: April 28, 2020, Accepted: October 6, 2020

Abstract: Recently, how to learn the distributed representation for a given graph dataset is intensively studied. Among them, Graph2vec unsupervisedly learns the distributed representation of entire graphs that is useful for graph classification. This paper first points out two drawbacks of Graph2vec: (1) Edge labels cannot be handled and (2) Graph2vec cannot always evaluate the structural similarity properly, because the node label and the structural information are embedded into the distribution representation at the same time. This paper proposes a method to cope with the two drawbacks that exploits the line graphs (edge-to-vertex dual graphs) of given graphs. Especially, our method complements either the edge labels or the structural information with the distributed representation of the line graphs. Then, it appends the distributed representation of the line graph to that of the original graph. Experimentally, our method achieves significant improvements in graph classification task over Graph2vec for many benchmark datasets. The approach of GL2vec is applicable to any other unsupervised methods to learn distributed representations for graphs than Graph2vec.

Keywords: line graph, distributed representations of graphs, graph based pattern recognition

1. はじめに

グラフはノード（頂点）の集合とエッジ（辺）の集合で構成されるデータ構造である．グラフは複雑なシステムを記述する強力なツールとして，ソーシャルネットワーク分析，生物情報科学，化学情報科学，コンピュータビジョン，

[1] 電気通信大学
The University of Electro-Communications, Chofu, Tokyo 182–8585, Japan
[a)] chen@sd.is.uec.ac.jp
[b)] koga@sd.is.uec.ac.jp

自然言語処理などの多くの分野で使用されている．

そして，グラフデータを既存のパターン認識アルゴリズムで処理するために，ニューラルネットワークを用いてグラフの数値特徴ベクトルを学習するアプローチがさかんに研究されている．ニューラルネットワークで学習されたグラフの数値特徴ベクトルはグラフの分散表現とも呼ばれる．自然言語処理における Word2vec [1] や Doc2vec [2] などの分散表現の成功を受けて，言語モデルをベースとするグラフ分散表現が数多く提案されている．たとえば，Node2vec [3] はノード分類やリンク予測に有用なノードに対する分散表現を学習する．

ノードなどの部分構造に対して分散表現を学習する手法に較べると，グラフ分類やクラスタリングへの応用を想定しグラフ全体に対する分散表現を学習する手法は数が少ない．PATCHY-SAN [4]，DIFFPOOL [5]，GIN（Graph Isomorphism Network）[6] などは教示ありで分散表現を獲得する手法である．これらは学習データのクラスラベルも学習に利用するため，非教示型手法よりも高いグラフ分類精度を達成する．しかし，学習データのクラスラベルを準備するコストが大きい点が短所である．また，得られた分散表現はクラス分類に特化しているため，クラスタリングなど別のタスクに適用したときの汎用性に欠ける．

本研究は，グラフ全体に対して非教示で分散表現を学習する手法を研究対象とする．非教示型の手法は，学習時に分類タスク用のクラスラベルが不要であるため，学習データを用意するコストが小さいという利点を有する．非教示でグラフ全体に対する分散表現を学習する手法には文献 [7], [8], [9] などがあるが，教示あり手法と較べると研究例が少ない．その中で，Graph2vec [7] はノードラベル付きのグラフ集合から，非教示で，個々のグラフに対する分散表現を獲得する手法の代表例である．

本研究は Graph2vec を改良する手法を提案する．本論文ではまず，Graph2vec には (1) エッジラベルを取り扱えない，(2) 部分グラフを量子化する際にノードラベルと構造情報を同時に畳み込むため，構造の類似性を適切に判定できないという 2 つの限界があることを指摘する．本論文では，これら 2 つの課題をライングラフを用いて解決することを提案する．ライングラフは edge-to-vertex dual graph とも呼ばれ，グラフ G に対するライングラフ LG は，G のエッジを LG のノードにマッピングすることで生成される．これにより，LG では G のエッジ特徴をノード特徴として保有できる．さらに LG は，G のノードラベル情報を持たないので，G のノードラベルと独立に構造の類似性判定を行うのに適する．

提案手法では，Graph2vec では考慮できない元グラフ G のエッジラベルや構造情報をエッジ特徴として表現後，ライングラフ LG のノード特徴に変換する．そして，G の分散表現に LG の分散表現を連結することで，エッジラベルや構造情報を G の分散表現に補完する．この提案手法を GL2vec（Graph and Line graph to vector）と名付ける．実験により，GL2vec が多くのベンチマークデータセットに対してグラフ分類性能を Graph2vec より改善することを示す．本論文では，直接的には Graph2vec の改良手法を提案するが，グラフの分散表現とライングラフの分散表現を連結するというフレームワーク自体は，任意の非教示型の分散表現学習手法にも適用可能であり，汎用性が高い．

以下に本論文の構成を示す．まず 2 章で従来手法である Graph2vec を説明する．3 章では，Graph2vec の 2 つの限界を指摘する．4 章で，その 2 つの課題を克服する我々の手法 GL2vec を提案する．5 章は実験評価，6 章は関連研究である．7 章では，グラフの分散表現とライングラフの分散表現を連結するというフレームワークが Graph2vec 以外の非教示型の分散表現学習手法にも適用できることを実験的に示す．8 章は結論である．なお，本論文は我々の国際会議論文 [10] を発展させたものである．

2. Graph2vec [7]

Graph2vec はノードラベル付きグラフに対する分散表現を出力する．ここで，ノードラベル付きグラフ G を $G = (V, E)$ で表す．V はノード集合であり，$E \subset (V \times V)$ はエッジ集合である．各ノード $v \in V$ にはアルファベットのラベル $\lambda(v)$ が付与されている．

Graph2vec は，グラフの集合 $\{G_1, G_2, \ldots, G_N\}$ と特徴ベクトルの次元数 δ を与えられて，$\{G_1, G_2, \ldots, G_N\}$ から特徴ベクトルの集合 $\{\vec{G_1}, \vec{G_2}, \ldots, \vec{G_N}\}$ への写像を学習する．Graph2vec は自然言語処理におけるドキュメント（文書）の特徴ベクトルを学習するモデルである Doc2vec [2] を基にしている．Doc2vec では文書を単語集合として表し，ニューラル言語モデル skip-gram [1] を拡張した言語モデル DBOW [2] を用いて文書の分散表現を学習する．これに対して，Graph2vec では，1 つのグラフを局所的な根付き部分グラフの集合として表す．その後，グラフを文書とし，根付き部分グラフを単語として，DBOW に入力し各グラフの分散表現を学習する．

以下では，2.1 節でグラフから根付き部分グラフ集合を抽出する過程を説明し，2.2 節では構築した根付き部分グラフの集合からグラフの特徴ベクトルを学習する過程を説明する．

2.1 根付き部分グラフの抽出

Graph2vec では，まず個々のグラフ G から根付き部分グラフの集合 $c(G)$ を抽出する．最初に，根付き部分グラフの最大の高さ H をパラメータとして指定する．そして，G 内のすべてのノード v に対し，v を根とする $(H+1)$ 個の根付き部分グラフを生成する．この結果，G のノード数を n とすると，式 (1) のような $n(H+1)$ 個の根付き部

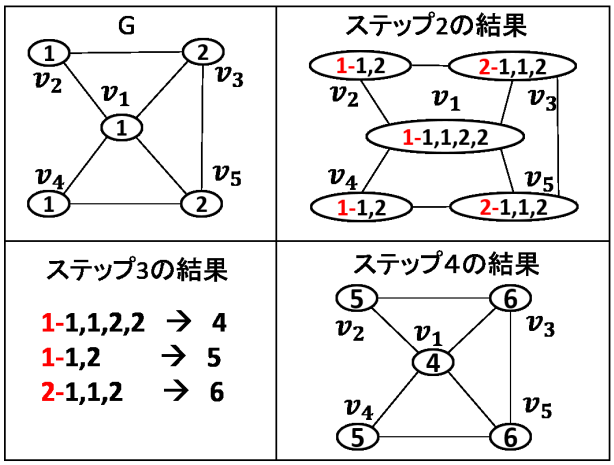

図 1 WL リラベリング
Fig. 1 WL Relabeling.

分グラフが生成され集合 $c(G)$ を構成する．ここで，$sg_i^{(t)}$ は i 番目のノード v_i を根とする，高さ t の根付き部分グラフである．直感的には部分グラフ $sg_i^{(t)}$ は v_i を中心とする t ホップの情報を記述する．これらの根付き部分グラフは WL（Weisfeiler-Lehman）リラベリング [11] により識別 ID に量子化される．

$$c(G)=\{sg_1^{(0)},sg_2^{(0)},\ldots,sg_n^{(0)},sg_1^{(1)},sg_2^{(1)},\ldots,sg_n^{(1)},\ldots,sg_1^{(H)},sg_2^{(H)},\ldots,sg_n^{(H)}\} \quad (1)$$

以下に，各ノードに対して高さ t の根付き部分グラフを生成し，識別 ID へ量子化する WL リラベリングのアルゴリズムを示す．このアルゴリズム内で，$\lambda^t(v)$ はノード v のラベルであるが，v を根とする高さ t の根付き部分グラフ $sg_v^{(t)}$ の識別 ID を表している．

(1) $\forall v \in V$ に対し，ノード v の隣接ノードのラベルを集めた多重集合 $M^t(v)=\{\lambda^{t-1}(u)|u \in \mathrm{Neighbors}(v)\}$ を作る．
(2) $M^t(v)$ の要素を昇順でソートした文字列 $S^t(v)$ を作る．その後，$S^t(v)$ の先頭に根ノードのラベル $\lambda^{t-1}(v)$ を連結する．
(3) 全単射なハッシュ関数を用いて $S^t(v)$ を識別 ID $\mathrm{Hash}(S^t(v))$ にマッピングする．すなわち，$\mathrm{Hash}(S^t(v))=\mathrm{Hash}(S^t(w))$ iff $S^t(v)=S^t(w)$．Hash は，radix ソートを使えば簡単に実装できる．
(4) ノード v の新ラベル $\lambda^t(v)=\mathrm{Hash}(S^t(v))$ とする．

ここで，ステップ (1)，(2) では，v の隣接ノードの高さ $t-1$ の根付き部分グラフおよび v 自身の高さ $t-1$ の根付き部分グラフから，v の高さ t の根付き部分グラフ $sg_v^{(t)}$ を合成し，ステップ (3)，(4) で根付き部分グラフ $sg_v^{(t)}$ を識別 ID である $\lambda^t(v)$ へ量子化している．WL リラベリングの実行例を**図 1** に示す．この図ではたとえば，v_5 を根とする高さ 1 の部分グラフ $sg_{v_5}^{(1)}$ が “2-1,1,2” と表記されている．ここで，先頭の “2” は根ノード v_5 のラベル，“1,1,2” は v_5 に隣接するノード v_1, v_4, v_3 のラベルである．ステップ (3) では $sg_{v_5}^{(1)}$ を表す “2-1,1,2” がハッシュ関数により識別 ID “6” にマッピングされる．こうして WL リラベリングを 1 回実行後，グラフ G から抽出された根付き部分グラフの集合 $c(G)=\{sg_{v_1}^{(0)},sg_{v_2}^{(0)},sg_{v_3}^{(0)},sg_{v_4}^{(0)},sg_{v_5}^{(0)},sg_{v_1}^{(1)},sg_{v_2}^{(1)},sg_{v_3}^{(1)},sg_{v_4}^{(1)},sg_{v_5}^{(1)}\}$ は識別 ID の多重集合 $c(G)=\{1,1,2,1,2,4,5,6,5,6\}$ で表現される．この WL リラベリングを H 回繰り返せば，高さ H までの $n(H+1)$ 個の根付き部分グラフ（= 識別 ID の多重集合）を抽出できる．

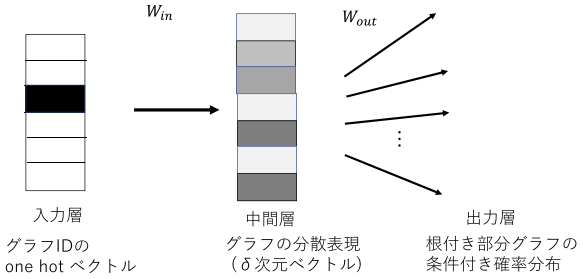

図 2 Graph2vec のネットワークモデル
Fig. 2 Network Model of Graph2vec.

2.2 分散表現の学習

すべてのグラフから根付き部分グラフ集合を抽出後，Graph2vec は DBOW モデルを用いて，各グラフの特徴ベクトルを学習する．DBOW は自然言語処理分野で文書の特徴ベクトルを算出するための言語モデルであり，隠れ層を 1 つだけ持つ順伝搬ニューラルネットワークである．

より詳しく述べると，グラフの集合 $\{G_1,G_2,\ldots,G_N\}$ と対応する根付き部分グラフの集合 $\{c(G_1),c(G_2),\ldots,c(G_N)\}$ を与えられたとき，Graph2vec は次元数 δ の各グラフ G_i の特徴ベクトル $\vec{G_i}$ と各部分グラフ sg の特徴ベクトル $\vec{sg}$ を学習する．Graph2vec のネットワークモデルを**図 2** に示す．入力層では，1 つのグラフ G をグラフ ID の one-hot ベクトルで表して入力する．入力層と中間層の間の重みは $\delta \times N$ の行列 W_{in} である．出力層では，G が入力されたという条件下で各根付き部分グラフを含む条件付き確率分布を出力する．重みを表す行列 W_{in} はグラフが内包する部分グラフを予測できるように学習する．具体的には，G_i から部分グラフ集合 $c(G_i)$ が抽出された事実に着目し，式 (2) の対数尤度関数を最大化するようにネットワークの重みを学習する．

$$\sum_{j=1}^{n_i(H+1)} \log P_r(sg_j|G_i) \quad (2)$$

ここで，n_i は G_i のノード数であり，sg_j は G_i から抽出された j 番目の根付き部分グラフである．確率 $P_r(sg_j|G_i)$ は式 (3) で定義される．

$$\frac{\exp(\vec{G_i}\cdot\vec{sg_j})}{\sum_{sg\in Voc}\exp(\vec{G_i}\cdot\vec{sg})} \quad (3)$$

ここで，Voc は全グラフにわたるすべての根付き部分グラフの集合である．また，グラフの特徴ベクトル $\vec{G_i}$ は式 (4) で定義される．

$$\vec{G_i} = W_{in} \cdot onehot(G_i). \tag{4}$$

つまり，$\vec{G_i}$ は W_{in} の i 列目である．つまり，W_{in} の各列が 1 つのグラフに対する分散表現となる．このようにして，様々なサイズのグラフが固定長の特徴ベクトルに変換される．

Graph2vec では，G_i と G_j が共通の根付き部分グラフを多く持つほど，$\vec{G_i}$ と $\vec{G_j}$ が特徴ベクトル空間内で近くなり，似ていると見なされる．

3. Graph2vec の限界

本章では，本研究に取り組むきっかけとなった Graph2vec の欠点を 2 つ指摘する．

1 つ目の欠点は，たとえ分類したいグラフデータにエッジラベルが付与されていても，Graph2vec はエッジラベルを利用できないということである．2.1 節で説明したとおり，Graph2vec は WL リラベリングを用いて根付き部分グラフを抽出する際に，ノードを対象にラベルの集約と更新を行う．その過程で，エッジの属性情報は完全に無視される．その結果，根付き部分グラフのコーパスにはエッジ属性が反映されず，そのようなコーパスを用いて学習したグラフの特徴ベクトルもエッジ属性を反映しない．しかし，エッジ属性は化合物における原子の結合タイプを表したり，データセットによっては重要な情報を含む．もしそれを活用できれば，そのようなデータセットに対し，より高いパターン認識精度を達成できる可能性がある．

2 つ目の欠点は抽出した根付き部分グラフを識別 ID へマッピングする際に，ノードラベルとグラフ構造を同時に畳み込むため，根付き部分グラフの識別 ID の多重集合 $c(G)$ が，G と他のグラフとの構造類似性を評価するのに適さない場合があることである．一般に，ノードラベル付きグラフの類似性は (1) ノードラベルの類似性と (2) 構造つまりグラフ形状の類似性の両者で決定される．Graph2vec は根付き部分グラフの識別 ID にノードラベルとグラフ構造の両者を同時に畳み込む．その結果，ノードラベルが一致している条件でのみ構造の同一性が評価されることになり，ノードラベルと独立に 2 グラフの構造が似ていることを判別するのは難しい．この現象を，**図 3** を例に説明する．

図 3 に示された 2 つのグラフ G，$G*$ は形状は完全一致しており，中央ノードだけがノードラベルが異なり，残りの 4 ノードのラベルは一致している．G と $G*$ に対して高さ $H = 1$ までの根付き部分グラフを抽出すると，その識別 ID の多重集合 $c(G)$ と $c(G*)$ は**図 4** のようになる．たとえば，グラフ G における v_5 を根とする部分グラフ $sg_{v_5}^{(1)}$ が

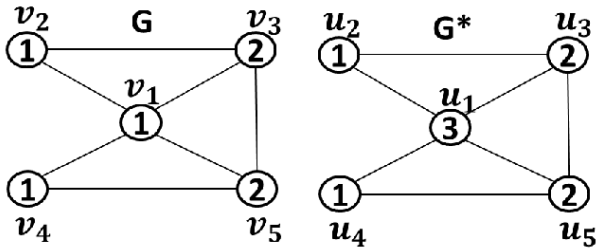

図 3 2 つのグラフ G，G*
Fig. 3 Graphs G and G*.

c(G)	$sg_{v_1}^{(0)}$	$sg_{v_2}^{(0)}$	$sg_{v_3}^{(0)}$	$sg_{v_4}^{(0)}$	$sg_{v_5}^{(0)}$	$sg_{v_1}^{(1)}$	$sg_{v_2}^{(1)}$	$sg_{v_3}^{(1)}$	$sg_{v_4}^{(1)}$	$sg_{v_5}^{(1)}$
	1	1	2	1	2	4 (1-1,1,2,2)	5 (1-1,2)	6 (2-1,1,2)	5 (1-1,2)	6 (2-1,1,2)
c(G*)	$sg_{u_1}^{(0)}$	$sg_{u_2}^{(0)}$	$sg_{u_3}^{(0)}$	$sg_{u_4}^{(0)}$	$sg_{u_5}^{(0)}$	$sg_{u_1}^{(1)}$	$sg_{u_2}^{(1)}$	$sg_{u_3}^{(1)}$	$sg_{u_4}^{(1)}$	$sg_{u_5}^{(1)}$
	3	1	2	1	2	9 (3-1,1,2,2)	7 (1-2,3)	8 (2-1,2,3)	7 (1-2,3)	8 (2-1,2,3)

図 4 部分グラフの識別 ID の多重集合 $c(G)$ と $c(G*)$
Fig. 4 Multisets of Subgraph IDs.

“2-1,1,2” で，識別 ID “6” にマッピングされる．一方，グラフ G*における u_5 を根とする部分グラフ $sg_{u_5}^{(1)}$ が “2-1,2,3” で識別 ID “8” にマッピングされる．$sg_{v_5}^{(1)}$ と $sg_{u_5}^{(1)}$ は形状は一緒であるが，ノードラベルの違いにより異なる識別 ID にマッピングされる．$c(G)$ と $c(G*)$ から判別できることは，G と $G*$ が同じラベルのノードを 4 つ持つことだけである．G と $G*$ の形状が同一であることはおろか，同一ラベルを持つ v_5 と u_5 のノード次数が一緒であることさえ判別困難である．この原因は，形状が同じ根付き部分グラフでもノードラベルが 1 つでも違えば異なる識別 ID になり，同一形状という情報が捨てられるためである．

以上をまとめると，Graph2vec では，ノードラベルとグラフ構造を同時に部分グラフの識別 ID に畳み込むため，グラフの構造類似性を適切に評価できない場合がある．

4. 提案手法

本章では，3 章で指摘した Graph2vec の 2 つの課題を改善するために，ライングラフ [12] を用いて Graph2vec を拡張した GL2vec を提案する．提案手法 GL2vec を説明する前に，まず 4.1 節でライングラフを説明する．

4.1 ライングラフ

グラフ $G = (V, E)$ を与えられて，対応するライングラフ $LG = (LV, LE)$ とは，G におけるエッジの隣接関係を表すグラフである．edge-to-vertex dual graph とも呼ばれる．まず，LG の各ノードは式 (5) のように G のエッジと 1 対 1 で対応する．

$$LV = \{v(e) | e \in E\} \tag{5}$$

エッジ集合 LE は以下のルールで構築される．

$$LE = \{(v(e_i), v(e_j)) | e_i \text{ と } e_j \text{ が } G \text{ で端点を共有する}\}$$

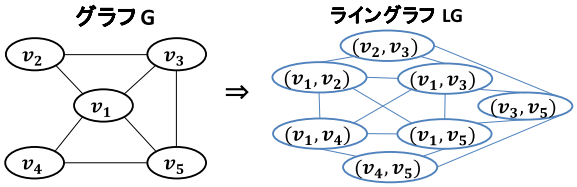

図 5 グラフ G と対応するラインングラフ LG
Fig. 5 Graph G and its Linegraph LG.

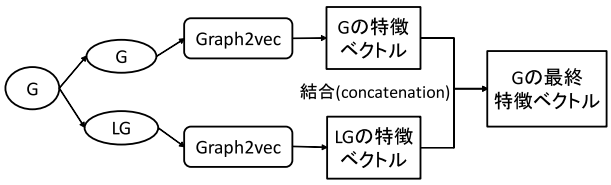

図 6 GL2vec
Fig. 6 GL2vec.

例を図 5 に示す．グラフ G におけるエッジ (v_1, v_2) とエッジ (v_1, v_5) は端点 v_1 を共用する．したがって，LG におけるノード (v_1, v_2) とノード (v_1, v_5) 間に辺が張られる．グラフ理論では，ノード v と隣接するエッジ数のことをノード v の次数 $\deg(v)$ と呼ぶ．同様に，エッジ e と隣接しているエッジ数のことをエッジ e の次数 $\deg(e)$ と呼ぶ．また，ラインングラフにおけるノード $v(e)$ の次数はグラフ G におけるエッジ e の次数と等しくなることが知られている．つまり，$\deg(v(e)) = \deg(e)$ となる．

Whitney graph isomorphism theorem [13] により，2 つの連結グラフ G_1, G_2 のラインングラフ LG_1 と LG_2 が同型であれば，唯 1 つの例外を除いて，G_1 と G_2 が同型であることが証明されており，グラフとそのラインングラフは同等な構造情報を持つ．唯一の例外は，完全グラフ K_3 と完全二部グラフ $K_{1,3}$ のラインングラフはどちらも K_3 になることである．

4.2 GL2vec

3 章で指摘したように，Graph2vec では (1) エッジラベルを扱うことができず，(2) 根付き部分グラフを量子化する際にノードラベルと構造情報を同時に畳み込むため，構造の類似性を認識できない場合がある．本研究では，Graph2vec では取り扱うのが困難なグラフ G の (1) エッジラベルと (2) 構造情報を，G に対するラインングラフ LG を活用して補う手法 GL2vec を提案する．我々がラインングラフに着目した理由は以下の 2 つである．

- G のエッジ e が LG のノード $v(e)$ に変換されるので，G のエッジラベルを LG のノードラベルとして扱える．
- LG は G のノードラベル情報を持たないので，G のノードラベルと独立に，構造の類似性を評価するのに適している．

GL2vec では，元グラフ G のエッジラベルと構造情報のどちらかをユーザが選択し，LG の特徴ベクトルに反映する．典型的には，G がエッジラベルを持つ場合，G のエッジラベルを LG のノードラベルとして与え，G のエッジラベルに基く LG の特徴ベクトルを獲得する．G がエッジラベルを持たない場合は，ラインングラフ LG のノード次数，すなわち G のエッジ次数を LG のノードラベルとし，G のノードラベルとは独立に G の構造情報を LG の特徴ベクトルに畳み込む．G のエッジ次数は G のグラフ形状のみによって決定される特徴であることに注意されたい．

ここで，G の構造情報に関しては，わざわざラインングラフを作らなくても G のノード次数を G のノードラベルに設定すれば，G の特徴ベクトルに畳み込めるのではないかと思われるかもしれない．それは事実だが，ラインングラフ LG の特徴に畳み込んだ場合は，より詳細なレベルで構造情報を記述できるという点が異なる．G のノード次数の集合と G のエッジ次数の集合はどちらも G の形状のみによって決定され，G の構造情報を表す特徴量である．ここで木以外の一般的な連結グラフでは，G のエッジ数は G のノード数より多い．したがって，G のエッジ次数の集合は，ノード次数の集合よりも多数の数値で構造情報を表現していることになり，より詳細なレベルで構造情報を記述していることになる．

LG の特徴ベクトルだけでは，G のノードラベルが無視されてしまうので，GL2vec では G の特徴ベクトルも同時に利用する．GL2vec のフレームワークを図 6 に示す．また，処理手順を以下に示す．

(1) 与えられたグラフの集合 $\{G_1, G_2, \ldots, G_N\}$ に対し，ラインングラフの集合 $\{LG_1, LG_2, \ldots, LG_N\}$ を作成する．LG_i のノードラベルは，グラフデータセットがエッジラベルを持っているか否かにより設定方法を変える．
- G_i がエッジラベルを持つ場合，G_i のエッジラベルを LG_i のノードラベルに設定する．
- G_i がエッジラベルを持たない場合，G_i のエッジ次数を LG_i のノードラベルに設定する．

(2) $\{G_1, G_2, \ldots, G_N\}$ に対して Graph2vec を適用し，各 G_i に対して δ 次元の特徴ベクトル $\vec{G_i}$ を得る．

(3) $\{LG_1, LG_2, \ldots, LG_N\}$ に対して Graph2vec を適用し，LG_i に対して δ 次元の特徴ベクトル $L\vec{G_i}$ を得る．

(4) G_i に対して，$\vec{G_i}$ と $L\vec{G_i}$ を結合した 2δ 次元のベクトルを G_i に対する最終的な特徴ベクトルとする．

最後に，ノードラベルとエッジラベルのどちらも持たないデータセットの処理について説明する．こうしたデータセットに対しては，Graph2vec ではノード次数を G_i のノードラベルに設定することを推奨しており，G_i の分散表現 $\vec{G_i}$ には G_i の構造情報が反映される．一方で，LG_i のノードラベルには G_i のエッジ次数が設定され，$L\vec{G_i}$ にも G_i の構造情報が反映される．しかし，上述したように $L\vec{G_i}$ には $\vec{G_i}$ よりも詳細なレベルの構造情報が反映される．

GL2vec では，$\vec{G_i}$ と $L\vec{G_i}$ を連結することで，粒度が異なる 2 レベルで構造情報を評価できる．

5. 実験評価

提案手法を評価するため，生物情報，化学情報，ソーシャルネットワークに関するベンチマークデータセットを用いて，グラフ分類（2 クラス）実験を実施する．

5.1 グラフデータセット

(1) エッジラベルなしのデータセット 6 個と (2) エッジラベルありのデータセット 7 個の合計 13 個のデータセットを用意した．各データセットの統計情報（サンプル数，平均ノード数，ノードラベル種類数，エッジラベル種類数）を**表 1** に示す．

5.1.1 エッジラベルなしデータセット

エッジラベルなしのデータセットは，MUTAG [7]，PTC [7]，PROTEINS [7]，NCI1 [7]，NCI109 [7] および IMDB-B [15] の 6 個である．最初の 5 つは文献 [7] で Graph2vec の評価に使用されていたものである．

MUTAG，PTC，NCI1，NCI109 は化学情報分野からのデータセットである．化合物をグラフに変換したときには，ノードが原子を表し，エッジが化学結合を表す．ノードは原子タイプのラベルを持つ．MUTAG データセットは，細菌に対する変異原性の影響に応じて 2 クラスに分類された 188 個の化合物で構成される．PTC データセットは 344 個の化合物から成り，ネズミにおける発がん性を示している．NCI1 と NCI109 はそれぞれ非小細胞肺癌細胞株と卵巣癌細胞株に対する活性についてスクリーニングされた化合物データセットのサブセットである．

PROTEINS はタンパク質に関するデータセットでノードが SSE と呼ばれる 2 次構造要素を表し，エッジはアミノ酸配列または 3D 空間における近傍性を表す．

表 1 データセットの統計量
Table 1 List of datasets.

Dataset	#samples	#nodes (average)	#node labels	#edge labels
MUTAG	188	17.9	7	-
PTC	344	25.5	19	-
PROTEINS	1,113	39.1	3	-
NCI1	4,110	29.8	37	-
NCI109	4,127	29.6	38	-
IMDB-B	1,000	19.8	-	-
MUTAG*	188	17.9	7	4
NCI33	2,843	30.2	29	4
NCI81	4,030	29.6	31	4
NCI83	3,867	29.5	28	4
NCI123	5,260	28.8	33	4
NCI145	3,182	30.3	27	4
DBLP	19,456	100.5	41,324	3

IMDB-B はグラフがノードラベルもエッジラベルも持たないデータセットである．このデータセットは，映画コラボレーションデータベースから変換されたグラフで構成される．より具体的には，「アクション」と「ロマンス」の 2 ジャンルの映画コラボレーションネットワークを最初に構築する．このネットワークのノードは俳優/女優であり，同じ映画に登場する場合はそれらの間にエッジがある．次に，この 2 つのネットワークから各俳優/女優の ego-network を抽出しグラフデータとする．グラフ分類タスクでは，各 ego-network を表すグラフが「アクション」と「ロマンス」のどちらに所属するかを判定する．

5.1.2 エッジラベルありデータセット

エッジラベルありのデータセットは，MUTAG* [14]，NCI33 [16]，NCI81 [16]，NCI83 [16]，NCI123 [16]，NCI145 [16] および DBLP [16] の 7 個である．

MUTAG*，NCI33，NCI81，NCI83，NCI123，NCI145 は化学情報分野からのデータセットである．MUTAG*データセットは MUTAG と同じで，さらにエッジが化学結合タイプでラベル付けされている．NCI33，NCI81，NCI83，NCI123 および NCI145 はそれぞれ黒色腫の癌細胞株，大腸癌細胞株，乳癌細胞株，白血病細胞株，および腎臓癌細胞株に対する活性についてスクリーニングされた化合物データセットである．エッジは化学結合タイプでラベル付けされている．

DBLP データセットは，コンピュータサイエンスの参考文献データから変換されたグラフで構成されている．各論文は下記のルールでグラフへ変換される．(1) 各論文は論文ノードになる．(2) 論文タイトルに含まれるキーワードはキーワードノードになり，論文ノードと接続される．また，同じ論文ノードに接続されたキーワードノード群は全結合する．(3) 論文の間に引用関係があれば，対応する論文ノードペアにエッジを張る．論文ノードは論文番号を，キーワードノードはキーワードをノードラベルとして持つ．また，エッジには両端点が論文ノードであるかキーワードノードであるかを表すラベルが付与される．つまり，エッジラベルは（論文–論文），（論文–キーワード），（キーワード–キーワード）の 3 種類である．グラフ分類タスクは，各論文の分野が「データベースとデータマイニング」あるいは「コンピュータビジョンとパターン認識」のどちらであるかを推定する 2 クラス分類問題である．

5.2 実験設定

GL2vec は，既存の Graph2vec ソフトウェアパッケージをサブルーチンとして呼び出して実装した．Graph2vec のパラメータは，根付き部分グラフの最大の高さ $H = 3$ とし，特徴ベクトルの次元数 $\delta = 1{,}024$ に設定する．グラフ G の特徴ベクトルとラインググラフ LG の特徴ベクトルを連結するため，GL2vec は 2,048 次元の特徴ベクトルを生成

表 2 エッジラベルなしデータセットに対する分類正解率（平均 ± 標準偏差）%
Table 2 Classification accuracy for datasets without edge labels.

データセット名	MUTAG	PTC	PROTEINS	NCI1	NCI109	IMDB-B
GL2vec	**86.58**±7.34	60.14±8.21	69.64±3.87	**79.50**±2.17	**79.17**±1.87	**73.65**±5.61
Graph2vec	83.95±7.14	60.86±8.75	70.85±5.46	78.47±2.00	76.40±2.34	72.00±4.41
p 値	0.047	0.74	0.33	$< 10^{-2}$	$< 10^{-4}$	0.032
LineGraph2vec	85.79±9.82	52.14±9.26	64.69±6.76	72.71±1.92	71.19±2.20	72.55±6.00

表 3 エッジラベルありデータセットに対する分類正解率（平均 ± 標準偏差）%
Table 3 Classification accuracy for datasets with edge labels.

データセット名	MUTAG*	NCI33	NCI81	NCI83	NCI123	NCI145	DBLP
GL2vec	**88.16**±7.80	**81.44**±2.81	**80.47**±2.00	**77.00**±1.90	**74.91**±2.28	**78.59**±2.34	**91.58**±0.70
Graph2vec	83.95±7.14	79.50±2.56	77.26±2.38	75.27±1.92	72.16±1.90	76.25±2.85	90.14±0.90
p 値	0.019	$< 10^{-4}$	$< 10^{-7}$	$< 10^{-2}$	$< 10^{-7}$	$< 10^{-3}$	$< 10^{-8}$
LineGraph2vec	87.63±7.30	76.81±2.92	76.65±1.73	72.47±1.69	70.89±1.88	74.78±2.27	82.49±0.71
Graph2vec_edge	83.95±8.95	78.33±2.61	77.58±1.60	75.66±2.34	72.98 ±2.02	77.30 ±2.09	90.32±0.69
ラベルと次数の併用	88.16± 7.80	82.05±2.75	81.89±2.03	77.60±1.56	75.35 ±1.96	80.20 ±2.07	91.24±0.77
Graph2vec+Node2vec	85.00±6.67	80.02±2.27	77.33±2.42	75.68±1.73	73.23 ±2.00	77.01±2.43	91.00±0.74
Graph2vec+中心性	83.68±7.62	80.18±2.43	77.97±2.66	75.59±1.86	73.26 ±2.03	76.91±2.64	90.80±0.88

する．

グラフ分類タスクでは，各グラフの（特徴ベクトル，所属クラス）ペアを線形 SVM に学習させて分類器を作る．各データセット内から 90%のサンプルをランダムに選択し学習データとし，残りの 10%のサンプルをテストデータとする．学習データとテストデータを変えて分類実験を 20 回繰り返し，テストデータでの平均分類精度を報告する．SVM の正則化パラメータ C は $[0.01, 0.1, 1, 10, 100, 1000]$ の範囲から学習データに対する 5-fold cross validation を実施し最良パラメータを選択した．

5.3 実験結果

5.3.1 Graph2vec との比較

エッジラベルなしのデータセット 6 個に対する，従来手法 Graph2vec および提案手法 GL2vec のグラフ分類精度を**表 2** に示す．これらのデータセットのうち 5 つは [7] で Graph2vec の評価にすでに使用されているが，我々の実験では，Graph2vec は文献 [7] とほぼ同じ分類精度を再現した．

表 2 より，GL2vec が 6 個中，4 つのデータセット（MUTAG，NCI1，NCI109，および IMDB-B）で Graph2vec よりも分類精度が高くなった．

ここで，GL2vec と Graph2vec の平均分類精度に有意な差があるかを，20 回の分類実験の結果を対象とした対応のある t 検定により，5%の有意水準で検定した．p 値を表 2 に示す．GL2vec の方が平均分類精度が高かった 4 つのデータセットに関しては p 値が 5%を下回り，有意な差が確認できた．一方で，Graph2vec の方が平均分類精度が上回る PROTEINS，PTC については有意な差が確認できないという結果になった．以上の結果から，ノードラベルの影響を受けずにライングラフの特徴ベクトルで構造情報を補完する GL2vec のアプローチは決定的ではないものの有望であるといえる．

このなかで，IMDB-B はノードラベルを持たないデータセットであり，Graph2vec では元のグラフのノード次数をノードラベルに設定して，構造の類似性を評価している．これに対して，GL2vec では，4.2 節の最後で述べたように，元のグラフとライングラフの両者を用いて複数の解像度で構造の類似性を評価しており，単一レベルで構造的な類似性を評価する Graph2vec よりも高い分類精度を達成したと解釈できる．

PROTEINS に対して GL2vec が Graph2vec を上回れなかった理由は次のように考察される．PROTEINS ではノードラベルの種類数が 3 と非常に少ない．そして，個々のグラフの多くは，ノードラベルを 2 種類しか含まない．ノードラベルの種類数が少ないと，3 章で指摘したノードラベルとグラフ構造を同時に識別 ID に畳み込む Graph2vec の短所が緩和される．このため，GL2vec の戦略が効果を発揮できなかったと考えられる．

次にエッジラベルありのデータセット 7 個に対するグラフ分類精度を**表 3** に示す．7 個すべてのデータセットに対し，提案手法が従来手法より分類精度が上回った．さらに，GL2vec と Graph2vec とで平均分類精度に優位な差があるかを統計的に検定したところ，すべてのデータセットに対して，有意な差が確認できた．p 値を表 3 に示す．この結果は，Graph2vec ではエッジラベルを取り扱えないという欠点を，GL2vec がエッジラベルをライングラフの分散表現で補完することで克服できたことを示している．またこの結果は，エッジラベルはグラフ分類タスクに有用な情報であることも示している．

全体では，GL2vec は 13 個のデータセットのうち 11 個で Graph2vec よりもグラフ分類精度が上回り，ライングラフの分散表現がエッジ特徴を適切に補完できることを確認できた．

5.3.2 グラフとライングラフを両方用いることの効果

GL2vec ではライングラフ LG の分散表現を生成するが，あくまでも元グラフ G の分散表現を補完するためという位置づけであり，ライングラフの分散表現を単独で使用することは想定していない．実際，ライングラフの分散表現単体では，グラフ分類性能を向上できない．表 2，表 3 の 5 行目にライングラフの分散表現のみを使用した場合（LineGraph2vec）の分類精度を示す．LineGraph2vec は，13 個すべてのデータセットに対して分類制度が GL2vec より低く，MUTAG，MUTAG*，IMDB-B 以外の 10 個のデータセットに対しては Graph2vec よりも分類精度が低かった．このことから，グラフのエッジ特徴はノードラベルよりクラス識別能力への寄与が小さく，G の分散表現と LG の分散表現を併用する GL2vec の戦略は妥当であるといえる．

5.3.3 エッジラベルをライングラフで取り扱う効果

GL2vec は，グラフ G のエッジラベルをライングラフ上で WL リラベリングする．しかし，グラフ G のエッジラベルを G に対する WL リラベリングに組み込むことは不可能ではない [11]．具体的には，ノード $v \in V$ に対する根付き部分グラフを作るときに，（ソートされた）隣接ノードラベルの多重集合 $M^t(v) = \{\lambda^{t-1}(u) | u \in \mathrm{Neighbors}(v)\}$ に（ソートされた）隣接エッジラベルの多重集合 $MEL(v)$ を連結し，その後で根ノードのラベル $\lambda^{t-1}(v)$ を先頭に連結すればよい．つまり，根付き部分グラフは

$$\lambda^{t-1}(v)\text{-}[M^t(v), MEL(v)]$$

という形式になる．G における WL リラベリングをこのように変更した Graph2vec を Graph2vec_edge と名付ける．Graph2vec_edge は，GL2vec と以下の 2 点が異なる．

(1) エッジラベルをリラベリングで更新できない．WL リラベリングの対象となるのは，根付き部分グラフの根であるノードラベルだけである．
(2) 2 つの部分グラフは，ノードラベルとエッジラベルの両方が一致してはじめて同一であると判断される．GL2vec では元グラフではノードラベル，ライングラフでは（G での）エッジラベルが一致すれば，2 つの部分グラフを同一と判断する．

Graph2vec_edge をエッジラベルありデータセットに適用したときの分類正解率を表 3 に示す．Graph2vec_edge の分類正解率はほぼ Graph2vec と同じであり，GL2vec を下回った．この結果は，GL2vec がエッジラベルをノードラベルから分離する方式の有効性を示す．

5.3.4 エッジ次数とエッジラベルの併用

エッジラベルありデータセットに対して，GL2vec では G のエッジラベルを LG のノードラベルに設定するが，その代わりに G のエッジ次数を LG のノードラベルに設定することも可能である．たとえば，G のエッジラベルをノードラベルとするライングラフ LG_1 および G のエッジ次数をノードラベルとするライングラフ LG_2 を作成し，3 個の特徴ベクトル $\vec{G}, L\vec{G}_1, L\vec{G}_2$ を結合した 3δ 次元の特徴ベクトルを生成することも可能である．このアプローチは有望だが，5.3.1 項でエッジ次数を使用した場合に分類精度が Graph2vec を下回るデータセットがあったことから提案手法には含めていない．また，連結される特徴ベクトル数が増えるに連れ，特徴ベクトル間の重み付けにも配慮が必要になる．なお，エッジラベルありデータセットに対して，上記のやり方で 3δ 次元の特徴ベクトルを生成した場合の分類精度を表 3 の「ラベルと次数の併用」の行に示す．7 個のデータセット中，MUTAG*，DBLP 以外の 5 データセットで分類精度が向上し，エッジ次数特徴が有望であることが再確認できた．

5.3.5 他の特徴量との比較

GL2vec では，Graph2vec による分散表現 $\vec{G}_i$ にライングラフで生成されたエッジ特徴の分散表現 $L\vec{G}_i$ を連結する．本節では，$\vec{G}_i$ に別の特徴量を連結する手法と GL2vec との分類性能を比較し，エッジラベル特徴の有用性を考察する．

別の特徴量としては，

(1) Node2vec [3] を適用して得られた G_i の各ノードに対する分散表現の平均ベクトル ave_node(G_i)
(2) 近接中心性ヒストグラム

の 2 種類を試した．(1) については，ave_node(G_i) の次元数を $L\vec{G}_i$ に揃えて 1,024 とした．(2) の近接中心性はノードがグラフのどれだけ中心にあるかを表す特徴量で，グラフ $G_i = (V, E)$ のノード $v \in V$ の近接中心性 $cen(v)$ は $cen(v) = \dfrac{n-1}{\sum_{v' \in V-v} d(v, v')}$ と定義される．$d(v, v')$ は v, v' 間のホップ数であり，n は G_i のノード数である．$cen(v)$ の値域は $0 \leq cen(v) \leq 1$ であり，1 に近いほど中心性が高い．近接中心性を G_i の全ノードに対して算出し，0.05 きざみで 20 次元のヒストグラム HG(G_i) を生成した．

$\vec{G}_i$ に ave_node(G_i) を連結する手法および，$\vec{G}_i$ に HG(G_i) を連結する手法のエッジラベルありデータセットに対する分類精度を表 3 の「Graph2vec+Node2vec」と「Graph2vec+中心性」の行にそれぞれ示す．MUTAG*で「Graph2vec+中心性」が Graph2vec の分類精度を下回ったことを除けば，これらの手法は Graph2vec より高い分類精度を達成した．一方でこれらの手法は全データセットに対して，分類精度が GL2vec より低い．このことよりグラフ分類におけるエッジラベル特徴の有用性を確認できた．

6. 関連研究

6.1 ライングラフの利用例

まず，グラフベースパターン認識でライングラフを用いた関連手法を紹介する．Bai ら [17] は，マッチしたエッジペア数から 2 つのグラフ間の類似性を測定するエッジベースのグラフカーネル EMBK を提案した．2 つのエッジがマッチするかを判断するために，ライングラフを用いて元グラフ G のエッジに対する特徴ベクトルを生成する．Bai らは，エッジベースのマッチングカーネルが，元グラフから定義されるノードベースのマッチングカーネルよりも優れていることを示した．本研究の GL2vec と EMBK の違いは下記の 2 点である．(1) EBMK はグラフの特徴ベクトルを直接には生成しない．一方，GL2vec は任意のベクトルベースの機械学習アルゴリズムに適用可能なグラフの特徴ベクトルを生成する．(2) EBMK は元のグラフを無視してライングラフのみ使用するのに対し，GL2vec では元のグラフとライングラフの両方を利用してグラフの分散表現を生成する．DPGCNN [18] は，GAT（Graph Attention Networks）[19] を拡張したグラフ畳み込みネットワークである．GAT はノード特徴からアテンションスコアを計算するが，DPGCNN はエッジ特徴からアテンションスコアを計算する．GL2vec と同様に，DPGCNN もグラフのエッジ特徴をライングラフのノード特徴に変換する．DPGCNN は end-to-end モデルでありノード分類やリンク推定の目的でノードやエッジなどの特徴量を教示ありで学習するのに対して，GL2vec では非教示でグラフ分類に有用なグラフ全体に対する特徴ベクトルを獲得する．

6.2 非教示型の分散表現生成

ここでは，非教示でグラフ全体に対して分散表現を生成する関連研究を記述する．この分野はここ 1，2 年で急激に発展しており，非教示型の分散表現学習手法のライブラリを提供する試みも始まっている [20]．

伝統的にはグラフ分類ではグラフカーネル [11], [21] が使われて来た．グラフカーネルは，SVM のような特定の分類器と組み合わされることを前提とし，個々のグラフに対する分散表現を直接生成しないものも多い．分散表現を生成する手法も，分散表現が高次元疎ベクトルになりがちである．

ニューラルネットに基づく手法は，グラフカーネルよりも低次元で密な分散表現を生成できる．GE-FSG（Graph Embedding based on Frequent SubGraphs）[8] は本研究と同様にエッジラベルを取り扱える手法である．GE-FSG ではまずグラフデータセットの頻出部分グラフを求める．そして，個々のグラフ G_i を内包する頻出部分グラフの集合として表現してから分散表現を生成する．この過程でエッジラベルありで頻出部分グラフを発見すれば，分散表現にエッジラベル情報が反映される，GE-FSG はグラフ分類性能が Graph2vec を上回ると報告されている [8]．GE-FSG は，GL2vec と較べると (1) 根付き木以外の部分グラフも生成でき，(2) 出現回数の少ないノイズを排除できる点が優れている．一方で，頻出部分グラフを 1 つも含まないグラフに対しては分散表現を生成できないので，データセットの全グラフに対して分散表現を生成できるとは限らないという欠点がある．また，部分グラフの同型性判定の基準も異なり，GE-FSG では（グラフ形状に加えて）エッジラベルとノードラベルの両者が一致しないと同型と判定されないが，GL2vec では片方が一致すれば G_i と LG_i のどちらかで同型と判定される．文献 [22] では，レシピフローグラフを対象に頻出部分グラフの代表元を選択することを提案している，しかし，代表元の選択にドメイン知識を用いている．また，頻出部分グラフ数を減らしているため，分散表現を生成できないグラフが増えることが懸念される．

Infograph [23] は，G_i の部分グラフに対する分散表現と G_i 自体に対する分散表現の相互情報量を最大化するように学習することで，Graph2vec よりグラフ分類の精度が高い分散表現を生成する．また UGRAPHEMB（Unsupervised Graph-level Embedding）[9] は，グラフ間の編集距離を分散表現に反映させ，グラフ編集距離を類似度とするクラスタリングや類似検索と同様の出力結果を出せる分散表現を生成する．SEED（Sampling，Encoding and Embedding Distributions）[24] では，まず WEAVE と呼ばれるランダムウォークの一種を用いて G_i から部分構造をサンプリングし，オートエンコーダで部分構造に対する分散表現を生成する．そして，部分構造に対する分散表現から G_i に対する分散表現を出力する多層パーセプトロンを学習している．UGRAPHEMB と SEED は GL2vec と異なり inductive な手法であり，分散表現の学習に使われなかった新たなグラフに対しても分散表現を生成できる点が優れる．GL2vec はエッジ特徴を用いた分散表現の改良を目的としており，これら 3 手法とは改良の方向性が異なる．UGRAPHEMB と SEED ではエッジラベルを考慮しない，Infograph は，エッジラベルを分散表現に反映できるが，リラベリングで更新しない．

文献 [25] はトポロジー情報を考慮した WL リラベリングの拡張である P-WL（Persistent WL）を提案している．この手法では，グラフ $G_i = (V, E)$ を，エッジが 1 つもなく V のみからなる初期状態から，エッジを 1 本ずつ追加して G_i を復元するプロセスを考える．このとき，追加されるエッジ e が異なる連結成分を連結するか，同一連結成分内でサイクルを作るかに着目し，e の連結成分度，サイクル度を算出する．そして，ノード v の連結成分度，サイクル度を隣接エッジ集合の連結成分度，サイクル度から定める．WL リラベリングにおいて識別 ID の多重集合を作る際には，ノードの連結成分度，サイクル度で多重度を重み

付けし，グラフの構造情報を多重度に反映する．P-WL ではグラフの構造情報で多重度を重み付けするだけなので 3 章で指摘したノードラベルと構造情報を同時に識別 ID に畳み込むという WL の性質を継承している．GL2vec は，エッジラベルなしグラフ G_i に対して，G_i のノード情報をまったく持たないライングラフ LG_i を生成し，G_i のノードラベルに依存しない構造情報のみで定まる識別 ID の多重集合を生成する点が異なっている．

GL2vec の大きな特色は，原理的には任意の非教示な分散表現学習手法をサブルーチンとして呼び出せるメタなフレームワークであることである．たとえば，Graph2vec の代わりに GE-FSG をサブルーチンとして呼び出すことも可能である．このようにして，非教示な分散表現学習の進化を取り込むことが可能である．実際に，GE-FSG をサブルーチンとして呼び出した場合の実験評価を 7 章で後述する．

6.3 エッジ特徴の利用

ここでは，近年のエッジ特徴を用いたグラフニューラルネットワークについて述べる．GCN [26] は end-to-end モデルで教示ありでノードの分散表現を獲得する．GCN は周辺ノードの情報をメッセージパッシングで集約（aggregation）することを繰り返し，学習を進める．この際，エッジの重みに従って集約への貢献度を変える．つまり，GCN では接続行列の要素を {0,1} の 2 値ではなく重み付けするためエッジ特徴が存在し，本研究のように明示的に分散表現に反映させる対象ではない．文献 [27] は GCN のエッジ重みを多次元，つまり複数種類に拡張している．文献 [28] では，論文の共著者ネットワークなどノードとエッジの両者にテキスト特徴が付与されているグラフを想定し，ノードの分散表現を獲得する GERI を提案した．GERI では，元のグラフ G に単語ノードを追加した heterogeneous なグラフ HG を構築し，HG 上でのランダムウォークによりノードの分散表現を獲得する．G におけるノード特徴とエッジ特徴は，単語ノードと G のノード間にエッジを張るかを決定するために使われる．GERI はノードの分散表現を生成する手法であり，グラフ全体に対して分散表現を生成しようとすると余分な単語ノードが悪影響を及ぼす可能性が懸念される．文献 [29] では SNS を対象にオートエンコーダを利用してノードの分散表現を生成した．損失関数にエッジラベルの不一致度を表す項を加え，エッジラベルを分散表現に反映している．ただし，グラフ全体に対する分散表現は取り扱っていない．文献 [30] では，ノードではなくエッジを主体とするメッセージパッシングを行う EMNN（Edge Memory Neural Network）を提案している．GL2vec もライングラフ側でエッジを主体とするメッセージパッシングを実行している．しかし，EMNN では最初にエッジ特徴と両端ノードのノード特徴を結合している．GL2vec ではノード特徴とエッジ特徴を分離してメッセージパッシングを行う点が EMNN と異なる．

7. 他の非教示型分散表現学習手法への適用

グラフの分散表現とライングラフの分散表現を連結するという GL2vec のアプローチは，任意の非教示な分散表現学習手法をサブルーチンとして呼び出すことが可能であり，汎用性に優れている．そこで，Graph2vec とは別の GE-FSG [8] への適用を試みた．ライングラフを併用する GE-FSG アルゴリズムを GE-FSG-LG と名付ける，GE-FSG-LG の処理手順を以下に示す．

(1) グラフ集合 $\{G_1, G_2, \ldots, G_N\}$ に対し，ライングラフの集合 $\{LG_1, LG_2, \ldots, LG_N\}$ を作成する．LG_i のノードラベルは，GL2vec と同様に設定する．
(2) $\{G_1, G_2, \ldots, G_N\}$ に対して支持度 θ_1 以上の頻出部分グラフを列挙し，各 G_i を包含する頻出部分グラフの集合として表す．その後 GE-FSG を適用し，G_i に対する δ 次元の特徴ベクトル $\vec{G_i}$ を得る．
(3) $\{LG_1, LG_2, \ldots, LG_N\}$ に対して対して支持度 θ_2 以上の頻出部分グラフを列挙し，各 LG_i を頻出部分グラフの集合として表す．そして GE-FSG により，LG_i に対する δ 次元の特徴ベクトル $L\vec{G}_i$ を得る．
(4) G_i に対して，$\vec{G_i}$ と $L\vec{G}_i$ を結合した 2δ 次元のベクトルを G_i に対する最終的な特徴ベクトルとする．

5.1 節に示した 13 個のデータセットに，オリジナルの GE-FSG および GE-FSG-LG を適用しグラフ分類精度を評価する実験を行った．GE-FSG のコードは，GE-FSG の開発者が公開[*1]しているものを使用した．次元数は文献 [8] と揃えて $\delta = 128$ とした．また，ライングラフに対する最小指示度は $\theta_2 = 0.5$ に固定した．元グラフに対する最小指示度 θ_1 はデータセットによって異なるが，文献 [8] で取り扱われているデータセットに関しては NCI1 以外は文献 [8] と同じ値を使用した．[8] では NCI1 に対して $\theta_1 = 0.2$ とするが，この値だと頻出部分グラフをまったく包含しないグラフが出現したため $\theta_1 = 0.15$ に下げた．

表 4, **表 5** に分類精度をまとめる．PROTEINS, IMDB-B についてはライングラフに対して，DBLP については元グラフとライングラフの両者に対してバイナリで提供されている頻出部分グラフ抽出アルゴリズムが短時間で完了せず，グラフ分類を実施できなかった．残りの 10 個のデータセットすべてに対して，GE-FSG-LG が GE-FSG の分類精度を上回った．特にエッジラベルありデータセットに対しては，NCI83，NCI123，NCI145 で GE-FSG-LG が GE-FSG の分類精度を約 20%上回るなど性能差が顕著であった．

GE-FSG はエッジラベルありデータセットに対しては，

[*1] https://github.com/nphdang/GE-FSG

表 4 GE-FSG：エッジラベルなしデータセットに対する分類正解率
Table 4 GE-FSG: Classification accuracy for datasets without edge labels.

データセット名	MUTAG	PTC	PROTEINS	NCI1	NCI109	IMDB-B
GE-FSG-LG	**83.16**±7.0	**60.29**±6.7	N/A	**78.59**±2.3	**76.15**±1.6	N/A
GE-FSG	81.05±5.4	56.86±9.9	71.43±3.1	75.13±1.3	72.57±2.2	68.00±3.5
θ_1	0.35	0.15	0.55	0.15	0.2	0.15

表 5 GE-FSG：エッジラベルありデータセットに対する分類正解率
Table 5 GE-FSG: Classification accuracy for datasets with edge labels.

データセット名	MUTAG*	NCI33	NCI81	NCI83	NCI123	NCI145
GE-FSG-LG	**81.58**±7.2	**75.96**±1.9	**85.56**±2.3	**87.65**±1.1	**92.47**±0.9	**83.1**±2.2
GE-FSG	75.79±5.4	69.79±1.4	65.01±2.7	66.80±2.6	69.35±1.8	69.56±1.5
θ_1	0.35	0.2	0.2	0.2	0.1	0.2

エッジラベル付き頻出部分グラフを生成する．つまり，エッジラベルを利用する．GE-FSG と GE-FSG-LG との大きな違いは，グラフ同型の判定方法である．GE-FSG ではグラフ形状に加えてノードラベルとエッジラベルの両者が一致する必要があるが，GE-FSG-LG のラングラフ側ではグラフ形状とエッジラベルが一致するだけで同型と判定する．この実験結果から，エッジラベルをノードラベルと独立に評価するというアプローチは化合物分析で有効といえる．

また，PROTEINS 以外のデータセットでは，GL2vec が GE-FSG を上回るグラフ分類性能を達成した．これは，GL2vec が GE-FSG-LG と同様にエッジラベルをノードラベルと分離しているためと考えられる．一方で，表 5 に拠ると，GE-FSG-LG は GL2vec を上回る分類精度を達成する．これは，頻出でない部分グラフは考慮しないという GE-FSG のアイデアが妥当であることを示す．実際，データセットで一度しか出現しない根付き部分グラフを使用しないことで Graph2vec の性能を改善できるという研究報告もあり，GL2vec でもそのような手段を採用すべきと考えられる．

PTC については，GL2vec が Graph2vec より分類精度が低かったにもかかわらず，GE-FSG-LG は GE-FSG より分類精度が高かった．このことから，頻出でない部分グラフを考慮することの影響で，GL2vec が Graph2vec を下回った可能性が考えられる．

本章により，グラフの分散表現とラングラフの分散表現を連結する GL2vec のアプローチが複数の非教示型の分散表現学習手法でうまく機能することを示せた．

8. まとめ

本論文では Graph2vec を拡張した GL2vec（Graph and Line graph to vector）という手法を提案した．本論文ではまず，Graph2vec の限界を 2 つ指摘した．1 つ目はエッジラベルを扱えないことであり，2 つ目は抽出した根付き部分グラフを識別 ID へマッピングする際に，ノードラベルと構造情報を同時に畳み込むため，構造の類似性を適切に判定することが困難な場合があることである．提案手法 GL2vec では，Graph2vec では考慮し難いグラフ G のエッジラベルや構造情報をエッジ特徴として表現後，ラングラフ LG のノード特徴に変換してから LG の分散表現に畳み込む．そして，G の分散表現に LG の分散表現を連結し，グラフ G のエッジラベルや構造情報を LG の分散表現により補完する．実験では，多くのベンチマークデータセットに対し，GL2vec は Graph2vec と較べてグラフ分類タスクの精度を改善した．

今後の研究課題を述べる．まず，GL2vec がどのようなデータセットに対して最も有効に動作するかを明らかにしたい．また，ラングラフは元グラフよりノード数・エッジ数が大きく，処理コストが大きいという課題がある．ラングラフをよりサイズの小さいグラフで近似することも研究の方向性として価値がある．

謝辞 本研究は JSPS 科研費 JP18K11311 の助成を受けたものである．

参考文献

[1] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality, *Advances in Neural Information Processing Systems 26*, pp.3111–3119 (2013).
[2] Le, Q. and Mikolov, T.: Distributed Representations of Sentences and Documents, *Proc. 31st International Conference on Machine Learning*, PMLR, Vol.32, No.2, pp.1188–1196 (2014).
[3] Grover, A. and Leskovec, J.: node2vec: Scalable Feature Learning for Networks, *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.855–864 (2016).
[4] Niepert, M., Ahmed, M. and Kutzkov, K.: Learning Convolutional Neural Networks for Graphs, *Proc. ICML 2016*, pp.2014–2023 (2016).
[5] Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W. and Leskovec, J.: Hierarchical Graph Representation Learning with Differentiable Pooling, *Proc. NIPS 2018*, pp.4805–4815 (2018).

[6] Xu, K., Hu, W., Leskovec, J. and Jegelka, S.: How Powerful are Graph Neural Networks?, *Proc. ICLR 2019* (2019).
[7] Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y. and Jaiswal, S.: graph2vec: Learning Distributed Representations of Graphs, *Proc. 13th International Workshop on Mining and Learning with Graphs* (2017).
[8] Nguyen, D., Luo, Y., Nguyen, T.D., Venkatesh, S. and Phung, D.: Learning Graph Representation via Frequent Subgraphs, *Proc. SIAM International Conference on Data Mining* (2018).
[9] Bai, Y., Ding, H., Qiao, Y., Marinovic, A., Gu, K., Chen, T., Sun, Y. and Wang, W.: Unsupervised Inductive Graph-Level Representation Learning via Graph-Graph Proximity, *Proc. 28th International Joint Conference on Artificial Intelligence*, pp.1988–1994 (2019).
[10] Chen, H. and Koga, H.: GL2vec: Graph Embedding Enriched by Line graphs with Edge Features, *Proc. 26th International Conference on Neural Information Processing*, pp.3–14 (2019).
[11] Shervashidze, N., Schweitzer, P., van Leeuwen, E., Mehlhorn, K. and Borgwardt, K.: Weisfeiler–Lehman Graph Kernels, *Journal of Machine Learning Research*, Vol.12, pp.2539–2561 (2011).
[12] Harary, F.: *Graph Theory*, Addison–Wesley, pp.71–83 (1972).
[13] Whitney, H.: Congruent Graphs and the Connectivity of Graphs, Hassler Whitney Collected Papers, Birkhäuser Boston, pp.61–79 (1992).
[14] Benchmark Data Sets for Graph Kernels, available from ⟨http://graphkernels.cs.tu-dortmund.de⟩
[15] Yanardag, P. and Vishwanathan, S.V.N.: Deep Graph Kernels, *Proc. 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1365–1374 (2015).
[16] Zhu, X., Zhang, C., Pan, S. and Yu, P.: Graph Stream Classification using Labeled and Unlabeled Graphs, *Proc. IEEE 29th International Conference on Data Engineering*, pp.398–409 (2013).
[17] Bai, L., Zhang, Z., Wang, C. and Hancock, E.R.: An Edge-based Matching Kernel for Graphs through the Directed Line Graphs, *Proc. International Conference on Computer Analysis of Images and Patterns*, pp.85–95 (2015).
[18] Monti, F., Shchur, O., Bojchevski, A., Litany, O., Günnemann, S. and Bronstein, M.M.: Dual-primal Graph Convolutional Networks, arXiv preprint arXiv: 1806.00770 (2018).
[19] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y.: Graph Attention Networks, *Proc. 6th International Conference on Learning Representations* (2018).
[20] Scherer, P. and Lio, P.: Learning distributed representations of graphs with Geo2DR, *Proc. ICML 2020 Workshops in Graph Representation Learning and Beyond* (2020).
[21] Borgwardt, K.M. and Kriegel, H.: Shortest-path Kernels on Graphs, *Proc. ICDM 2005*, pp.74–81 (2005).
[22] Ninomiya, A. and Ozaki, T.: Learning Distributed Representation of Recipe Flow Graphs via Frequent Subgraphs, *Proc. 11th Workshop on Multimedia for Cooking and Eating Activities*, pp.25–28 (2019).
[23] Sun, F., Hoffman, J., Verma, V. and Tang, J.: InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization, *Proc. ICLR 2020* (2020).
[24] Wang, L., Zong, B., Ma, Q., Cheng, W., Ni, J., Yu, W., Liu, Y., Song, D., Chen, H. and Fu, Y.: Inductive and Unsupervised Representation Learning on Graph Structured Object, *Proc. ICLR 2020* (2020).
[25] Rieck, B., Bock, C. and Borgwardt, K.: A Persistent Weisfeiler–Lehman Procedure for Graph Classification, *Proc. 36th International Conference on Machine Learning*, pp.5448–5458 (2019).
[26] Kipf, T.N. and Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks, *Proc. International Conference on Learning Representations* (2017).
[27] Gong, L. and Cheng, Q.: Exploiting Edge Features for Graph Neural Networks, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.9211–9219 (2019).
[28] Sun, G. and Zhang X.: A Novel Framework for Node/Edge Attributed Graph Embedding, *Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining* (*PAKDD 2019*), pp.160–182 (2019).
[29] Goyal, P., Hosseinmardi, H., Ferrara, E. and Galstyan, A.: Capturing Edge Attributes via Network Embedding, *IEEE Trans. Computational Social Systems*, Vol.5, No.4, pp.907–917 (2018).
[30] Withnall, M., Lindelëf, E., Engkvist, O. and Chen, H.: Building Attention and Edge Message Passing Neural Networks for Bioactivity and Physical-Chemical Property Prediction, *Journal of Cheminformatics*, Vol.12, No.1 (2020).

陳 宏

1993年生．2016年（中国）東北大学秦皇島分校通信工学専攻卒業．2020年電気通信大学大学院情報・ネットワーク工学専攻修士課程修了．在学中はグラフニューラルネットワークの研究に従事．同年富士通クラウドテノロジーズ入社．

古賀 久志 （正会員）

1971年生．1995年東京大学大学院理学系研究科情報科学専攻修了．1995年富士通研究所入社．博士（理学）．2003年電気通信大学大学院情報システム学研究科講師，現在，電気通信大学大学院情報理工学研究科准教授．専門は類似検索アルゴリズム．現在はストリームデータ処理の研究に従事．