F-020

双対グラフを用いたグラフの分散表現学習
Learning Distributed Representations of primal-dual Graphs

Chen Hong† 古賀 久志‡
Hong Chen Hisashi Koga

1. はじめに

グラフ構造は、複雑なオブジェクトを表現するための強力なツールであり、ソーシャルネットワーク分析、生物情報科学、化学情報科学などの多くの分野で広く使用されている．近年、グラフデータを既存の機械学習アルゴリズムに適用するために、グラフデータに対して数値特徴ベクトル（分散表現とも呼ばれる）を学習する技術が注目されている．これらの技術の多くは、自然言語処理における word2vec [1]のような埋め込みアルゴリズムをグラフ向けに拡張したものであり、多くの言語モデル（skip-gram [1]）ベースのアルゴリズムが提案されている．代表的な手法は node2vec [3]と graph2vec [4]である．node2vec はノードに対する分散表現を獲得する手法で、ノード分類，リンク予測などのタスクで利用される．

一方，graph2vec はノードラベル付きのグラフを対象に，グラフ全体に対する分散表現を獲得する手法で、グラフ分類、グラフクラスタリングなどのタスクに適用される．とくに，グラフ集合を与えられて非教示で個々のグラフに対する特徴ベクトルを得る．graph2vec は自然言語処理における doc2vec[2]を基にしている．doc2vec ではドキュメントを単語集合として表すのに対し，graph2vec では 1 つのグラフを局所的な部分グラフの集合として表す．

本研究では，graph2evc を改良する手法を提案する．本稿ではまず graph2vec の以下の 2 つの課題を指摘する：(1)部分グラフを量子化する際にノードラベルとグラフ構造(structure)を同時に畳み込むため，構造の類似性を適切に表せない場合がある．また，(2)エッジラベルを扱えない．

本研究では，この課題に対処するため，元のグラフ G のノード双対グラフ LG(line graph)を利用することを提案する．line graph では G のエッジが LG のノードにマップされるので，G のエッジラベルを LG のノードラベルとして持たせられる．また，LG は G のノードラベルを持たないので構造情報を G のノードラベルと独立に表現できる．とくに提案手法では，G に対する分散表現と LG に対するに分散表現を結合し，ノードラベルと，構造情報またはエッジラベルを反映したベクトル表現を生成する．実験により，グラフ分類タスクにおける多数のベンチマークデータセットに対して，提案手法が graph2vec よりも分類精度を向上できることを示す．

本論文の残りの部分は次のように構成されている．2 章では、グラフ分散表現、skip-gram モデル、および graph2vec の背景知識を紹介する．3 章では line graph の定義と提案手法について説明する．4 章は、実験結果を述べ、5 章で結論を述べる．

†‡電気通信大学大学院 情報理工学研究科
Graduate School of Informatics and Engineering,
The university of electro-communications

2. 背景知識

2.1 グラフ分散表現

$G = \{V, E, \lambda_v, \lambda_e\}$ をラベル付き無向グラフを表すものとする．ここで、V はノードの集合であり、$E \subseteq (V \times V)$ はエッジの集合である． λ_v は、関数 $\lambda_v: V \to \mathcal{L}$ で、アルファベット$\mathcal{L}$ からすべてのノード$v \in V$ に一意のラベルを割り当てる．また、λ_e は$\lambda_e: E \to \ell$ の関数で、アルファベットℓ からすべてのエッジ $e \in E$ に一意のラベルを割り当てる．

N 個のグラフの集合 $\mathbf{G} = \{G_1, G_2, \ldots G_N\}$ を考えると、グラフ分散表現学習の手法はニューラルネットワークを用いて $\mathbf{G}$ から特徴ベクトルの集合$f(\mathbf{G})$への写像を学習する．ここでは、$f(\mathbf{G}) = \{f(G_1)^\delta, f(G_2)^\delta, \ldots f(G_N)^\delta\}$ 、 δ はベクトルの次元数である．注目に値するのは、優れた表現学習方法には、構造情報、ノード属性、エッジ属性などのグラフのプロパティを取得する必要があるということである．

2.2 Skip-gram モデル

Skip-gram [1]モデルは、単一の隠れ層を持つニューラルネットワークアーキテクチャである．分散表現を学習したいターゲットオブジェクトが one-hot ベクトルの形でエンコードされて、入力層の入力とする．そして、隠れ層における各ニューロンの状態が入力に対応する分散表現を表す．出力層の出力は、コンテキストオブジェクトに対し入力を条件とした予測確率分布である．モデルは、ウィンドウによって選択されるコンテキストにおける予測されたオブジェクトの尤度を最大化することを目的として、学習を進む．モデルが収束した後、類似な目標オブジェクトはベクトル空間における類似な位置にマッピングされる．

2.3 Graph2vec

Graph2vec [4] はノードラベル付きのグラフ集合$\mathbf{G} = \{G_1, G_2, \ldots G_N\}$に対して非教示でグラフの分散表現を学習する手法である．

Graph2vec では，まず個々のグラフ G_i を根付き部分グラフの集合と表現する．事前に根付き部分グラフの最大深さ H を設定して、各ノードを根として、Weisfeiler-Lehman [5] 再ラベル操作の繰り返しにより、深さ h が 0 から H までの根付き部分グラフを作る．特に、Weisfeiler-Lehman 再ラベル操作で圧縮したラベルを根付き部分グラフとして使われる．

t 回目の Weisfeiler-Lehman 再ラベル操作の手順を以下に示す．

(step 1) ：$\forall v \in V$ に対し、近傍ノードラベルの多重集合 $\mathrm{Multiset}^t(v) = \{\lambda_n{}^{t-1}(u) | u \in \mathrm{Neighbors}(v)\}$ を作る．

(step 2) ： $\mathrm{Multiset}^t(v)$ の要素を昇順でソートした文字列$\mathrm{String}^t(v)$を作る．その後、$\mathrm{String}^t(v)$の先頭に根ノードのラベル$\lambda_n{}^{t-1}(v)$を加える．

(step 3)　：ハッシュ関数 H を用いて $\mathrm{String}^t(v)$ をラベルに $\mathrm{H}(\mathrm{String}^t(v))$ に変換する．このハッシュ関数 H は異なる文字列を異なるラベルに写像することが要求されるが，基数ソートを使用すれば簡単に実装できる．

(step 4)　：$\mathrm{H}(\mathrm{String}^t(v))$ をノード v の新しいラベル ${\lambda_n}^t(v)$ とする．

大雑把には ${\lambda_n}^t(v)$ はノード v を根とする深さ(ホップ数) t の局所部分グラフの識別 ID を表す．したがって，Step 2 では，v の隣接ノードの深さ t-1 の根有部分グラフと v 自身の深さ t-1 の根有部分グラフから，v の深さ t の根有部分グラフを合成し，Step 3 で根有部分グラフを量子化している．Weisfeiler-Lehman 再ラベル操作の実行例を図 1 に示す．

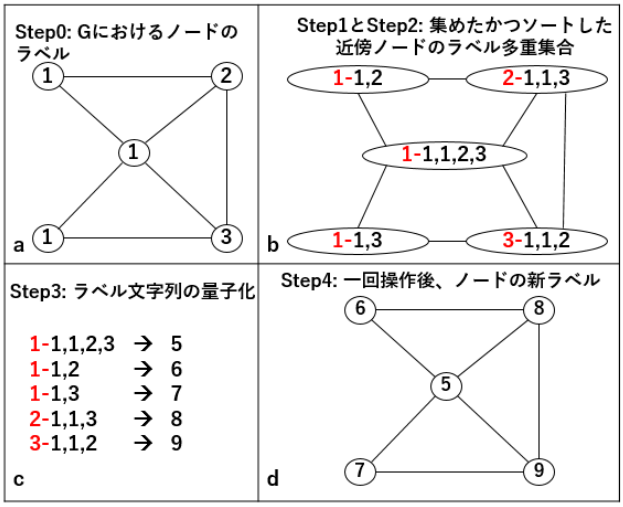

図 1　グラフ G に対する 1 回 W-L 再ラベル操作

Graph2vec では部分グラフから構成されたグラフに skip-gram モデルを適用する．グラフの集合 $\mathbf{G} = \{G_1, G_2, \dots G_N\}$ とグラフ G_i からサンプリングされた根付き部分グラフのコンテキスト $\mathrm{c}(G_i) = \{sg_1, sg_2, \dots sg_{l_i}\}$ に対して、G_i と sg_j の δ 次元の分散表現($\overrightarrow{G_i} \in \boldsymbol{R}^{\delta}$ と $\overrightarrow{sg_j} \in \boldsymbol{R}^{\delta}$)を学習する．学習モデルには、$sg_j$ が G_i からサンプリングされたことに基づき、下記のような対数尤度関数の最大化に目指す：

$$\sum_{j=1}^{l_i} \log P_r(sg_j|G_i) \qquad (1)$$

ここで、条件付き確率 $P_r(sg_j|G_i)$ が下記のように定義する：

$$\frac{\exp(\overrightarrow{G_i} \cdot \overrightarrow{sg_j})}{\sum_{sg \in Voc} \exp(\overrightarrow{G_i} \cdot \overrightarrow{sg})} \qquad (2)$$

ここで、Voc は $\mathbf{G}$ のすべてのグラフにわたるすべての部分グラフの集合である．なお、モデルには、ネガティブサンプリング [1] を使用して効率的に学習できる．そして、モデルが収束した後、類似な構造（根付き部分グラフ）を持つグラフ同士が、ベクトル空間における類似な位置にマッピングされる．

3. 提案手法

本章では，まず 2.3 章で述べた graph2vec が持つ欠点を 2 つ説明し，その課題を克服する手法を提案する．

graph2vec の 1 つ目の欠点はエッジにラベルが付与されていても，それを利用できないことである．これは各ノードを根とする部分グラフを抽出する時点でエッジラベルを使用しないことから明らかである．

2 つ目の欠点は，抽出した根付き部分グラフを量子化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，構造の類似性を適切に表現でき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．さて，graph2vec における根有部分グラフの識別 ID への量子化はノードラベルとグラフの構造を同時に畳み込んでいる．しかし，この方式ではノードラベルが一致している条件でのみ構造の同一性を評価するので，2 個のグラフの構造が類似していることが判別できない場合がある．この事を図 2 の例で示す．図 2 は形状が同一で中央ノードのラベルだけが異なる 2 つのスターグラフ G,G*である．

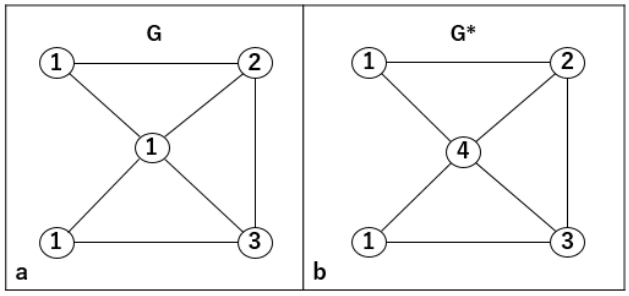

図 2　形状が同一で中央ノードのラベルだけが異なる 2 つのグラフ G,G*

G,G*に対する深さが 1 以下の量子化した根付き部分グラフの多重集合 c(G1),c(G2)は以下のようになる．

c(G)={1,1,1,2,3,5(1-1,1,2,3),6(1-1,2),7(1-1,3),8(2-1,1,3),9(3-1,1,2)}

c(G*)={1,1,2,3,4,10(1-2,4),11(1-3,4),12(2-1,3,4),13(3-1,2,4),14(4-1,1,2,3)}

c(G1)と c(G2)からは「G1 と G2 はノードラベルが一致するノード[1,1,2,2]を 4 つ持つ」ことしか判別できない．G1 と G2 の形状が同一であることはおろか，ノードラベルが一致するノードの次数が一緒であることすら判別困難である．これは，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，形状が同一であるという情報は捨てられるためである．

提案手法では，graph2vec で表現するのが困難な(1)エッジラベルと(2)グラフの構造情報を，元のグラフに対するラインングラフの分散表現で補う．以下 3.1 章でラインングラフを説明し，3.2 章で提案アルゴリズムを記述する．

3.1 ラインングラフ(edge-to-vertex dual)

グラフ $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ に対応するラインングラフ $\mathrm{LG} = (\mathrm{LV}, \mathrm{LE})$ とは，G のエッジの隣接関係を表すグラフである．G の各エッジが LG のノードになる．つまり，$\mathrm{LV} = \{v(e) | e \in E\}$．また．エッジ集合 LE は以下のルールで構築される．

$\mathrm{LE} = \{((v(e_i), v(e_j)) | e_i$ と e_j が G において端点を共有する$\}$

LG のノードの $v(e)$ の次数は，G のエッジ e の端点の次数の合計を用いて以下のように表現できることが知られている．

$\deg(v(e)) = \deg(v_a) + \deg(v_b) - 2$，　ただし，$e = (v_a, v_b)$

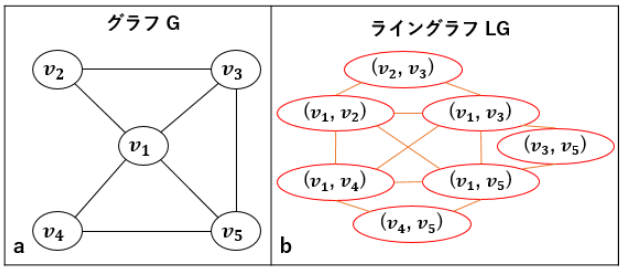

図 3　グラフと対応するラングラフ．

図 3 にラングラフの例を示す．ここでは，G のエッジ(v_1, v_2)とエッジ(v_1, v_4)が端点 1 を共有するので、LG でノード(v_1, v_2)とノード(v_1, v_2)が接続される

3.2 ラングラフを利用したグラフ分散表現の強化

graph2vec で表現するのが困難なグラフ G の(1)エッジラベルと(2)構造情報を，G に対するラングラフ LG を利用して補う手法を提案する．提案手法を PDgraph2vec(Primal and Dual graph2vec)と呼ぶ．本研究でエッジ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，構造の類似性をノードラベルとは独立に評価するのに適している．

PDgraph2vec では，元のグラフ G のエッジラベルと構造情報の片方をユーザが選択し，LG の分散表現に反映する．典型的には，G がエッジラベルを持たない場合，ラングラフ LG においてノード次数をノードラベルとすることで G のノードラベルとは独立に G の構造情報を LG の分散表現に畳み込む．G がエッジラベルを持つ場合は、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づいた LG の分散表現を獲得する．

その一方で，LG に対する分散表現のみを使用すると，逆に G のノードラベルが無視されてしまうので，G に対する分散表現も同時に使用する．提案手法の手順は下記のように構成される．

(step 1)　$\mathbf{G} = \{G_1, G_2, \ldots G_N\}$ の各グラフ G_i をラングラフ LG_i に変換して、$\mathbf{LG} = \{LG_1, LG_2, \ldots LG_N\}$ とする．さらに、変換したLGに対し W-L 再ラベル操作を実行するため、事前にノードにラベル λ_{lv} を与えるときには、データセットにより、下記のような二つ場合で分ける：

① G_i がエッジラベルを持たない場合、LG_i の各ノード lv_jのラベルを次数$\deg(lv_j)$ とする．

② G_i がエッジラベルを持つ場合、LG_i の各ノード lv_jのラベルを，対応する G のエッジラベルとする．．

(step 2)　$\mathbf{G}$ に対して graph2vec モデルを適用する．その結果、G_i に対して δ 次元の特徴ベクトル$f(G_i)$ が得られる．

(step 3)　$\mathbf{LG}$ に対して graph2vec モデルを適用する．その結果、LG_i に対して δ 次元の特徴ベクトル$g(LG_i)$ が得られる．

(step 4)　G_i に対して、$f(G_i)$ と$g(LG_i)$ を連結した2δ 次元のベクトルF_i をG_i に対する最終的な特徴ベクトルとする．

4. 実験

本章では、提案手法の優位性を検証するために、ソーシャルネットワーク、生物情報分野と化学情報分野分野におけるいくつかのベンチマークデータセットを用いてグラフ分類タスクを行う．

補助特徴量としてのラングラフの特徴ベクトルが元のグラフの特徴ベクトルと連結する形で、分類タスクの精度を向上させるのかどうかを検証する．

4.1 データセット

実験では二種類のデータセットを用意する．エッジラベルなしのデータセット [4] エッジラベルありのデータセット [7] [8] に対応するグラフの数、平均ノード数、ノードラベルの数とエッジラベルの数に関する統計量には表 1 で表す．

4.1.1 エッジラベルなしデータセット

MUTAG、PTC、NCI1、NCI109．これらのデータセットは化学情報学の分野からのデータである．化学データ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．なお、ノードは原子タイプによってラベル付けされている．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からバランスを考慮してサンプリングしたサブセットであり，4110 個と 4127 個のグラフで構成される．

PROTEINS はタンパク質に関するデータセットで、ノードが 2 次構造要素を表し、エッジがアミノ酸配列または 3D 空間の近傍を表す 1113 個のグラフで構成される．

4.1.2 エッジラベルありのデータセット

MUTAG*, NCI33, NCI83．これらのデータセットも化学情報学の分野からのデータである．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 および NCI83 データセットはそれぞれ黒色腫の癌細胞株および乳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からバランスを考慮してサンプリングしたサブセットであり、2843 個および 3867 個のグラフで構成される．さらに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 19456 個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する．(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

†電気通信大学大学院　情報理工学研究科
Graduate School of Informatics and Engineering,
The university of electro-communications

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．

このデータセットに対するタスクは，各論文の分野が「データベースとデータマイニング」或は「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 クラス分類である．

表 1　データセットの統計量

Dataset	#samples	#nodes (avg.)	#distinct node labels	#distinct edge labels
MUTAG	188	17.9	7	-
PTC	344	25.5	19	-
PROTEINS	1113	39.1	3	-
NCI1	4110	29.8	37	-
NCI109	4127	29.6	38	-
MUTAG*	188	17.9	7	4
NCI33	2843	30.2	29	4
NCI83	3867	29.5	28	4
DBLP	19456	10.5	41324	3

4.2 実験設定

根付き部分グラフの最大深さ H を 3 にする．Graph2vec モデルで学習した分散表現の次元数を 1024 次元にする (つまり連結したの分散表現が 2048 次元)．最終の分散表現がグラフの特徴量として SVM を用いて分類する．分類するときに学習セットとテストセットをランダムに 90%、10% の割合で分けて、5 分割交差検証で分類器のハイパーパラメータを設定する．20 回の実験結果で分類精度を評価する．

4.3 実験結果

エッジラベルなしデータセットに対するグラフ分類タスクの正解率を表 2 に，エッジラベルありデータセットに対する正解率を表 3 に示す．

表 2　エッジラベル無しグラフ分類の正解率 (平均±標準偏差)%

Datasets	MUTAG	PTC	PROTEINS	NCI1	NCI109
Graph2vec	83.68 ± 7.02	61.00 ± 5.58	72.50 ± 6.16	75.82 ± 2.72	75.87 ± 2.27
PDgraph2vec	86.58 ± 5.78	60.57 ± 4.41	70.09 ± 5.52	77.77 ± 2.34	79.69 ± 2.04

表 3　エッジラベルありグラフ分類の正解率 (平均±標準偏差)%

Datasets	MUTAG*	NCI33	NCI83	DBLP
Graph2vec	83.68 ± 7.02	78.95 ± 1.82	75.90 ± 1.66	90.63 ± 0.59
PDgraph2vec	87.63 ± 7.50	81.30 ± 217	77.29 ± 1.31	92.27 ± 0.62

表 3 の 結果から見ると、MUTAG、NCI1、NCI109 三つのデータセットで提案手法が従来手法よりそれぞれ 2.90%、1.95%、3.82%で分類精度が上がっている．そして PTC に対して、提案手法と従来手法が同じぐらい精度が出る．なお、PROTEINS に対して、正解率が 2.41%で下がっている．また、表 4 の結果から見ると、すべてのエッジラベルありのデータセットに対して、提案手法が従来手法より分類精度が回っている．

つまり、9 個のデータセットに対して、提案手法が従来手法より 8 個のデータセットで分類精度を回っている．

実験結果により、補助特徴量としてのラインググラフの分散表現が元のグラフの分散表現と連結する形で、分類タスクの精度を向上させることが明らかになった．

5.　結論

本研究では、ラインググラフでサポートされるエッジ情報を活用してグラフの分散表現を強化するための graph2vec の改良手法としての PDgraph2vec を提案した．ラインググラフの次数分布や元のグラフおけるエッジラベルなどの情報を活用することで、ノード属性に基づく分散表現とエッジ情報に基づく分散表現を結合し，ノード属性とエッジ情報の両方を考慮したベクトル表現を生成する．実験により，グラフ分類タスクにおける多数のベンチマークデータセットに対して，提案手法が graph2vec よりも分類精度を向上できることを示した．

謝辞

Graph2vec の作者に、従来手法のソースコードとデータセットを公開してくれたことに感謝する．

参考文献

[1] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., “Distributed representations of words and phrases and their compositionality”, In: Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, USA (2013)
[2] Le, Q., Mikolov, T., “Distributed representations of sentences and documents”, In: Proceedings of the 31st International Conference on International Conference on Machine Learning, Beijing, China (2014).
[3] Grover, A., Leskovec, J. “DeepWalk: Online Learning of Social Representations”, In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA (2016)
[4] Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., Jaiswal, S., “graph2vec: Learning Distributed Representations of Graphs”, In: Proceedings of 13th International Workshop on Mining and Learning with Graphs, Halifax, Nova Scotia, Canada. (2017)
[5] Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M., “Weisfeiler–Lehman Graph Kernels”, Journal of Machine Learning Research 12, 2539–2561 (2011)
[6] https://en.wikipedia.org/wiki/Line_graph
[7] Zhu, X., Zhang, C., Pan, S., Yu, P., “Graph stream classification using labeled and unlabeled graphs”, In: Proceedings of the 2013 IEEE International Conference on Data Engineering, Brisbane, Australia (2013)
[8] https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets