Vol.2020-DBS-172 No.10
2020/12/22

確率的なリラベルを用いたグラフ分類の精度向上

辻川 拓摩[1,a)]　猪口 明博[1]

概要：現実世界には様々なデータが存在し，データ解析の重要性はとても大きくなっている．様々なデータの中から，本論文ではグラフ構造を持つデータの解析に注目し，グラフで表現できるデータを適切なクラスに分類することを目的とする．グラフを分類するためには，グラフ間の類似度を求める必要があり，そのためのアプローチとして Label Aggregate Kernel (LAK) を扱う．この LAK では，リラベルを用いて，ある頂点から h ステップ以内で到達可能な頂点集合で誘導される部分グラフで類似度を測る．本論文では既存の LAK で抽出できなかった部分グラフを抽出して，類似度を求めれば更に正確な類似度を測れると考え，確率的にリラベルする新たな方法を提案する．既存 LAK と提案手法の分類精度を比較した結果，提案手法の分類精度が高くなるという結果を得た．

Improving Graph Classification using Stochastic Relabeling

1. はじめに

近年，コンピュータの普及により身の周りには膨大な量のデータが満ち溢れている．その膨大な量のデータを解析し，その中に潜在的に存在する有用な知識を発見するデータマイニング技術の研究が注目されている．データから有用な知識を発見するためには，様々なデータ形式に柔軟に対応できる解析手法が求められる．その解析手法の 1 つであるグラフ構造で表現されたデータを解析する方法をグラフマイニングと言う．

グラフ構造で表現できるデータは多数存在する．例えば，鉄道や SNS のネットワークや分子構造などが挙げられる．分子構造をグラフで表現する場合は，原子をグラフの頂点，結合をグラフの辺と対応させる．このように，身の周りに溢れている様々なデータをグラフ構造で扱うことができるため，グラフマイニングは大きな関心を集めている．

本研究では，グラフの分類問題に焦点を当て，その分類精度を向上させる手法を提案する．グラフを分類できることによって様々な利点がある．新薬開発において，1 つの薬を開発するのに 12 年程の期間と 26 億ドルの資金が必要といわれている [1]．この開発コストを減らすために，IT 創薬に注目が集まっている．創薬化学の分野では，互いに類似している化合物は同じ性質を持つことが知られている．この性質を基にして化合物を適切なクラスに分類することができれば，新規化合物の発がん性の有無などといった予測が可能となり，新薬開発がより促進されると期待できる．

グラフを分類するための方法の 1 つとして，クラス間を分離する境界を求める方法がある．この境界は，グラフ間の類似度を求めることにより求まる．文献 [2] に示された手法では，最初にグラフをいくつかの部分グラフに分解し，次にグラフ間同士で共通の部分構造を数え上げることで，グラフ間の類似度を求めていた．しかし，グラフ間の類似度を求めるにあたって既存手法で利用される部分グラフは，ある頂点から h ステップ以内で到達可能なすべての頂点集合 V' による誘導部分グラフに限られていた．この処理において，今まで抽出しなかった部分グラフを抽出してグラフ間の類似度を求めれば，目的変数の正確な予測に寄与する説明変数が増えるため，分類精度の向上に繋がると期待できる．そのため，本研究では V' による誘導部分グラフのみではなく，V' の一部の頂点と V' の間の一部の辺で構成される部分グラフを抽出してグラフ間の類似度を測る新たな提案手法を提案する．

本稿の構成は以下の通りである．まず 2 節でグラフ分類問題について定義し，本研究で扱うグラフカーネルについて説明する．3 節では，既存手法の課題とその課題を克服するために新たな提案手法を提案する．4 節では，3 節で提

[1] 関西学院大学院 理工学研究科
〒 669-1337 兵庫県三田市学園 2 丁目 1 番地
[a)] dpd19039@kwansei.ac.jp

案した手法と近年盛んに研究されている Graph Convolution Network との関連を議論する．5 節では，提案手法の性能を評価するために実験を行い，既存手法との性能の差を検証する．6 節で，本稿をまとめ結論を述べる．

2. グラフ分類問題

2.1 グラフのベクトル表現

本論文では，ラベル付きグラフの分類問題を扱う．そこで, はじめに本稿で扱うラベル付きグラフについて説明する．無向グラフは $g = (V, E, \Sigma, \ell)$ で表される．ここで，V は頂点の集合，$E \subseteq V \times V$ は辺の集合，$\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_{|\Sigma|}\}$ は頂点ラベルの集合であり，$\ell : V \to \Sigma$ は頂点にラベルを割り当てる関数である．さらに，グラフ g の頂点集合と辺集合はそれぞれ $V(g)$ と $E(g)$ で，頂点 $v \in V(g)$ の隣接頂点は $N(g, v)$ で表される．本研究では頂点のみがラベルを持つと想定しているので，辺の種類は，文献 [3] によって，頂点とそのラベルに変換されたうえで扱われる．

グラフ $g = (V, E, \Sigma, \ell)$ と $g' = (V', E', \Sigma', \ell')$ が与えられ，$\forall v, v_1, v_2 \in V'$ に対して，単射 $\varphi : V' \to V$ が以下を満たすとき，g' を g の部分グラフと呼ぶ．

- $(\varphi(v_1), \varphi(v_2)) \in E$ if $(v_1, v_2) \in E'$
- $\ell'(v) = \ell(\varphi(v))$
- $\ell'((v_1, v_2)) = \ell((\varphi(v_1), \varphi(v_2)))$

さらに，$(\varphi(v_1), \varphi(v_2)) \in E$ iff $(v_1, v_2) \in E'$ であるとき，g' を g の誘導部分グラフと呼ぶ．

入力として $D = \{(g_i, y_i)\}_{i=1}^n$ が与えられたとする．ここで，$y_i \in \{-1, +1\}$ とする．本研究で扱うグラフ分類問題は，D を学習データとして，$y_i = f(g_i) = f'(\boldsymbol{\phi}(g_i))$ を満たす関数 f あるいは f' を D から学習する問題である．ここで，$\boldsymbol{\phi} : \mathcal{G} \to \mathbb{R}^p$ はグラフ $g \in \mathcal{G}$ を p 次元のベクトルに変換する関数である．あるグラフ g をベクトル $\boldsymbol{\phi}(g)$ に変換する単純な方法は，様々な p 個のグラフの集合 $S = \{g_{s1}, g_{s2}, \ldots, g_{sp}\}$ を事前に用意し，g_{si} の g における部分グラフとしての出現回数 ϕ_i を $\boldsymbol{\phi}(g)$ の i 次元目の要素とする方法である [4]．しかし，この単純な方法では以下に示す 3 つの課題がある．

(1) g_{si} が g に部分グラフとして含まれるかの部分グラフ同型判定問題は NP 完全である．
(2) 正例と負例を正しく分類するために必要な特徴 g_{si} は事前には既知でない．このため正確な分類のために有効な S を事前に用意することは容易ではない．
(3) 2 つ目の課題を解決するために，分析対象となるグラフに部分グラフとして含まれるすべてグラフを列挙，あるいはある閾値以上に出現する部分グラフを列挙する手法が考えられる [5]．しかし，この列挙問題は，頂点と辺の組み合わせ問題であり，それらのグラフをすべて列挙することは容易ではない．また，S の要素を連結グラフ [6] に限定したとしても，同様である．

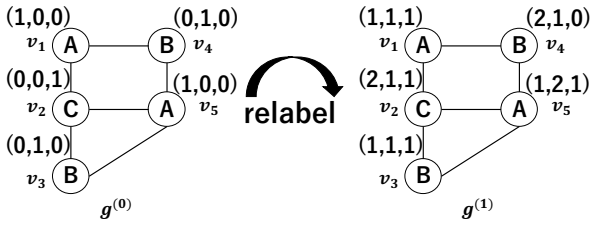

図 1 リラベルの例

これらの課題を部分的に解決する手法として LAK (Label Aggregation Kernel) がある [2]．LAK では，まず頂点ラベルに応じて，各頂点に one-hot ベクトルを割り当てる．具体的には，頂点 v がラベル σ_i を持つとき，この頂点に対して i 番目の要素が 1 である $|\Sigma|$ 次の one-hot ベクトル $\boldsymbol{\ell}^{(0)}(u) = \mathbf{1}_i$ を割り当てる．次に，ある頂点 v がもつベクトル $\boldsymbol{\ell}^{(0)}(v)$ と v の隣接頂点 $N(g, v)$ がもつベクトルを

$$\boldsymbol{\ell}^{(h+1)}(v) = \boldsymbol{\ell}^{(h)}(v) + \sum_{u \in N(g,v)} \boldsymbol{\ell}^{(h)}(u) \qquad (1)$$

で足し合わせ，v がもつベクトルを更新する．この更新をリラベルと呼ぶ．これを η 回繰り返すことにより，グラフの各頂点は以下の $\eta + 1$ 個のベクトルをもつ．

$$D(g, v) = \{\boldsymbol{\ell}^{(0)}(v), \boldsymbol{\ell}^{(1)}(v), \ldots, \boldsymbol{\ell}^{(\eta)}(v)\} \qquad (2)$$

ここで，$D(g, v)$ は多重集合である．$\boldsymbol{\ell}^{(h)}(v)$ は v から h ステップ以内で可達な頂点がもつラベルの分布を表している．このため，h 回リラベルされた 2 つのグラフが同じベクトル $\boldsymbol{\ell}^{(h)}(v)$ をもつことは，これらのグラフが v から h ステップ以内で可達な頂点集合で誘導される部分グラフを持っていると考える．以上より，グラフ g を次式でベクトル化する．

$$\boldsymbol{\phi}(g) = (\phi_1, \phi_2, \ldots, \phi_p)^T = \sum_{h=0}^{\eta} \sum_{v \in V(g)} \mathbf{1}_{id(\boldsymbol{\ell}^{(h)}(v))} \qquad (3)$$

ここで，id は $\boldsymbol{\ell}^{(h)}(v)$ が S において何番目の要素であるかを返す関数であるとする．

リラベルの動作例を図 1 を用いて説明する．ここでは $|V(g)| = 5$，$|\Sigma| = 3$ とし，グラフ $g^{(0)}$ を 1 回リラベルする例を示す．$|\Sigma| = 3$ より，$g^{(0)}$ の各頂点には $(1, 0, 0)$，$(0, 1, 0)$，$(0, 0, 1)$ のいずれかの one-hot ベクトルが割り当てられる．$g^{(1)}$ の頂点 v_2 の頂点ラベルは $(2, 1, 1)$ であり，これは頂点 v_2 から 1 ステップ以内で到達可能な頂点がもつラベルラベル $\sigma_1, \sigma_2, \sigma_3$ がそれぞれ 2 個，1 個，1 個存在することを意味する．

2.2 グラフカーネル

リラベルの計算量は $O(|V(g)|\overline{d}|\Sigma|\eta)$ であり，グラフの頂点数 $|V(g)|$，平均次数 $\overline{d}$，ラベル数 $|\Sigma|$，リラベル回数 η に対して線形であるので，前述の 3 つの課題を克服してい

る．また，これを Support Vector Machine (SVM) に用いる場合，2 つのサンプル g_1 と g_2 に対応するベクトルの内積 $\boldsymbol{\phi}(g_1)^T\boldsymbol{\phi}(g_2)$ の計算が必要になる，この内積は 2 つのグラフの類似度（より一般的には計量と呼ばれる [7]）を表している．

$$
\begin{aligned}
&k(g_1, g_2) = \boldsymbol{\phi}(g_1)^T\boldsymbol{\phi}(g_2) \\
&= \left(\sum_{h=0}^{\eta}\sum_{v\in V(g_1)} \mathbf{1}_{id(\boldsymbol{\ell}^{(h)}(v))}\right)^T \left(\sum_{h=0}^{\eta}\sum_{u\in V(g_2)} \mathbf{1}_{id(\boldsymbol{\ell}^{(h)}(u))}\right) \\
&= \sum_{h=0}^{\eta}\sum_{v\in V(g_1)}\sum_{u\in V(g_2)} \delta(\boldsymbol{\ell}^{(h)}(v), \boldsymbol{\ell}^{(h)}(u)) \qquad (4)
\end{aligned}
$$

ここで δ はクロネッカーのデルタである．このグラフカーネル $k(g_1, g_2)$ の計算量は，実装を工夫することで 2 つのグラフの頂点数の積に比例はせず，一方のグラフの頂点数に比例する程度に抑えることが可能である．

2.3 LAK の課題

LAK によって抽出可能な部分グラフ $g_s \in S$ は，グラフのある頂点から h ステップ以内で到達可能な頂点によって誘導される g の連結な部分グラフに限定される．図 2 は，図 1 の各頂点から 1 ステップ以内で到達可能な頂点集合で誘導される連結部分グラフと各頂点を 1 回リラベルした後に各頂点が持つベクトルを表したものである．2 節で挙げた課題 3 では，あるグラフ g のすべての連結な部分グラフを列挙することを述べている．一方で，式 (2) で示される $D(g, v)$ の要素は g の連結な “誘導” 部分グラフに相当するラベルに限られている．このように，2 節において LAK が “部分的解決する” 手法だと述べた理由は，LAK がグラフ g のすべての部分グラフを S の候補としているわけではなく，その一部しか探索していないからである．図 1 に示されるグラフの連結部分グラフのうち，図 2 に掲載されていないものを図 3 に示す．この図 3 に示されたグラフは従来の LAK では S の要素として抽出できないものである．しかし，これらの中に，目的変数の予測に寄与するものが含まれている可能性があり，目的変数に関わる特徴の種類を $D(g, v)$ に増やすことができれば，分類精度の向上を期待できる．そこで，本研究では，$D(g, v)$ の要素を連結な部分グラフに相当するラベルに拡張し，式 (4) で示されるグラフカーネルを計算することを考える．

3. 確率的なリラベル

前節で述べた LAK の課題を解決するために，確率的なリラベルとそれを用いたグラフカーネルを提案する．提案手法（これ以降，pLAK と呼ぶ）の擬似コードを Algorithm 1, 2, 3 に示す．Algorithm 1 の τ とそれに関わる繰り返しについては後で説明するため，まずは $\tau = 1$ として，Algorithm 1 の 4 行目の繰り返しはないものとして，読んでいただきたい． Algorithm 1 は，入力として与えられたグラフ g_1^{in} と g_2^{in} を η 回リラベルしながら，それらのグラフをベクトル表現し，それらの内積を返す．Algorithm 1 では，h 回リラベルされたグラフをベクトル表現する関数 vec からなる Algorithm 2 とグラフを辺選択確率 p に基づいて確率的にリラベルする関数 $relabel$ からなる Algorithm 3 が用いられる．

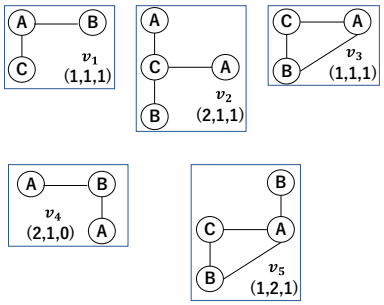

図 2 LAK で抽出可能な連結誘導部分グラフ

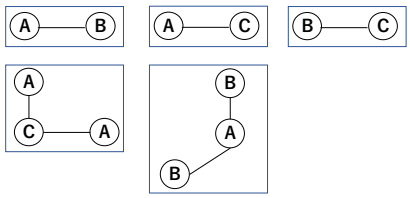

図 3 LAK で抽出できない連結部分グラフ

Algorithm 1: 2 つグラフ g_1 と g_2 の計量

Data: $g_1^{in}, g_2^{in}, \tau, \eta$, and p
Result: $k(g_1, g_2)$

```
map ← ∅;
ϕ_1 ← 0;
ϕ_2 ← 0;
for t ∈ [1, τ] do
    g_1 ← g_1^in;
    g_2 ← g_2^in;
    for h ∈ [0, η] do
        ϕ_1 ← ϕ_1 + vec(g_1, h, map);
        ϕ_2 ← ϕ_2 + vec(g_2, h, map);
        g_1 ← relabel(g_1, p, h);
        g_2 ← relabel(g_2, p, h);
return (1/τ) ϕ_1^T ϕ_2;
```

Algorithm 2 では，グラフ g の各頂点のラベル $\boldsymbol{\ell}^{(h)}(v)$ をキー，整数値を値とするハッシュ関数 map を用いて，$\boldsymbol{\ell}^{(h)}(v)$ が S の何番目のグラフに相当するかを返す．その値を $value$ とすると，$value$ 次元目が 1 である one-hot ベクトルを得て，$\boldsymbol{\phi}$ に加算し，$\boldsymbol{\phi}$ を返す． Algorithm 3 では，グラフ g の辺集合 $E(g)$ から辺を確率 p で選択する関数 $select$ を用いて，確率的なリラベルを実現する．4 行目では，グラフ g の辺の部分集合をもつグラフ g' が用いられる．例

えば，図 1 の $g^{(0)}$ の辺のうち，辺 (v_2, v_5) が $select$ により選ばれなかったとする．この場合，$g^{(1)}$ の v_5 がもつラベルは $(1, 2, 0)$ となり，図 1 で示されるリラベルとは異なった動作となる．また，この $(1, 2, 0)$ は図 3 の右下のグラフに相当するため，この節で提案した pLAK は，前述の課題を克服しているといえる．

Algorithm 2: vec

Data: g, h, and map
Result: h 回リラベルされた g のベクトル表現 $\boldsymbol{\phi}$

$\boldsymbol{\phi} \leftarrow \mathbf{0}$;
for $v \in V(g)$ **do**
  $key \leftarrow \boldsymbol{\ell}^{(h)}(v)$;
  $value \leftarrow map(key)$;
  if $value = null$ **then**
    $map \leftarrow map \cup \{(key, value)\}$;
    $value \leftarrow |map|$;
  $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \mathbf{1}_{value}$;
return $\boldsymbol{\phi}$;

Algorithm 3: $relabel$

Data: g, p, and h
Result: h 回リラベルされたグラフ g'

$E \leftarrow select(E(g), p)$;
$g' \leftarrow (V(g), E, \mathbb{Z}^{|\Sigma|}, null)$;
for $v \in V(g)$ **do**
  $\boldsymbol{\ell}^{(h+1)}(v) \leftarrow \boldsymbol{\ell}^{(h)}(v) + \sum_{u \in N(g',v)} \boldsymbol{\ell}^{(h)}(u)$;
$g' \leftarrow (V(g), E, \mathbb{Z}^{|\Sigma|}, \boldsymbol{\ell}^{(h+1)})$;
return g';

続いて，説明を先送りした τ の役割について説明する．式 (4) で示されるグラフカーネルは，引数として与えれる 2 つのグラフ g_1 と g_2 が似ていれば大きな値を返し，似ていなければ小さな値を返す．ここでは説明のため以下の 2 つの場合を考える．1 つ目は，Algorithm 1 に対して同型のグラフを引数として与え，$relabel$ において $select$ が空集合を返す場合である．この場合，辺を含む部分グラフは S の要素として抽出されず，$h \geq 1$ に対する $vec(g_1, h, map)$ と $vec(g_2, h, map)$ の内積はゼロになる．このため，Algorithm 1 が返す値は，同型なグラフを与えているにも関わらず，小さくなる．2 つ目の場合は，Algorithm 1 に対して同型でないグラフを引数として与え，それらのグラフを確率的にリラベルしたとしても，2 つのグラフが共通のラベルを持つ場合である．リラベルしても 2 つのグラフが共通のラベルをもつため，Algorithm 1 が返す値は，1 つ目の場合よりも大きくなるかもしれない．これは，類似したグラフに対して大きな値を返すグラフカーネルの性質に反する．また，このような場合，学習データ D の g_i と g_j に対する $k(g_i, g_j)$ を (i, j) 要素にもつカーネル行列 K は，半整定値性を満たさず，SVM などの学習アルゴリズムに適用することができない．このような現象は，ある頂点の隣接頂点を $select$ により確率的に選択して，2 つのグラフの類似度を計算することから生じている．確率的な隣接頂点の選択の影響を緩和するために，Algorithm 1 の 5 行目から 11 行目を τ 回繰り返す．このため，pLAK は LAK に比べ τ 倍の計算時間を要し，これが pLAK の欠点である．

4. Graph Convolution Network との関連

近年, 深層学習分野でもグラフの分類手法は盛んに研究されている．それらの手法の多くは，Graph Convolution Network（GCN）[8] [9] を用いている．グラフ g の隣接行列を A とする．また，次数行列を $D = diag(d_1, d_2, \ldots, d_{|V(g)|})$ とする．ここで，A の (i, j) 要素を a_{ij} とすると $d_i = \sum_{j=1}^{|V(g)|} a_{ij}$ である．GCN では，グラフの各頂点は m 次の特徴ベクトル（この特徴ベクトルは，one-hot ベクトルとは限らない）をもち，i 番目の頂点 v_i の特徴ベクトル $\boldsymbol{x_i}(v_i)$ を i 行目にもつ行列 $X^{(0)}$ を

$$X^{(h+1)} = \sigma\left(D^{-1}(I + A)X^{(h)}W^{(h)}\right) \qquad (5)$$

で，順伝番する．ここで，$W^{(h)}$ は逆伝搬により学習される学習パラメータ，σ は活性化関数である．

この節では GCN と LAK の関連を述べ，さらに pLAK との関連を議論する．式 (5) において，σ を恒等関数，$W^{(h)}$ を単位行列とすると，式 (5) は

$$X^{(h+1)} = D^{-1}(I + A)X^{(h)} \qquad (6)$$

となる．さらに，ある頂点がもつ特徴ベクトルをその頂点がもつ離散ラベルに応じた one-hot ベクトルとすると，式 (6) の i 行目は，

$$\boldsymbol{x}^{(h+1)}(v) = \frac{1}{|N(g, v)|}\left[\boldsymbol{x}^{(h)}(v) + \sum_{u \in N(g,v)} \boldsymbol{x}^{(h)}(u)\right]$$

となる．この式は式 (1) とほぼ等価である．したがって，式 (1) は学習パラメータ $W^{(h)}$ や活性化関数のない順伝搬のみの GCN と見なすことができる．

GCN をはじめ多くの深層学習法は確率的な挙動ができない．変分自己符号化器（VAE: Variational Autoencoder) [10] [11] では，本来，確率的な挙動が必要となるが，それが難しいため，ニューラルネットワークの外部で乱数をとり，その乱数を入力として与えることによって，擬似的に確率的な挙動を実現している．これに対して，pLAK は，Algorithm 3 において，つまり pLAK の内部において，確率的に隣接頂点を選択し，足し合わせる隣接頂点のベクトルを得る．このため，pLAK は近年多数提案されているグラフに対する深層学習法では抽出できない潜在変数を S の要素として抽出できている可能性がある．

5. 評価実験

5.1 実験設定

提案手法を Java を用いて実装し，pLAK で得られるカーネル行列を LIBSVM [12] の入力として与えて，評価実験を行った．分類精度は 10 回交差検定で得られたものである．pLAK の分類精度を評価するために表 1 に示す 2 種類のデータセットを用いる．1 つ目のデータセットは MUTAG [13] であり，このデータセットには 188 個の化合物と各化合物に対する変異原性の有無の情報が含まれている．2 つ目のデータセットは ENZYMES [14] であり，このデータセットは 600 個のタンパク質の構造情報が含まれている．また，それぞれのタンパク質に対して 1 から 6 までの 6 種類の酵素番号がクラスとして割り当てられている．

表 1 評価用データセットの概要

	MUTAG	ENZYMES
グラフ数	188	600
グラフの最大頂点数	84	126
平均頂点数	53.9	32.6
ラベル数 $\|\Sigma\|$	12	3
クラス数	2	6
平均次数	2.1	3.8

5.2 カーネル行列の半正定値性

3 節で述べたように τ が小さいとき pLAK で求められるカーネル行列が半正定値性を満たさない可能性があるので，τ をどの程度に設定するとカーネル行列が半正定値性を満たすのかを確認する．これを確認するために，カーネル行列の最小固有値を求めた．図 4 と図 5 は，2 つのデータセットに対して，τ を変化させたときの，カーネル行列の最小固有値の変化である．この実験では，ある τ に対して，乱数の種値を変え，10 回の実験を行い，最小固有値の平均値と最小値を掲載している．この実験では $\eta = 4$, $p = 0.99$ に設定した．τ が小さいとき，最小固有値の値は 0 よりも小さいが，τ を 1000 以上に設定することで，0 に収束することを確認できた．このため，これ以降の実験では，MUTAG では $\tau = 10000$ に，ENZYMES では $\tau = 1000$ に固定して評価実験を行った．

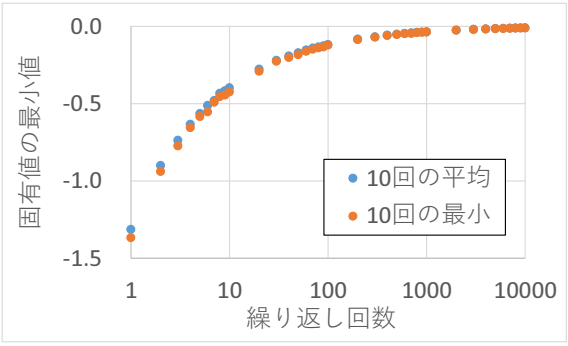

図 4 τ の変化に対するカーネル行列の最小固有値の変化（MUTAG）

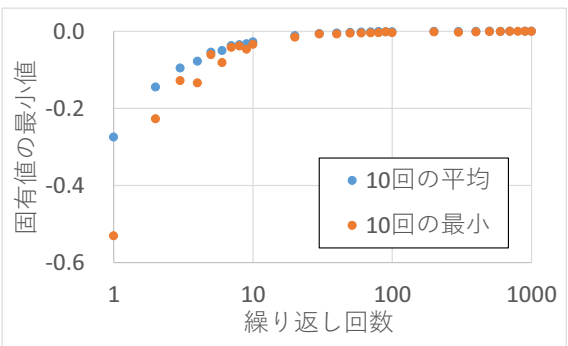

図 5 τ の変化に対するカーネル行列の最小固有値の変化（ENZYMES）

5.3 分類精度

次にリラベル回数 η と辺選択確率 p を変化させながら，分類精度がどのように変化するのかを調査した．その実験結果を図 6 と図 7 に示す．確率 $p = 1.0$ のときは，すべての辺を選択することになるので，この時の分類精度は従来の LAK と同じである．よって，この実験結果は，LAK と pLAK の比較でもある．まず，図 6 の MUTAG に対する LAK の実験結果を見ると $\eta = 6$ のときに分類精度が最も高くなった．これは，ある頂点から 6 ステップ以内で到達可能な頂点集合で誘導される部分グラフが，分類精度の向上に寄与する特徴量になることを示唆している．続いて，pLAK の分類精度を見ると辺選択確率 $p = 0.991$，リラベル回数 $\eta = 4$ のとき，分類精度は最も高くなり，LAK に比

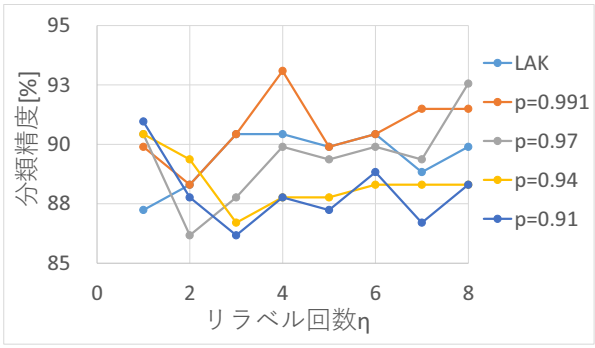

図 6 η と p の変化に対する分類精度の変化（MUTAG）

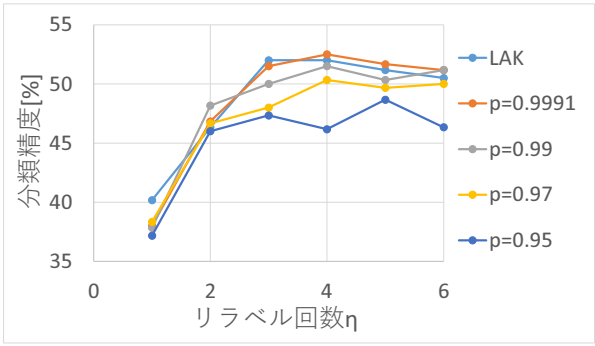

図 7 η と p の変化に対する分類精度の変化（ENZYMES）

べ分類精度が約 2.7%向上した．また，p を 0.991 から下げると分類精度が減少することが分かった．

図 6 の実験結果を得るために設定した辺選択確率 p は非常に大きい．この理由を考察する．図 8 は 3 頂点からなる小さなグラフである．v_1 に着目し，v_1 の周辺にラベル C をもつ頂点があるかどうかを $\ell^{(h)}(v_1)$ で表現するためには，このグラフを少なくとも 2 回リラベルする必要がある．図 8 はその 2 回のリラベルの過程を示している．pLAK では，確率 p である辺をグラフ g に残すかを決めるが，v_1 の周辺にラベル C をもつ頂点があるかどうかを $\ell^{(h)}(v_1)$ で表現するためには，1 回目のリラベルで辺 (v_1, v_2) と (v_2, v_3) が残され，2 回目のリラベルで (v_1, v_2) が残される必要がある．このため，2 回のリラベルによって，$\ell^{(2)}(v_1)$ の 3 番目の要素がゼロでない確率は p^3 となる．一般的に，s 回リラベルされることで，ある頂点 v から s ステップの位置にある 1 つの頂点が，v のラベルに影響を及ぼす確率 q は $p^{\frac{s(s+1)}{2}}$ となる．図 9 に示すように，$p = 0.99, s = 4$ のとき $q = 0.90$ で大きな値になる．しかし，$p = 0.9, s = 4$ のとき $q = 0.35$ となり，辺選択確率 p が大きくても q は小さくなる．このため，この実験では，pLAK の辺選択確率は非常に大きく設定されており，実際，p が大きな値の時に，pLAK の分類精度は高くなる．同様の傾向は，ENZYMES に対する実験に対しても見られた．

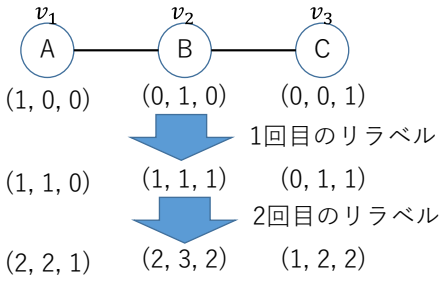

図 8 リラベルの過程

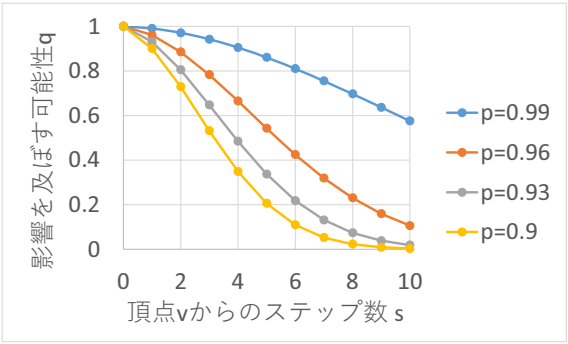

図 9 v から s ステップの位置の頂点が v に影響を及ぼす確率 q

6. まとめ

本論文では，グラフ分類問題において，これまで抽出できなかった部分グラフを特徴量として抽出するために，確率的にリラベルする pLAK を提案した．提案した pLAK を幾つかの実データセットに適用し，従来の LAK に比べ僅かながらではあるが分類精度向上を達成した．この分類精度向上は，LAK では特徴量として抽出できなかった部分グラフを pLAK が抽出できたからだと考えられる．

参考文献

[1] 関嶋 政和. オープンイノベーションによる IT 創薬. 情報管理, Vol. 58, No. 12, pp. 900–907 (2016).
[2] Tetsuya Kataoka and Akihiro Inokuchi. Hadamard Code Graph Kernels for Classifying Graphs. *Proc. of International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pp. 24–32 (2016).
[3] Shohei Hido and Hisashi Kashima. A Linear-Time Graph Kernel. *Proc. of International Conference on Data Mining (ICDM)*, pp. 179–188 (2009).
[4] Hiroto Saigo and Koji Tsuda. Iterative Subgraph Mining for Principal Component Analysis. *Proc. of International Conference on Data Mining (ICDM)*, pp. 1007–1012 (2008).
[5] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data. *Proc. of Principles of Data Mining and Knowledge Discovery (PKDD)*, pp. 13–23 (2000).
[6] Xifeng Yan and Jiawei Han. gSpan: Graph-Based Substructure Pattern Mining. *Proc. of International Conference on Data Mining (ICDM)*, pp. 721–724 (2002).
[7] 椿真史, Kevin Duh, 新保仁, 松本裕治. ニューラルネットワークによる意味構成とそのカーネル埋め込みを用いた多層非線形類似度学習. 情報処理学会研究報告, Vol. 2015, No. 5, pp. 1–12 (2015).
[8] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *Proc. of International Conference on Learning Representations (ICLR)*, (2017).
[9] Matteo Togninalli, M. Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten M. Borgwardt. Wasserstein Weisfeiler-Lehman Graph Kernels. *Proc. of Advances in Neural Information Processing Systems*, pp. 6436–6446 (2019).
[10] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *Proc. of International Conference on Learning Representations (ICLR)*, (2014).
[11] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *Proc. of International Conference on Machine Learning (ICML)*, pp. 1278–1286 (2014).
[12] Chih-Chung Chang and Chih-Jen Lin. Software available at (https://www.csie.ntu.edu.tw/ cjlin/libsvm/).
[13] Asim Kumar Debnath, Rosa L. Lopez de Compadre, Gargi Debnath, Alan J. Shusterman, and Corwin Hansch. Structure-activity Relationship of Mutagenic Aromatic and Heteroaromatic Nitrocompounds. Correlation with Molecular Orbital Energies and Hydrophobicity. *Journal of Medicinal Chemistry*, Vol. 34. pp. 786–797 (1991).
[14] Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. Updates and Major New Developments. *Nucleic Acids Research*, Vol. 32. pp. D431–D433 (2004).