和文タイトル

English Title

著者 1 氏名 [1*]　著者 2 氏名 [1,2]
Author1[1]　Author2[1,2]

[1] 所属機関 1 (和文)
[1] Affiliation 1 (English)
[2] 所属機関 2
[2] Affiliation 2

Abstract:

1　はじめに

aaa[2] takigawa[3]

本章では、グラフ分類問題に対して部分グラフ共起を用いたモデル学習手法を提案する。また、実データを用いた実験により、部分グラフ共起のもたらす効果を検証する。

2　部分グラフ共起を用いた学習について

本稿では、部分グラフの共起構造を考慮したモデル構築法を提案する。まず、構築するモデルを定義する。存在しうるすべての部分グラフ集合を $\mathfrak{T}$ とする。その時、入力のグラフそれぞれに対して $|\mathfrak{T}|$-次元のベクトル

$$\boldsymbol{x} = I(t \subseteq G_n), \forall t \in \mathfrak{T}$$

が考えられる。このベクトルに対して、仮説 (hypotheses) と呼ばれる、弱学習器を以下で定義する。

$$h(\boldsymbol{x}; t, \omega) = \omega(2x_t - 1)$$

$\omega \in \Omega = \{-1, 1\}$ はパラメータである。この時、以下のモデルを考える。

$$f(\boldsymbol{x}) = \sum_{(t,\omega)\in\mathfrak{T}\times\Omega} \alpha_t h(\boldsymbol{x}; t, \omega) + \sum\sum \alpha_{i,j} x_i x_j$$

1 つ目の項は 1 次の項であり、2 つ目の項が部分グラフの共起を表現しており、i と j の部分グラフが同時に現れる場合を考えた 2 次の項である。本研究では、3 次以上の項は考えず、2 つの部分グラフの共起のみ考慮する。

ここで、$|\mathfrak{T}|$ は、入力グラフ集合に対するすべての部分グラフ数存在し、全列挙は現実的に困難であることが知られている。これに対して共起を考えると、単純に $|\mathfrak{T}|^2/2$ の特徴量が増加する。そのため、列挙をした後に学習する手法では解くことが現実ではないと考えられる。よって、本稿では、既存手法である gBoost 法がブースティングの手法であることに着目して、真に全列挙せず、必要なときに必要な部分のみを探索することで全部分グラフとその共起を考慮したグラフ分類手法を提案する。これを提案手法 1:厳密法とし、2.1 節に示す。

提案手法 1 では、厳密に全部分グラフとその共起を用いたモデル構築が行う。しかし、厳密に探索せずに良い特徴を選ぶことができれば、探索を減らし計算時間を削減できる可能性がある。この時、適度に探索を少なくすることでモデル構築にかかるコストを減らすことが期待できる。そこで、提案手法 2:Top-k 法を提案し、2.2 節に示す。

2.1　提案手法 1:厳密法

全部分グラフとその共起を用いたグラフ分類の手法を提案する。まずグラフ分類を、入力グラフに存在する部分グラフを特徴量としてモデル構築を考える。この時、2 種類のモデル学習手法が存在する。1 つ目は、存在する部分グラフを列挙した後、任意の学習モデルを構築する方法である。これは存在する部分グラフが膨大な数であるため、一般的にグラフデータに対する出現数や部分グラフのサイズで探索を打ち切る必要がある。2 つ目は、探索を随時行いながらモデル構築を行う手法である。ブースティング法でモデルを学習す

*連絡先：(所属機関名)
(所属機関住所)
E-mail:

ることで、毎回必要な部分グラフの探索を必要な部分のみ行うことで、全部分グラフを用いたモデル学習を行うことができる手法である。

ここで、部分グラフの共起を考えた学習モデルを考える。入力グラフに含まれる部分グラフを全列挙することは現実的ではないことは知られており、先程述べたように、何らかの指標を用いて列挙をおこなったとする。この時、列挙された部分グラフ数 N に対して、その共起構造は $N^2/2$ 存在する。部分グラフ数 10000 列挙したとすると、50000000 の共起が存在する。そのため、これらすべての特徴を用いた線形モデルの学習には、その数のパラメータの予測が必要である。しかしこれに対して、一般的に用いられているデータ数は数百から数千であることが多い。よって、実際に必要である特徴の数はそれほど多くないと考えられる。今回は、L1 正則化によって実際に使う特徴数を削減できる既存手法である gBoost 法をベースにモデル構築を行うことで、必要な部分グラフやその共起を探索しつつ効率よくモデル構築を行う手法を提案する。

グラフ分類に対して、部分グラフとその共起を用いたモデルを構築するために、gBoost 法を拡張する。基本的なアイデアは、gBoost 法の繰り返し行われる部分グラフ探索に共起を探索するように拡張するものである。まず、グラフ探索を 1 度行い、この時最大 gain を持つグラフが 1 つに決まる。そして、共起を探索する際、bound がこの現在の最大 gain を超えない部分グラフ同士のみ共起を考えれば十分である。

Algorithm1 に擬似コードを示す。まず、gBoost 法と同様の部分グラフ探索を行い、存在する部分グラフの中で、最大の gain を持つ部分グラフを発見する。続いて project 関数を呼び出し、再帰的に共起を探索する 1 つ目の部分グラフを探索する。この時、既に見つけている最適パターンを用いることで枝刈りが有効である。そして、共起を考えられるパターンそれぞれに対して Cooc_project 関数を適用する。この時、1 つ目と 2 つ目の部分グラフを入れ替えた実質同様の共起を探索しないために、部分グラフ探索アルゴリズムである gSpan[2] で用いられたグラフ表現である DFS コードとその辞書順 (以下、DFS 辞書順) を用いる。

アルゴリズムの効率化を図るために、次に説明する 2 つの技巧を使うべきである。このアルゴリズムにおいて、gain や bound の計算は 1 回は計算量のそれほどかからない処理であるが、これが $|\mathfrak{T}|^2/2$ に相当する数となると話は別である。そのため、実際にアルゴリズム中で計算する共起の種類を可能な限り削減する必要がある。まず、1 つ目は、部分グラフ同士が包含関係にあるときである。つまり、ある部分グラフが共起を考えようとしている部分グラフに含まれている場合、その共起は片方の部分グラフと同じ gain、bound を持つことがわかる。そのため、そのような場合、gain や bound

Algorithm 1 部分グラフとその共起の探索 (厳密法)

procedure 最適な部分グラフ、あるいはその共起
  グローバル変数 : $g^*, \boldsymbol{\omega}^*, p^*, pc^*$
  最適な部分グラフ $g^*, \boldsymbol{\omega}^*, p^*$ を探索 ▷ gBoost 法の探索
  for all $p \in$1 つの edge からなる DFS コード **do**
    project(p)
  end for
end procedure
function PROJECT(p)
  if p が最小 DFS コードでない **then**
    return
  end if
  p に対する gain $g(p)$,bound $b(p)$ を計算
  if $b(p) < g^*$ **then**
    return
  end if
  for all $t \in$1 つの edge からなる DFS コード **do**
    Cooc_project(p,t)
  end for
  for all $p' \in$ Rightmost Extension of p **do**
    project(p')
  end for
end function
function COOC_PROJECT(p,t)
  if t が最小 DFS コードでない **then**
    return
  end if
  if $p < t$ **then** ▷ DFS 辞書順
    return
  end if
  t に対する gain $g(t)$,bound $b(t)$ を計算
  if $b(t) < g^*$ **then**
    return
  end if
  p,t に対する gain $g(p,t)$,bound $b(p,t)$ を計算
  if $b(p,t) < g^*$ **then**
    return
  end if
  if $\omega \in \Omega$ それぞれに対して $g(p,t) > g^*$ **then**
    $g^* = g(p,t), p^* = p, \boldsymbol{\omega}^* = \boldsymbol{\omega}$
  end if
  for all $t' \in$ Rightmost Extension of t **do**
    Cooc_project(t')
  end for
end function

の計算は不要である。しかし、これを判定するには部分グラフ同型判定問題を解く必要があるが、この問題のクラスは NP 完全であることがわかっている。そのため、これを毎共起ごとに判定するのは非現実的である。そこで、gSpan アルゴリズムにより各部分グラフは DFS コードで表されていることを用いて一部効率化することを考える。ここでサイズ l の部分グラフが DFS コードで与えられた時、その DFS コードから 1 つずつ最後を除いたグラフがすべて存在し、共起の比較が存在する。これに関して gain と bound の計算をせず現在のノードをスキップする事ができる。2 つ目は、最適パターンの性質に関する技巧である。最適パターンは、親子関係にある部分グラフが同じ 0-1 のパターンを持つ時、親を取るという性質がある。よって、部分グラフを探索している時、親と同じ支持度を持つグラフに対して共起を考える必要がない。以上の 2 つを取り入れることで実際に共起を探索するノードを削減することができる。

2.2 提案手法 2:Top-k 法

提案手法 1 では厳密に全共起を探索することで、全部分グラフとその共起すべてを用いたグラフ分類を行った。しかし、計算時間の観点ですべてを見るのは非効率であることがある。毎イテレーションで必要となる部分グラフが違うことに対して、毎回すべての共起を見ず、適度に探索を効率化する手法が必要である。そこで本稿では、共起を探索する際、ある指標を与えて共起を考慮する部分グラフ k 個を決め、その部分グラフと全部分グラフの共起を探索する手法を提案する。このようにすることで、探索する特徴数は概算 $k*|\mathfrak{T}|$ となり、パラメータとして k を与え、$k=|\mathfrak{T}|$ の時、提案手法と同じモデル構築となるため、パラメータ k を与えることで、gBoost 法と厳密法のトレードオフを達成する。部分グラフパターンの gain $g(t)$、bound $b(t)$ を用いて

$$\lambda|g(t)|+(1-\lambda)b(t) \tag{1}$$

Top-k の指標をとする。gain は直接的にその部分グラフの良さを表し、bound はその部分グラフの子ノード、あるいはその部分グラフとの共起により良いパターンが見つかる可能性を表す。そのため、gain と bound のトレードオフとなるこの指標を用いる。

Algorithm2 に擬似コードを示す。まず、厳密法と同様に、gBoost 法と同様の部分グラフ探索を行い、存在する部分グラフの中で、最大の gain を持つ部分グラフを発見する。同時に、先に述べた指標 (1) にしたがって、Top-k の部分グラフを探索する。そして、その Top-k パターンそれぞれに対して Cooc_project 関数を適用する。この時、DFS 辞書順による探索打ち切りはしない。

Algorithm 2 部分グラフとその共起の探索 (Top-k)

```
 1: procedure 最適な部分グラフ、あるいはその共起
 2:     グローバル変数 : g*, ω*, p*, pc*
 3:     最適な部分グラフ g*, ω*, p* と Top-k 部分グラフを探索   ▷ gBoost 法の探索
 4:     for all p ∈ Top-k 部分グラフ do
 5:         project(p)
 6:     end for
 7: end procedure
 8: function PROJECT(p)
 9:     for all t ∈1 つの edge からなる DFS コード do
10:         Cooc_project(p,t)
11:     end for
12: end function
13: function COOC_PROJECT(p,t)
14:     if t が最小 DFS コードでない then
15:         return
16:     end if
17:     t に対する gain g(t),bound b(t) を計算
18:     if b(t) < g* then
19:         return
20:     end if
21:     p,t に対する gain g(p,t),bound b(p,t) を計算
22:     if b(p,t) < g* then
23:         return
24:     end if
25:     if ω ∈ Ω それぞれに対して g(p,t) > g* then
26:         g* = g(p,t), p* = p, ω* = ω
27:     end if
28:     for all t' ∈ Rightmost Extension of t do
29:         Cooc_project(t')
30:     end for
31: end function
```

2.3 効率的なモデル構築

本節では、2 つの提案手法に対してより効率よくモデルを構築するための手法を提案する。gBoost 法は主問題を LP で定式化し、その双対問題を列生成法で解く。この時、理論上追加する列の順に関係はない。そこで、まず第一ステップとして、既存手法である gBoost 法で通常の部分グラフで収束条件を満たすまで部分グラフの追加を行う。そしてその後、共起を探索するステップへと移行する。このようにすることで、本当に必要な共起を後に探索する構造となり、探索する共起の数を削減することに期待できる。

これを各提案手法に対して適応した 4 種類の提案手法と既存手法との関係について考察するために 3 節で実データを用いた実験を行う。

3 実験

3.1 使用したデータセット

本稿では、Saigo ら [1] によって用いられた 4 つのデータセットを使用した。表 1 に各データセットのグラフ数、平均ノード・エッジ数、正例と負例の数を示す。

データ名	グラフ数	平均ノード数	平均エッジ数	正例	負例
CPDB	684	25.2	25.6	341	343
Mutag	188	26.3	28.1	125	63
CAS	4337	30.3	31.3	2401	1936
AIDS(CAvsCM)	1503	59.0	61.6	422	1081

表 1: 使用したデータセット

3.2 精度に関する実験

3.2.1 実験設定

3.3 計算時間に関する実験

3.3.1 実験設定

謝辞

参考文献

[1] Hiroto Saigo, Sebastian Nowozin, Tadashi Kadowaki, Taku Kudo, and Koji Tsuda. gboost: a mathematical programming approach to graph classification and regression. *Machine Learning*, Vol. 75, No. 1, pp. 69–89, 2009.

[2] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), 9-12 December 2002, Maebashi City, Japan*, pp. 721–724, 2002.

[3] 瀧川一学. 多数のグラフからの統計的機械学習. 深化する機械学習技術の進展とその応用特集号. システム/制御/情報, Vol. 60, No. 3, pp. 107–112, 2016.