1K1-3

全部分グラフ指示子に基づく決定木勾配ブースティング

Gradient Boosting Decision Tree Learning over All Possible Subgraph Indicators

横山 侑政 *1　瀧川 一学 *1*2
Yusei Yokoyama　Ichigaku Takigawa

*1北海道大学大学院情報科学研究科
Graduate School of Information Science and Technology, Hokkaido University

*2科学技術振興機構，さきがけ
JST, PRESTO

We propose an algorithm for learning gradient boosting decision trees over all possible subgraph indicators for a givin set of graphs. Recent advances in machine learning provide several existing methods that can perform simultaneous feature learning from all possible indicators of subgraphs occurring in a given set of graphs. However these previous methods result in linear models in terms of subgraph indicators. In contrast, gradient boosting decision trees learning gives a nonlinear model of subgraph indicators. We first derive a lower bound of the mean squared error sum of data partitions by the given subgraph feature. Then we present a learning algorithm based on subgraph search with branch and bound. Experimental validations on two benchmark datasets are also reported.

1. はじめに

グラフは自然言語処理 [1]，RNA 二次構造 [2]，低分子化合物の構造式 [3] などの知識処理に幅広く用いられている重要なデータ構造である．近年こうした科学分野のグラフデータが公共データベースに蓄積・整備されるようになり，有効な利活用が喫緊の課題となっている．特に，グラフデータからの教師あり学習は，生命科学や物質科学における構造活性相関や構造物性相関の定量的モデルとして機械学習分野で研究されており，より高精度で高効率な手法が求められている．

グラフデータからの教師あり学習では，特徴量として部分グラフの有無 (部分グラフ指示子) を用いることが多い．この場合，検査する部分グラフ特徴の選択が学習モデルの予測精度を左右する．しかし，適切な特徴集合はデータによって異なる．一方で，与えられたグラフデータに生起している全ての部分グラフを列挙することは，現実的には計算困難である．例えば，300 個程度の典型的な分子グラフデータに生起する部分グラフの総数は，少なく見積もっても 100 億個に上る．従って，グラフカーネル法 [4] や ECFP 法 [5] など，対象となる部分グラフのクラスを予め発見的に制限する手法や，gBoost 法 [6] や Adaboost に基づく手法 [1] など，必要な部分グラフのみを効率的に探索しながら学習を行う方法が提案されている．後者のアプローチでは全ての部分グラフ指示子を探索候補にできるため，有効な特徴量を見逃すリスクがない．そのため，様々なデータに対して適応的に良い予測モデルを構築できる．

従来のアプローチは深さ 1 の決定木を弱学習器としたブースティングに基づいており，実数データに対しては非線形な予測モデルとなるが，0-1 変数となる部分グラフ指示子に関しては線形な予測モデルとなる．実際に，創薬分野の特徴量との比較 [7] においても，部分グラフ指示子の空間で非線形な予測モデルを構築すれば精度向上が期待できるという知見が得られている．

そこで，本論文では，全部分グラフ指示子を対象として決定木の勾配ブースティングを用いた非線形な予測モデルを学習する手法を提案する．決定木は，2 値のブール変数である部分グラフ指示子との適合性が高く，効率よく簡潔な予測ルールを与えることができる．勾配ブースティングの弱学習器として決定木を採用し，予測精度の向上を試みる．

連絡先: 横山侑政: yuuseitori@ist.hokudai.ac.jp，瀧川一学: takigawa@ist.hokudai.ac.jp

2. 準備

本研究では，グラフデータからの教師あり学習問題を考える．すなわち，訓練データは N 個のグラフで，個々のグラフ G_i にはラベル y_i が付与されており，任意のグラフ G に対してラベル y を予測する $y = f(G)$ なる関数 f をこれらのデータから学習することを目的とする．ここで対象とするグラフは，無向グラフで，各頂点と各辺にラベルが付与されているものとする．

特徴量としては全ての部分グラフ指示子を用いる．ただし，部分グラフは連結なもののみを考える．グラフデータに生起する全部分グラフを $g_1, g_2, \ldots, g_K$ とすると，グラフ G の特徴ベクトル X は $X = (\mathbb{I}(G \supseteq g_1), \ldots, \mathbb{I}(G \supseteq g_K))$ となる．ここで $G \supseteq g_j$ は，グラフ G がグラフ g_j と同型な部分グラフを持つことを表し，各部分グラフ指示子 $\mathbb{I}(G \supseteq g_j)$ は $G \supseteq g_j$ が真のとき 1 を返し，偽のとき 0 を返す．また，$G \not\supseteq g_j$ は，グラフ G がグラフ g_j と同型な部分グラフを持たないことを示す．現実的なデータで生起する全ての部分グラフを陽に列挙するのは計算困難である．そこで提案手法では特徴ベクトル X を陽には構築せず，構築した場合と同じ結果が得られる学習方法を構築する．

決定木においてデータを分割するとき，データ集合 $D = \{(G_i, X_i, y_i, r_i) : i = 1, \ldots, N\}$ を考える．ただし，r は残差を示す．データ集合 D を部分グラフ g_j で分割するとは，データ集合 $D_1(g_j) = \{G_i : \mathbb{I}(G_i \supseteq g_j) = 1\}$ と $D_0(g_j) = \{G_i : \mathbb{I}(G_i \supseteq g_j) = 0\}$ に分けることをさす．明らかに $D = D_1(g_j) \cup D_0(g_j)$ となる．

データ集合 D に含まれるデータの残差 r_i の 2 乗誤差を

$$\mathrm{MSE}(D) = \min_c \sum_{G_i \in D} \frac{1}{2}(r_i - c)$$

と書く．c はそれぞれ D に含まれるデータの残差の平均値となる．つまり，$c = \sum_{G_i \in D} r_i / |D|$ である．

3. 決定木の勾配ブースティング

3.1 勾配ブースティング法 [8][9]

勾配ブースティング法は，弱学習器を加法的に追加する学習方法である．弱学習器には何らかの回帰モデルを使えば良いが，決定木が採用されることが多く，本研究でもこれを用いる．非線形な予測モデルである決定木を弱学習器にした勾配ブースティング法は，非線形な予測モデルを構築する．勾配ブースティング法により最終的に学習される予測モデル f_t は t 本の決定木から構成され，$f(X) = T_1(X) + T_2(X) + \cdots + T_t(X)$ と書ける．ただし， T_j は 1 本の決定木を示し，実数値を返す．勾配ブースティング法は多クラス分類や回帰でも確立されているが，本研究では簡単のため 2 値分類 $y \in \{+1, -1\}$ のみを扱う．

勾配ブースティング法では，以下の式のように，損失を最小化する関数 (予測モデル) f^* を学習することを目指す．

$$f^* = \min_f \sum_{i=1}^{N} \Phi(y_i, f(X_i))$$

ただし，損失関数 Φ は逸脱度で，以下のように書ける．

$$\Phi(y, f(X)) = \ln(1 + \exp(-2yf))$$

勾配ブースティング法による学習方法を説明する．勾配ブースティング法では，順番に回帰問題を解くことで，加法的に弱学習器を追加していく．特徴ベクトル X_i のグラフ G_i に N 個に対して，残差 r_i を予測する回帰問題を， $(G_i, r_i)_N$ と表すことにする．最初の決定木は $r_i^{(1)} = y_i$ とし，与えられたラベルをそのまま回帰問題として解く．r の右上の括弧つき数字は，何本目の決定木のためのものかを示す．以降，t 本目の決定木は，各データの残差 $r_i^{(t)}$ を計算し， 回帰問題 $(G_i, r_i^{(t)})$ を解く．残差は損失関数の微分を用いて，以下のように計算する．

$$r_i^{(t)} = -\frac{\partial \Phi(y, f_{t-1}(X_i))}{\partial f} = \frac{2y}{e^{2yf} + 1}$$

データの元のラベル y_i と残差 r_i の符号は常に一致する．

3.2 決定木 [10]

決定木は非線形な予測モデルである．入力データの特徴ベクトル X に対して質問を積み重ね，予測した値 y を出力する．本稿では決定木を勾配ブースティング法の弱学習器として捉え，以降は残差 r を予測するものとする．

決定木学習は与えられた訓練データから，関係関数を表現する決定木を構築する手法である．決定木の構築は，各ノードにおいてデータ集合 D を特徴ベクトルの要素を使って D_1 と D_0 に分割する $(D = D_1 \cup D_0)$ ことで進められる．分割されたデータ集合 D_1, D_0 は，それぞれ子ノードへ送られる．子ノードは送られてきたデータ集合を分割する．このように，分割が再帰的に繰り返されて決定木が成長していく．弱学習器としての決定木では，データ集合のサイズが一定以下になったり，木の深さが一定に達した時点で分割をやめる．分割をやめたノードは葉ノードとし，そのノードのデータ集合の残差の平均値を予測値として付与する．

回帰のための決定木のデータ分割を考える．データ集合 D を分割する内部ノードは，2 乗誤差和を下げるように D_1, D_0 を計算する．具体的には以下を解く．

$$\min_{D_1, D_0} \left[\min_{c_1} \sum_{X_i \in D_1} \frac{1}{2}(r_i - c_1) + \min_{c_0} \sum_{X_i \in D_0} \frac{1}{2}(r_i - c_0) \right] = \min_{D_1, D_0} \Big[\mathrm{MSE}(D_1) + \mathrm{MSE}(D_0) \Big] \quad (1)$$

と書ける．

4. 提案手法

4.1 決定木のデータ分割におけるグラフ探索問題

提案手法では，部分グラフの有無を特徴ベクトルとして，3. 節で紹介した決定木の勾配ブースティング法を使う．部分グラフの有無を特徴ベクトルとする場合，訓練データのそれぞれのグラフが，どの部分グラフを持つか計算し，特徴ベクトルを構築する必要がある．しかし，訓練データのグラフのどれか 1 つ以上に生起する部分グラフの数が膨大になるため，全ての部分グラフの有無を調べるのは困難である．一つの解決法として，考える部分グラフを訓練データに頻出なものや辺数が一定以下のものに限定し，特徴ベクトルを構築する方法がある．これは，特徴ベクトルを予め計算するので，既存の様々な予測モデルを適応することができる反面，考える特徴が限定的になり重要な特徴を見逃してしまうリスクがある．もう一つの解決策として，全ての部分グラフのうち学習に有効だと思われる部分グラフの有無のみを計算する gBoost 法 [6] や Adaboost に基づく手法 [1] がある．これらの手法は重要な特徴を見逃すリスクが低いが，部分グラフの有無の線形モデルとなっている．そこで提案手法は，陽には計算しないが全ての部分グラフの有無を特徴として，非線形モデルである決定木の勾配ブースティング法を学習する．

全部分グラフ指示子の空間で決定木の勾配ブースティングを実現するためには，各弱学習器を与える決定木構築時に式 (1) を解く必要がある．部分グラフ指示子を特徴量とする場合，決定木の各ノードにおける質問は，各データのグラフ G_i が部分グラフ g_j を含むかどうか，となる．従って，2. 節の記法を用いて，式 (1) は

$$\min_{g} \left[\min_{c_1} \sum_{G_i \in D_1(g)} \frac{1}{2}(r_i - c_1) + \min_{c_0} \sum_{G_i \in D_0(g)} \frac{1}{2}(r_i - c_0) \right] = \min_{g} \Big[\mathrm{MSE}(D_1(g)) + \mathrm{MSE}(D_0(g)) \Big] \quad (2)$$

と書ける．つまり，決定木の構築時の各ノードでのデータ分割は，式 (2) の最小値を与える部分グラフ g の探索問題に帰着される．

4.2 探索木の子孫に対する 2 乗誤差和の下限

全ての可能な部分グラフについて値を調べれば式 (2) を解くことができるが，生起する部分グラフの全列挙は組み合わせ爆発により現実的な時間では通常不可能である．そこで，全ての部分グラフの有無を探索せずに，式 (2) を計算する方法が必要である．この説明のために，まずは，ある部分グラフ g でデータを分割したときの 2 乗誤差和が分かっているときに，別の $g' \supseteq g$ となる部分グラフ g' で分割したときの 2 乗誤差和の下限値 bound を考える．

訓練データ G_i について，$G_i \not\supseteq g \Rightarrow G_i \not\supseteq g'$ となることから以下のことが言える．データ集合 D を分割するとき，部分

グラフ g の有無で分割されたデータ集合を $D_1(g), D_0(g)$，部分グラフ g' の有無で分割されたデータ集合を $D_1(g'), D_0(g')$ とする．$D_0(g)$ の要素 (グラフ) は g を部分グラフに持たないので，g' も持たず，$D_0(g')$ に含まれる．$\tilde{D} \supseteq D_0(g)$ となる集合 $\tilde{D}$ を考えると，$D_1(g') = D_1(g) \setminus \tilde{D}$，$D_0(g') = D_0(g) \cup \tilde{D}$ となる．

以上より，グラフ g' でデータを分割したときの 2 乗誤差和の下限値 bound について

$$\begin{aligned}\text{bound} &= \min_{\tilde{D}} \left[\text{MSE}(D_1(g) \setminus \tilde{D}) + \text{MSE}(D_0(g) \cup \tilde{D})\right] \\ &\leq \min_{g'} \left[\text{MSE}(D_1(g')) + \text{MSE}(D_0(g'))\right]\end{aligned}$$

が言える．ただし，$\min_{\tilde{D}}$ は $\tilde{D} \subset D_0$ となる全ての集合を，$\min_{g'}$ は $g' \supseteq g$ となる全てのグラフを考えるものとする．

さらに，実際に bound を計算するときには $\tilde{D}$ は全ての集合を考える必要はなく，次のようにこくりつ的な計算が可能である．式 (2) において $c_1 > c_0$ のときは，$\tilde{D}$ に $D_1(g)$ のグラフのうち r_i が最小のものを入れ，2 乗誤差和を計算する．次は $D_1(g) \setminus \tilde{D}$ のうち r_i が最小のものを，さらに $\tilde{D}$ に追加し 2 乗誤差和を計算する．その次も同じく $D_1(g) \setminus \tilde{D}$ のうち r_i が最小のものを，さらに $\tilde{D}$ に追加する．$D_1(g) \setminus \tilde{D}$ の要素数が 0 になったら終了する．$c_1 < c_0$ の場合は最大のものを $\tilde{D}$ に入れればよい．

4.3 提案アルゴリズム

部分グラフの有無を調べる順番を工夫すると，効率的に式 (2) を解くことができる．データ中に生起する部分グラフは，木型の探索木によって効率的に調べられることが知られている [11]．本研究では，gSpan[11] アルゴリズムで用いられる探索木を使用する．探索木の例を図 1 に示す．探索木の各ノードはひとつの部分グラフを示す．ただし根ノードは便宜上設けたもので，グラフを示さない．あるグラフ g に辺を 1 つ追加して g' をとなるとき，g と g' の間に辺を引き，g を g' の親ノードとする．このとき，g' に既に親ノードがある場合は辺を引かないことにする．$g' \supseteq g$ の関係が成り立つ．

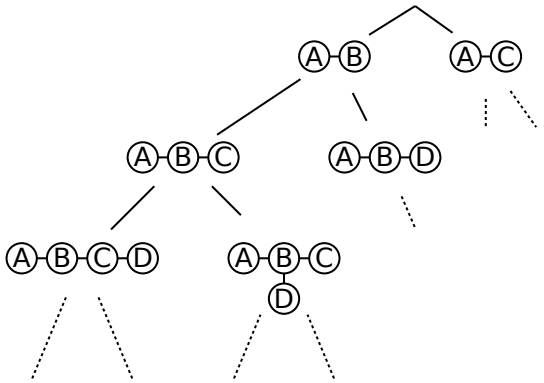

図 1: 部分グラフ列挙のための木

探索木を深さ優先で辿りながら部分グラフの有無を計算すると，効率的に枝刈りができる．提案手法の全体的なアルゴリズムを Algorithm1 に示す．各ノードでは，そのノードの示すグラフ g で分割したときの 2 乗誤差和を計算する．今まで計算してきた 2 乗誤差和の中で最小のものを tmp_best とし，これは随時更新する．$g' \supseteq g$ で分割したときの下限値 bound を計算し，tmp_best $\leq$ bound であれば，g を示すノードの子ノード全てを枝刈りする．深さ優先探索を続け，各ノードで同じ処理を繰り返す．

Algorithm 1 提案手法

```
function GradientBoosting(D)
    for 1, 2, ... do
        for i ← 1, N do
            r_i ← ∂Φ(y_i, f(X_i))/∂f
        end for
        MakeDecitionTree(D) で決定木 T を作る
        モデル f に 決定木 T を追加する
    end for
end function

function MakeDecitionTree(D)
    if 葉ノードになる基準を満たす then
        r_i の平均値を返す葉ノードにする
    else
        g ← FindBestSplit(D)
        g で分割する内部ノードにする
        MakeDecitionTree(D_1(g))
        MakeDecitionTree(D_0(g))
    end if
end function

function FindBestSplit(D)
    repeat
        g ← 列挙木の深さ優先で次のノードのグラフ
        value ← MSE(D_1(g) + MSE(D_0(g)))
        if value < tmp_best then
            tmp_best ← value
            tmp_best_g ← g
        end if
        bound ← min_D̃ [MSE(D_1(g) \ D̃) + MSE(D_0(g) ∪ D̃)]
        if tmp_best ≤ bound then
            列挙木の g の子ノードを全て枝刈り
        end if
    until 列挙木の探索が終わるまで
    return tmp_best_g
end function
```

5. 実験

2 つの典型的なベンチマークデータに対して提案手法で分類問題を解き，既存手法と正答率を比較する．10 分割交差検証を用い，正答率を評価する．

5.1 使用するデータ

本研究ではベンチマークとしてよく用いられている Mutag, CPDB を使用した．これらは，ある化合物が突然変異誘発性を持つかどうかという実験により得られたデータである．化合物の構造式を分子グラフで表現した．原子の情報 (炭素原子，水素原子など) をノードのラベルで，結合の情報 (単結合，二重結合など) をエッジのラベルで表現した．詳細を表 1 に示す．

5.2 モデルのパラメータ

提案手法には，決定木の数，決定木の深さ，分割時の最小データ数，分割をやめるデータ数，分割時の 2 乗誤差和の減

表 1: 使用したデータセット

	グラフ数	平均ノード数	平均エッジ数
Mutag	167	17.93	19.79
CPDB	601	13.77	14.22

表 2: 検証したパラメータ

	Mutag	CPDB
決定木の数	1 – 500	1 – 500
決定木の深さ	3, 5	3, 5
分割時の最小データ数	1, 10	1, 10
分割をやめるデータ数	0, 25	0, 25, 45
分割時の 2 乗誤差和の減少必要値	0, 0.01	0, 0.01
部分グラフの最大エッジ数	6, 8	6, 8

少必要値，部分グラフの最大エッジ数の計 6 つのパラメータがある．決定木の数は，勾配ブースティング法でアンサンブルに用いる決定木の数を決めるパラメータである．決定木の深さは，深さがその値になれば分割をやめる．決定木のノードに送られてくるデータの数を $|D|$，分割時のデータの数をそれぞれ $|D_1|, |D_0|$ とする．分割時の最小データ数が m のとき，$|D_1| < m$ または $|D_0| < m$ となる分割は採用しない．分割をやめるデータ数が n のとき，$|D|$ が n 以下ならば分割をやめる．分割時の 2 乗誤差和の減少必要値が e のとき，データ集合 D の値の 2 乗誤差を E，データ集合 D_1, D_0 の 2 乗誤差和 を E' として，$E - E' < e$ となる分割を採用しない．部分グラフの最大エッジ数 x のとき，特徴ベクトルのグラフ g のうち辺の数が x 以下のグラフのみ考える．これは図 1 の列挙木を深さ x までしか探索しないということである．提案手法では，これらのパラメータの組み合わせを表 4 の全てで試し，一番良い正答率を最終的な結果とした．

Random Forest (RF)[12] と XGBoost の実験結果も併記する．これらの結果は [7] による．Mutag の RF では，10 個以上の訓練データのグラフに現れる辺数 7 以下の部分グラフの有無を特徴とし，100 本の決定木を用いている．XGBoost では，15 個以上に現れる辺数 7 以下の部分グラフの有無を特徴としている．CPDB の RF では，1 個以上に現れる辺数 5 の部分グラフの有無を特徴とし，300 本の決定木を用いている．XGBoost では，1 個以上に現れる辺数 4 の部分グラフの有無を特徴としている．

6. 結果と考察

実験の結果得られた正答率を表 3 に載せる．両方のデータにおいて，提案手法が最良の正答率を出していることが分かる．

提案手法について，表 3 の正答率を出したパラメータを表 4 に示す．

探索した特徴数の平均値は，Mutag で 4853.8，CPDB で 6627.7 であった．

実験したデータの範囲では，提案手法の正答率が良いことが分かった．これは，全部分グラフを特徴としており重要な特徴を見つけられていること，部分グラフの非線形なモデルになっていることが原因だと考えられる．

今後は詳しい解析のために，実験データを増やす．また，部分グラフの有無を特徴とした場合，線形モデルと非線形モデルにどの程度の表現力の差があるのか考察を深める．

表 3: 正答率

	提案手法	gBoost[6]	RF	XGBoost[13]
Mutag	**0.914**	0.840	0.861	0.840
CPDB	**0.794**	0.778	0.788	0.775

表 4: 各データに対する最良パラメータ

	Mutag	CPDB
決定木の数	87	184
決定木の深さ	3	5
分割時の最小データ数	10	1
分割をやめるデータ数	25	25
分割時の 2 乗誤差和の減少必要値	0	0
部分グラフの最大エッジ数	8	6

7. 謝辞

本研究は JSPS 科研費 26330242, 16K13852 および JST さきがけの助成を受けたものです．

参考文献

[1] Kudo, T., Maeda, E., Matsumoto, Y.: An Application of Boosting to Graph Classification. *NIPS 2004*, pp. 729–736 (2004)

[2] Karklin, Y., Meraz, R.F., Holbrook, S.R.: Classification of Non-Coding RNA Using Graph Representations of Secondary Structure. *Pacific Symposium on Biocomputing*, pp. 5–16 (2005)

[3] Takigawa, I., Mamitsuka, H.: Graph Mining: Procedure, Application to Drug Discovery and Recent Advances. *Drug Discovery Today*, Vol. 18, No. 1–2, pp. 50–57 (2013)

[4] Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, Vol. 12, pp. 2539–2561 (2011)

[5] Rogers, D., Hahn, Dm: Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, Vol. 50, No. 5,
pp. 742–754 (2010)

[6] Saigo, H., Nowozin, S., Kadowaki, T., Kudo, T., Tsuda, K.: gBoost: A Mathematical Programming Approach to Graph Classification and Regression. *Machine Learning*, Vol. 75, No. 1, pp. 69–89 (2009)

[7] 越野 沙耶佳，岡崎 文哉，瀧川 一学: 定量的構造活性相関予測における化合物特徴表現の実験的検証 *2017* 年度 人工知能学会全国大会 *(*第 *31* 回*)*，2017 年 5 月 23 日–5 月 26 日

[8] Friedman, Jerome H. : Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232 (2001)

[9] Mason, Llew, et al: Boosting Algorithms as Gradient Descent *NIPS* (1999)

[10] Breiman, Leo, et al: Classification and regression trees. *CRC press* (1984)

[11] Yan, X., Han, J.: gSpan: Graph-Based Substructure Pattern Mining. *ICDM 2002*, pp. 721–724 (2002)

[12] Breiman, Leo.: Random forests. *Machine learning*, 45.1, pp. 5–32 (2001)

[13] Chen, Tianqi, and Carlos Guestrin : Xgboost: A scalable tree boosting system. *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM*, pp. 785–794 (2016)