

# 順序項木パターンに対する学習済GCNをオラクルとした 質問学習アルゴリズムのランダム化による精度向上

東山 的生<sup>1,a)</sup> 内田 智之<sup>1,b)</sup> 正代 隆義<sup>2,c)</sup> 松本 哲志<sup>3,d)</sup>

**概要:** 質問学習モデルは、常に正答を返す教師（オラクル）に質問を繰り返して教師の有する概念を同定する機械学習モデルである。小田ら（2022年）により、順序木データを学習させた高精度なグラフ畳み込みネットワーク（GCN）モデルを教師とし、GCNが学習した概念を表現する順序項木パターンを同定する質問学習アルゴリズムが提案された。本研究では、順序木データを学習させた2つのGCNモデル、R-GCNとGConvLSTMをそれぞれオラクルとした質問学習アルゴリズムを与え、それぞれが獲得した順序項木パターンの二値分類精度や可視化精度の違いについて考察する。さらに、ランダム化によって質問学習モデルの精度が向上したのでその詳細を報告する。

**キーワード:** 機械学習, データマイニング, グラフアルゴリズム, ニューラルネットワーク

## Enhancing accuracy through the randomization of a query learning algorithm using a pre-trained GCN as an oracle for term tree patterns

**Abstract:** The query learning model is a machine learning model that identifies the concept held by a teacher (oracle), by repeatedly asking queries to the oracle that always returns the correct answers. Oda et al. (2022) proposed a query learning algorithm that identifies the ordered term tree pattern that represents the concept learned by a high-accuracy Graph Convolutional Network (GCN) trained on ordered tree data, by using the GCN itself as an oracle. In this paper, the accuracies in binary classification problem and visualization problem are examined for the ordered term tree patterns, which are acquired by our query learning algorithm. The query learning algorithm uses two GCN models: R-GCN layer model and GConvLSTM layer model, and two query ordering: the left-first order and the random order. Finally, the details of how the accuracy of the query learning model is enhanced through randomization are discussed.

**Keywords:** Machine Learning, Data Mining, Graph Algorithm, Neural Network

### 1. はじめに

機械学習の手法の一つである深層学習は、人間の神経細胞の仕組みを模したニューラルネットワークを活用することで、大量のデータからデータに内在する特徴を見出し分

類することができる。また、明確なモデルを構築することなく複雑な構造をもつデータから学習できることが多く、画像処理や自然言語処理など多くの分野で活用されている。その一方で、予測根拠に関する透明性や説明性が低い、ブラックボックス型の学習モデルであることも知られている。

Angluin[1]により提唱された質問学習モデルは、学習者が常に正答を返す教師（オラクル）に質問を繰り返すことで、学習目標である表現を同定する学習手法である。小田ら [2] は順序木データを学習させた高精度 GCN(Graph Convolution Network) モデルをオラクルの代わりに用いることで、GCN モデルが学習した特徴表現である順序項木パターン [5] を獲得する質問学習アルゴリズム  $VA_{GCN}^{LearnOTT}$  (Visualization Algorithm based on LearnOTT using GCN)

<sup>1</sup> 広島市立大学  
Hiroshima City University, Asa-Minami Ku, Hiroshima 731-3194, Japan

<sup>2</sup> 福岡工業大学  
Fukuoka Institute of Technology

<sup>3</sup> 東海大学  
Tokai University

a) mi67013@e.hiroshima-cu.ac.jp

b) uchida@hiroshima-cu.ac.jp

c) shodai@fit.ac.jp

d) matsumoto@tsc.u-tokai.ac.jp

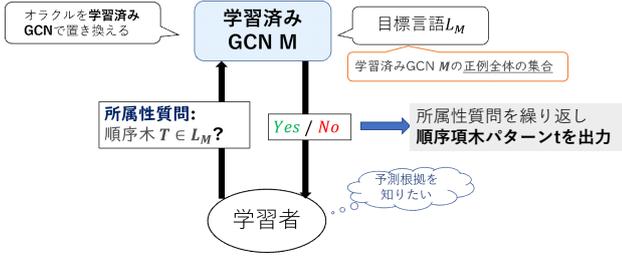


図 1 質問学習アルゴリズム  $VA_{GCN}^{LearnOTT}$  の概略図

を提案した. 図 1 に  $VA_{GCN}^{LearnOTT}$  アルゴリズムの概略図を示す. 本研究では, 正・負に二値分類させた順序木集合  $D$  を事前学習させた 2 つの高精度 GCN, GCN レイヤである R-GCNs(Relational Graph Convolutional Networks)[3] を用いた R-GCN モデルと GConvLSTM(Graph Convolutional Long Short-Term Memory)[4] レイヤを用いた GConvLSTM モデルをそれぞれ作成し, これら GCN モデルをそれぞれオラクルとした質問学習アルゴリズム,  $VA_{R-GCN}^{LearnOTT}$  と  $VA_{GconvLSTM}^{LearnOTT}$  を構築した. さらに, それぞれの質問学習アルゴリズムにランダム性を導入した質問学習アルゴリズム  $RVA_{R-GCN}^{LearnOTT}$  と  $RVA_{GconvLSTM}^{LearnOTT}$  を構築した. これら 4 つの質問学習アルゴリズムで出力された順序項木パターンを,  $D$  に関する二値分類精度である F 値について分析することで, GCN モデルの違いやランダム性の導入による精度向上について実証的に検討する. これにより, 事前学習させた高精度 GCN が有する概念を人が理解できるグラフパターン表現で可視化することに対して, 質問学習モデルが有用であることを示すことが本研究の目的である.

## 2. 準備

### 2.1 順序項木パターン

$\Lambda$  を有限アルファベット,  $\Lambda \cap \mathcal{X} = \emptyset$  を満たす  $\mathcal{X}$  を無限アルファベットとする. 根という特別な頂点を持ち, すべての子に順番がつけられている木を順序木という.  $V_t$  を頂点集合とし,  $E_t \subseteq V_t \times (\Lambda \cup \mathcal{X}) \times V_t$  を辺集合とする順序木  $t = (V_t, E_t)$  を順序項木パターンという. 辺  $e = (u, a, v) \in E_t$  における  $a \in \Lambda \cup \mathcal{X}$  を辺  $e$  の辺ラベルという. 特に  $a \in \mathcal{X}$  であるとき, 辺  $e$  を変数ラベルといい, 変数ラベルを持つ辺を変数という. 各変数ラベルが 2 回以上出現しない順序項木パターンを線形順序項木パターン(以降は単に順序項木パターンという)といい, 変数を持たない順序項木パターンを単に順序木という. 図 2 に, 例として順序項木パターン  $t$  と順序木  $T, T'$  を示す. 図において, 変数は変数ラベルを四角で囲って表し, 各内部頂点の長子から末子は順に左から右に描かれているものとする. 順序項木パターン全体の集合を  $OTT$ , 順序木全体の集合を  $OT$  で表す.

順序項木パターン  $t \in OTT$  と順序木  $T \in OT$  に対して,

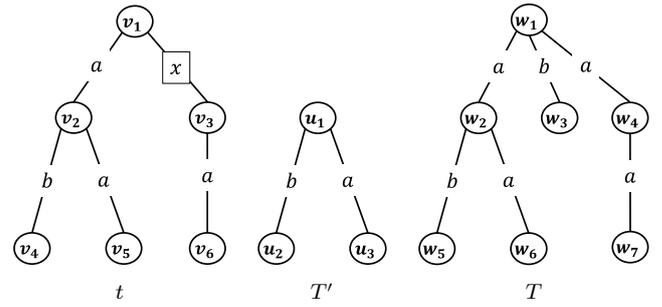


図 2 順序項木パターン  $t$  と順序木  $T, T'$

$t$  の各変数をそれぞれ適当な順序木で置き換えた結果,  $T$  と同型な順序木が得られる場合,  $t$  と  $T$  はマッチするという. 例えば, 図 2 の順序項木パターン  $t$  の頂点  $v_1, v_3$  と順序木  $T'$  の頂点  $u_1, u_3$  を同一視して  $t$  の変数 ( $v_1, x, v_3$ ) を置き換えることで順序木  $T$  と同型な順序木が得られる. このため,  $t$  と  $T$  はマッチする. 順序項木パターン  $t$  とマッチする順序木の集合を順序項木パターン言語といい,  $GL(t)$  と書く. また, 項木パターン言語全体の集合  $\{GL(p) \mid p \in OTT\}$  を  $OTPL$  と書く.

### 2.2 GCN モデル

グラフ構造データに対する深層学習モデルの一つに, グラフ畳み込みネットワーク GCN(Graph Convolutional Network)がある. GCN では, 頂点に特徴ベクトルを持つグラフを入力とし, 頂点間の関係を学習するために各頂点の特徴量を更新する. 目標頂点の特徴ベクトルを更新する際, 隣接頂点の特徴量を取り込む畳み込みを各層で行うことにより, 新たに割り振られた特徴量に隣接頂点の情報を反映することができる. 各層での特徴ベクトルの更新を経て, 最終的に得られた各頂点の特徴量を束ねることで得られる特徴量をもとにグラフの分類を行う. 畳み込み手法の違いでさまざまな GCN が提案されている. 本稿では, 10 層の R-GCNs 層から構築される GCN モデル(以後, R-GCN と略す)と, 3 層の GConvLSTM 層から構築される GCN モデル(以後, GConvLSTM と略す)の 2 つのモデルを構築し, 質問学習アルゴリズムのオラクルとして用いる. 次に R-GCN と GConvLSTM について説明する.

#### 2.2.1 R-GCN モデル

R-GCN 層では隣接頂点を畳み込む際, 辺ラベルにより定義される重み行列を隣接頂点の特徴ベクトルにかけ合わせる. これにより, グラフの辺ラベルも考慮して頂点の特徴量更新を行うことができる. R-GCNs 層の第  $(\ell + 1)$  層における出力が以下のように定義される.

$$h_i^{\ell+1} = \alpha \left( \sum_{r \in R} \sum_{j \in N_r^\ell} \frac{\ell}{c_{i,r}} W_r^\ell h_j^\ell + W_0^\ell h_i^\ell \right)$$

ここで,  $h_i^{\ell+1}$  は  $(\ell + 1)$  層目のノード  $i$  の特徴ベクトルを表す.  $\alpha$  は非線形の活性化関数を表す.  $R$  は辺のラベルの

集合を表し,  $N_i^r$  は頂点  $i$  に対して辺ラベル  $r$  で接続する頂点の集合を表す.  $h_i^l, h_j^l$  はそれぞれノード  $i$  の第  $l$  層目の特徴ベクトルと, 隣接ノードの特徴ベクトルを表している.  $c_{i,r}$  は正規化定数である.  $W_r^l, W_0^l$  は第  $l$  層の重み行列を表し,  $W_r^l$  は辺ラベル  $r$  により定義される.

### 2.2.2 GConvLSTM モデル

GConvLSTM 層は, グラフ構造データを扱う GCN とニューラルネットワークの一種である LSTM(Long Short-Term Memory) を組み合わせたレイヤである. LSTM は過去の層の情報を記憶し, 現在の特徴量更新の際にその情報を用いるため, データの長期依存関係を学習することができる. GCN はレイヤ数が増えるにつれて離れた頂点の情報を取り込んでいくため, 比較的隣接関係に関する情報が強い最初の層の情報が薄まって行ってしまう. しかし GConvLSTM は, LSTM の特性によりこの問題を緩和することができると考えられる. GConvLSTM 層の  $l$  層目の内部では, まず入力特徴量  $x^l$  に畳み込みを行う. R-GCNs と異なりこの畳み込み処理では辺ラベルの情報は考慮しない. その後, LSTM に入力し演算を通して出力を行う.  $(l+1)$  層でも同様に入力特徴量  $x^{l+1}$  に畳み込みを行って LSTM に入力するが, LSTM は入力  $x^{l+1}$  に加えて  $l$  層の演算結果を再び自分自身へ入力し, それをもとに演算を行う. これにより通常の畳み込みの演算に加えて, 過去の状態を考慮することが可能になる.

## 3. 質問学習アルゴリズム

### 3.1 順序項木パターンに対する質問学習アルゴリズム

Angluin[1] により提案された質問学習モデルでは, 学習アルゴリズムはオラクルに学習目標に対する質問を繰り返す行うことで, 学習者にとっては未知の学習目標を同定する. オラクルは学習者が獲得したい目標言語である  $GL_*$  を知っており,  $GL_*$  に対する質問に正しく答えることができる. 本稿では, 質問として, オラクルに順序項木  $T$  が学習目標であるグラフ言語  $GL_*$  に含まれるか否かを問う所属性質問を用いる. 2002 年に松本ら [6] は, 入力として与えられた  $n$  頂点の順序項木  $T \in GL_*$  を再帰的に変形しながら所属性質問をオラクルに繰り返し行うことで, 学習目標  $GL_*$  の表現である  $GL(t) = GL_*$  となる順序項木パターン  $t$  を同定する質問学習アルゴリズム LearnOTT を提案した.

### 3.2 GCN モデルをオラクルとする質問学習アルゴリズム

順序項木パターン言語  $L \in OTPL$  に対して  $L = GL(t)$  を満たす順序項木パターン  $t \in OTT$  を  $L$  の表現という.  $GL_* \in OTPL$  を学習目標とする. 順序項木  $T \in OT$  が  $T \in GL_*$  を満たすとき  $T$  を  $GL_*$  の正例といい, そうでないとき負例という.

小田ら [2] は, 質問学習アルゴリズム LearnOTT[6] のオラクルを, 事前学習として与えられた順序項木集合  $D$  を高精度で

## 質問学習アルゴリズム $VA_{GCN}^{LearnOTT}$

仮定: 学習済 GCN  $M$

入力:  $M$  が正と判断した定数個の順序項木の集合  $D$

出力: 順序項木パターン  $t$

- (1)  $C$  を空に初期化する
- (2)  $D$  の各順序項木  $T$  について以下の操作 (i),(ii),(iii) を行う
  - (i)  $T$  に最小正例発見処理と変数同定処理を行い順序項木パターン  $t_T$  を求める
  - (ii)  $t_T$  の F 値を求める
  - (iii)  $C$  に  $t_T$  とその F 値を追加する
- (3)  $C$  中の F 値が一番高い順序項木パターン  $t$  を出力する

二値分類できる GCN モデル  $M$  に変更し,  $D$  中の順序項木に対する,  $M$  による分類予測の根拠を表す順序項木パターンを出力する質問学習アルゴリズム  $VA_{GCN}^{LearnOTT}$  (Visualization Algorithm based on LearnOTT using GCN) を提案した. このアルゴリズムでは  $D$  を正例と負例の 2 種類に分類し, 正例集合の中の各順序項木に対して以下の 2 つの処理 (1),(2) を行う.

#### (1) 最小正例発見処理

入力する順序項木を  $T$  とする. 根を除く頂点とその親頂点との間の辺を縮約した木を  $T'$  とし, オラクルに対し所属性質問を行う. 所属性質問の返答が 'yes' のとき,  $T'$  を  $T$  とする. 所属性質問の返答が 'no' のとき, 縮約操作を取りやめ  $T$  のままとする. これらの操作を再帰的に行うことで辺数が最小の順序項木 (最小正例) を見つける.

#### (2) 変数同定処理

処理 (1) で得られた最小正例である順序項木  $t$  の各辺  $e$  をランダムに生成した 3 個の異なる順序項木で置き換えることで, 新しい 3 個の順序項木  $t_1, t_2, t_3$  を得る. 順序項木  $t_1, t_2, t_3$  の各順序項木が正例か否かを問う所属性質問をオラクルに対して行い, 全ての順序項木に対して返答が 'yes' のとき, 辺  $e$  を変数として記憶する. そうでないときは順序項木  $t$  のままとする. これらの操作を深さ優先走査の行きがけ順で行うことで変数となるべき辺を同定する. (1), (2) の処理後に, 学習目標  $GL_*$  の表現としての順序項木パターンを出力する. 入力順序項木  $T$  の頂点の数字は, 処理を行う順番 (深さ優先走査順) を表している. 1 番の頂点を削除した順序項木を学習済み GCN に対して所属性質問を行う. GCN が負例, つまり 'no' と返答しているため削除を取りやめる. 再び 2 番の頂点を削除し, 所属性質問を行う. 'yes' と返答しているため 2 番の頂点を削除する. その後は 3 番, 4 番と同様に最後の頂点まで処理を行う. 変数同定処理も同様に (1) で得られた順序項木に対して, 深さ優先走査の行きがけ順で全ての辺に対して処理を行う. GCN は常に正しい答えを返すとは限らない不完全なオラクルである. そのため, 正例の順序項木の集合を入力とすることで間違った出力結果となる可能性を抑える工夫が必要である. 小田ら [2] の質問

学習アルゴリズム  $VA_{GCN}^{LearnOTT}$  は、構築された順序項木パターンの集合のうち、一番多い回数構築された順序項木パターンを最終的に出力する。一方で、本実験では、構築された順序項木パターンのうち F 値が一番高いものを出力する。なお、F 値については次節で述べる。

## 4. 評価実験と考察

### 4.1 実験設定

質問学習アルゴリズム  $VA_{GCN}^{LearnOTT}$  の (2) の (i) の処理を深さ優先走査の行きがけ順で行うようにした、オラクルとして R-GCN モデルを用いた質問学習アルゴリズム  $VA_{R-GCN}^{LearnOTT}$  と GConvLSTM を用いた質問学習アルゴリズム  $VA_{GconvLSTM}^{LearnOTT}$  の 2 種類と、(2) の (i) の処理をランダムに行うようにした、オラクルとして R-GCN モデルを用いた質問学習アルゴリズム  $RVA_{R-GCN}^{LearnOTT}$  と GConvLSTM を用いた質問学習アルゴリズム  $RVA_{GconvLSTM}^{LearnOTT}$  の 2 種類、合わせて 4 種類の質問学習アルゴリズムをプログラミング言語 Python3.8.15 で実装した。実験では、OS が Windows11 の PC9 台を用いた。

学習データセット  $S_+ \cup S_-$  を学習した高精度 GCN モデル  $M$  と質問学習アルゴリズムにより得られた順序項木パターンの二値分類精度である F 値  $F_M$  および  $F_t$  を次のように定義する。  $M$  が分類した順序木のうち、正例と予想したものの集合を  $S_+^G$ 、負例と予想したものの集合を  $S_-^G$  で表すとき、  $F_M = 2 \cdot P_M \cdot R_M / (P_M + R_M)$  と定義する。ここで、  $P_M = |S_+ \cap S_+^G| / |S_+^G|$ 、  $R_M = |S_+ \cap S_+^G| / |S_+|$  である。同様に  $F_t$  は、  $S_+ \cap S_-$  のうち順序項木パターンとマッチする順序木集合を  $S_+^G$ 、マッチしない順序木集合を  $S_-^G$  としたときの  $F_M$  の値とする。  $P_M$  は Precision (適合率) であり、正例と予測したもののうち、どれだけ正解だったかを表している。  $R_M$  は Recall (再現率) であり、正例と予測したもののうち、どれだけ正例と予測できたかを表している。  $F_M$  は  $P_M$  と  $R_M$  の調和平均である。

### 4.2 人工データセット

実験で使用するデータセットの作成方法について説明する。まず、順序項木パターンをランダムに作成する。その後、順序項木パターンの変数部分をランダムに生成された順序木で置き換えて正例を作成する。順序木をランダムに生成し順序項木パターンとマッチしないものを負例とする。すべての順序項木パターンに対して上記の操作を行うことで二値分類されたデータセットを大量に作成した。

作成したグラフは順序木の簡潔データ表現の一つである、DFUDS (Depth-First Unary Degree Sequence) 形式で保存した。DFUDS は共通して開括弧 ‘(’ で始まり、深さ優先走査順で子を  $n$  個持つ頂点は、  $n$  個の ‘(’ と 1 つの閉括弧 ‘)’ で表現される。本実験では、開括弧を ‘0’、閉括

$$\begin{pmatrix} 0 & 1 & 3 & 0 \\ 1 & 1 & 2 & 1 \\ 1 & 2 & 0 & 2 \\ 1 & 3 & 1 & 1 \\ 2 & 1 & 0 & 2 \\ 2 & 2 & 0 & 1 \\ 2 & 3 & 0 & 1 \end{pmatrix}$$

図 3 図 2 の順序木  $T$  に対する初期特徴量

弧を辺ラベルに置き換えることで、辺ラベル付きの順序木に DFUDS を対応させた。また、頂点と辺ラベルを 1 対 1 で対応させるために、順序木の根の頂点に、親の頂点が存在すると仮定し、その頂点間を結ぶ辺のラベルを ‘#’ とする。

### 4.3 学習済み GCN の構築

データセットの順序木  $T$  の各ノードの初期特徴量  $H^0$  は、以下で定義される 4 次元のベクトル  $(h_1, h_2, h_3, h_4)$  で定義する。それぞれの特徴量を次に示す。

- $h_1$ :  $T$  における頂点の深さを表す非負整数 ( $h_1 \geq 0$ ) である。
- $h_2$ : 同じ頂点の深さにおける、頂点の順番を表す正整数 ( $h_2 \geq 1$ ) である。長子から末子に向かって番号が割り振られる。
- $h_3$ : 頂点の子供の数を表す非負整数 ( $h_3 \geq 0$ ) である。
- $h_4$ : 親ノードとの間の辺ラベルを表す正整数 ( $h_4 \geq 1$ ) である。1 は辺ラベル  $a$  を、2 は辺ラベル  $b$  を表す。ただし、頂点が根の場合は 0 とする。

図 2 の順序木  $T$  に対する初期特徴量  $H^0$  を図 3 に示す。

R-GCN モデルは、10 層の R-GCNs 層、読み込み層、全結合層により構築した。実験での、R-GCN モデル内部の処理の流れを図 4 に示す。入力される初期特徴量  $H^0$  は次元 (num\_node, 4) の行列であり、num\_node は順序木の頂点数を表す。まず R-GCN 層と活性化関数である ReLU 関数を用いて各頂点の特徴ベクトルの更新を行う。読み込み層では全ての頂点の特徴量をまとめて、グラフの特徴量を得る。ここでは、各特徴量の和をとる Add-pool を用いる。本実験では正・負の二値分類を行うため、全結合層である Linear で出力を 2 次元にする。最後に Softmax 関数を適用することで、モデルの出力ベクトルを得る。損失関数としては二値交差エントロピーを用いた。

GConvLSTM モデルは、3 層の GConvLSTM 層、読み込み層、全結合層により構築した。実験での、GConvLSTM モデル内部の処理の流れを図 5 に示す。R-GCN モデルと同様に、入力される初期特徴量  $H^0$  は行列であり、次元は (num\_node, 4) である。まず GConvLSTM 層と活性化関数である ReLU 関数を用いて各頂点の特徴ベクトルの更新を行う。読み込み層では各特徴量の平均をとる Add-mean を用いる。R-GCN モデル同様、全結合層である Linear で出力を 2 次元にした後、Softmax 関数を適用することでモデ

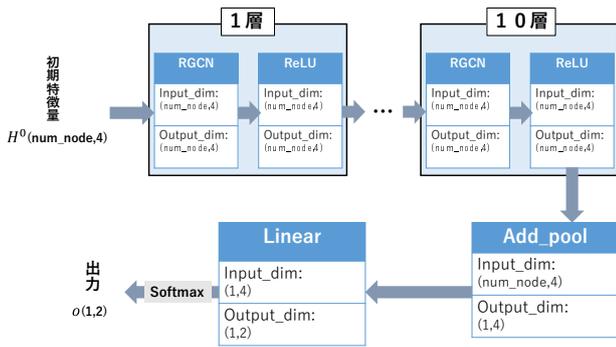


図 4 R-GCN モデルの内部処理の流れ

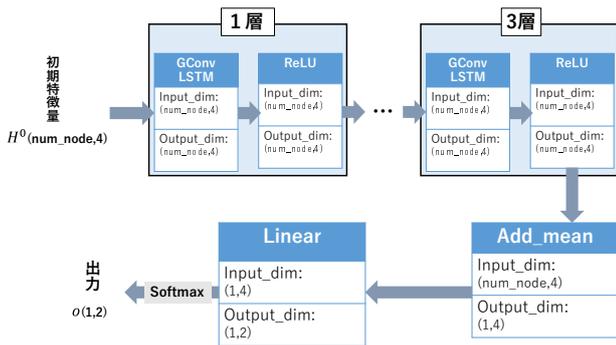


図 5 GconvLSTM モデルの内部処理の流れ

ルの出力ベクトルを得る。損失関数としては二値交差エントロピーを用いた。

#### 4.4 実験結果と考察

辺ラベル集合  $\Lambda = \{a, b\}$  について、4000 個の順序木の集合  $S_+$  と 5000 個の  $S_-$  の集合からなるデータセットを 1150 個人工的に作成した。各データセットは、 $S_+ \cup S_-$  を 5760 個の訓練データと 1440 個の検証データ、1800 個のテストデータに分けて GCN に学習させた。学習済み GCN の  $F_M$  は、ほぼすべてが 0.96 以上となった。図 6 に実行結果としてデータセット 490 を学習した GCN をオラクルとした、質問学習アルゴリズムの出力結果を示す。どちらも、データセットに対する  $F_t$  は 1.0 であるため、GCN の予測根拠を獲得できたといえる。各データセットに対して  $VA_{R-GCN}^{LearnOTT}$ ,  $VA_{GconvLSTM}^{LearnOTT}$ ,  $RVA_{R-GCN}^{LearnOTT}$ , および  $RVA_{GconvLSTM}^{LearnOTT}$  の 4 つの結果を取得し、得られたそれぞれの順序項木パターンの  $F_t$  をエクセルファイルに保存した。各アルゴリズムで得られたすべてのデータセットに対する結果から、 $F_t$  の分布をグラフにしたものを図 7 に示す。横軸は  $F_t$  を表しており、縦軸は横軸の数字以上の  $F_t$  を持つ順序項木パターンが全体の何%表れているかを表している。また  $F_t$  に対する確率密度関数を計算したグラフを図 8 に示す。質問学習アルゴリズムの違いにかかわらず、GConvLSTM のほうが R-GCN よりも  $F_t$  が大きい順序項木パターンを出力する割合が多くなった。R-GCN は辺ラベルを考慮することが可能だが、今回は辺ラベルが 2

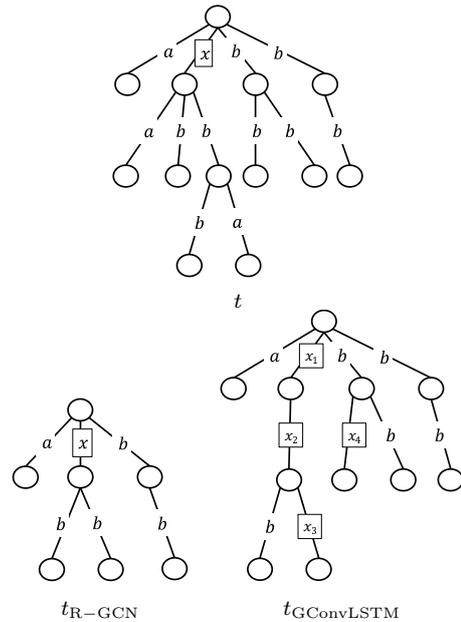


図 6 学習目標の表現である順序項木パターン  $t$  と得られた順序項木パターン  $t_{R-GCN}$  と  $t_{GconvLSTM}$

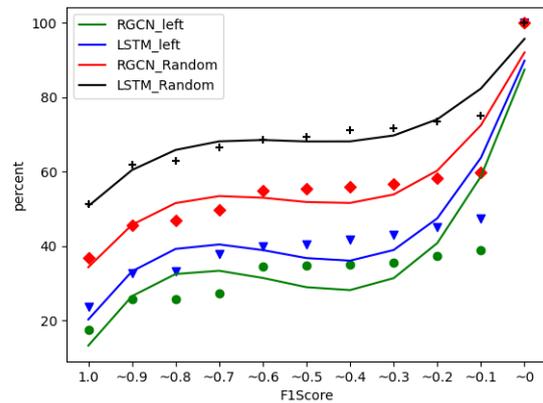


図 7 F 値の推移を示すグラフ

種類のデータセットを作成したため順序木の辺ラベルによる特徴の差があまり出ず、一方で GConvLSTM はグラフの遠くの情報を考慮することができ、かつ頂点間の隣接関係の情報をより深く学習することができるという特徴を持っているためこのような結果になったと推察される。またアルゴリズムの違いから比較すると、順序木への操作をランダムに行う  $RVA_{R-GCN}^{LearnOTT}$ ,  $RVA_{GconvLSTM}^{LearnOTT}$  がともに  $VA_{R-GCN}^{LearnOTT}$ ,  $VA_{GconvLSTM}^{LearnOTT}$  よりも高い精度となった。

#### 5. おわりに

本論文では順序木でモデル化できるグラフ構造データを 2 値分類させた高精度 GCN を構築し、それらをオラクルとした 2 種類の質問学習アルゴリズムにより GCN の予測根拠である順序項木パターンを獲得し、その F 値について分析する評価実験を行った。これにより質問学習アルゴリズムの有効性を示し、2 種類の GCN モデルの比較・考察を

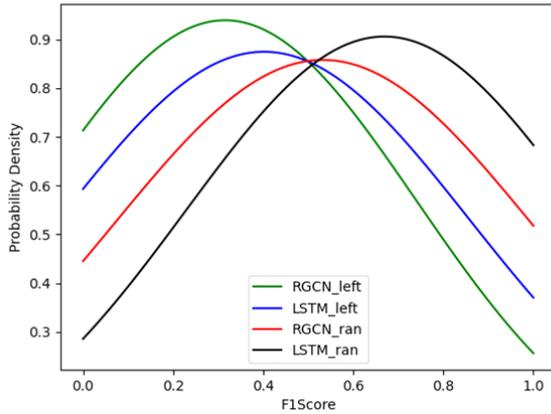


図 8 F 値に関する確率密度関数を示すグラフ

行った。さらに、質問学習アルゴリズム内での順序木に対する処理の順番を深さ優先走査の行きがけ順からランダムな順番に変更することで、獲得できる順序項木パターンの精度が向上することを明らかにした。今後の課題として、TTSP グラフなどほかのモデル化できるグラフ構造データに対する質問学習アルゴリズムの開発などが挙げられる。

**謝辞** 本研究は JSPS 科研費 JP19K12103, JP21K12021 の助成を受けたものです。

## 参考文献

- [1] Angluin, D.: *Queries and Concept Learning*, Machine Learning, 2(4):319-342 (1993)
- [2] 小田直季, 内田智之, 正代隆義, 松本哲志, 鈴木祐介, 宮原哲浩: 順序木パターンの質問学習アルゴリズムによるグラフ畳み込みネットワークの予測根拠の可視化, 人工知能学会全国大会論文集 (JSAI2022), p.2G4GS201 (2022)
- [3] Schlichtkrull, M., Thomas, N., Kipf, P., Welling, M., Berg, R. van den, Titov, I.: *Modeling Relational Data with Graph Convolutional Networks*, in Proc. ESWC 2018: The Semantic Web, pp.593-607 (2018)
- [4] Seo, Y., Defferrard, M., Vandergheynst, P., Bresson, X.: *Structured Sequence Modeling with Graph Convolutional Recurrent Networks*, Neural Information Processing, ICONIP 2018, pp.362-373 (2018)
- [5] Suzuki, Y., Shoudai, T., Uchida, T., Miyahara, T.: *Ordered Term Tree Languages Which are Polynomial Time Inductively Inferable from Positive Data*, Theoretical Computer Science, 350(1):63-90 (2006)
- [6] Matsumoto, S., Shoudai, T., Miyahara, T., Uchida, T.: *Learning Unions of Term Tree Languages using Queries*, in Proceeding of LA Summer Symposium, pp.21-1-21-10 (2002).