一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
IN2022-52(2023-01)

LSTMと条件付きVAEを用いた構造的特徴を高精度に指定可能なグラフ生成モデル

横山　昂大[†]　佐藤　良紀[†]　津川　翔[††]　渡部　康平[†]

† 長岡技術科学大学 大学院工学研究科　〒940–2188 新潟県長岡市上富岡町 1603–1
†† 筑波大学 システム情報系　〒305–8573 茨城県つくば市天王台 1–1–1
E-mail: †{s203193,s171039}@nagaokaut.ac.jp, ††s-tugawa@cs.tsukuba.ac.jp, †k_watabe@vos.nagaokaut.ac.jp

あらまし　近年，グラフを用いたアプリケーションの需要は高まっているが，実世界のグラフデータの利用はプライバシーなどの観点から難しい側面がある．そこで，既存のグラフ生成モデルは実世界のグラフの特徴を持ったデータセットの提供に役立つが，任意のグラフ特徴量だけを調整したグラフを生成することは難しい．我々は，この問題に対して与えられたグラフデータセットの特徴を模倣しつつ，特定の特徴量を連続的に調整したグラフを生成可能な生成モデル GraphTune を提案してきたが，特徴量の調整精度については十分な精度が得られていなかった．本稿では，GraphTune にグラフ特徴量の推定誤差をフィードバックする機構を追加し，交互に独立して学習させることにより劇的な精度向上を達成する手法を提案する．実グラフデータを用いた実験により，生成されるグラフの特徴量の分布が先鋭化し，高い精度で特徴量を調整可能であることを示した．

キーワード　ネットワーク，グラフ，条件付き VAE，LSTM，条件付き生成，機械学習

A Learning-Based Accurate Graph Reconstruction Model with Deep Learning

Takahiro YOKOYAMA[†], Yoshiki SATO[†], Sho TSUGAWA[††], and Kohei WATABE[†]

† Graduate School of Engineering, Nagaoka University of Technology
Kamitomiokamachi 1603–1, Nagaoka, Niigata 940–2188, Japan
†† Faculty of Engineering, Information and Systems, University of Tsukuba
1–1–1, Tennohdai, Tsukuba, Ibaraki 305–8573, Japan
E-mail: †{s203193,s171039}@nagaokaut.ac.jp, ††s-tugawa@cs.tsukuba.ac.jp, †k_watabe@vos.nagaokaut.ac.jp

Abstract　In recent years, there has been an increasing interest in developing models for learning and generating graphs using deep learning techniques. We developed graph generation model focusing on new aspects of graphs and compared it with baselines. The results showed that our proposed model outperforms the baseline on several indices.

Key words　Network, Graph, Conditional VAE, LSTM, Conditional generation, Machine learning

1. はじめに

近年，通信ネットワークやソーシャルネットワーク，データベース，化学などの幅広い分野でグラフデータを用いたアプリケーションは増加しており，ノード分類やコミュニティ検出，クエリ高速化といった例が挙げられる．それと同時に，グラフ上でのシミュレーションの重要性も高まっており，データマイニング分野において重要な情報を発見するためにシミュレーションを繰り返し行うことは，重要な情報を発見するための基本的なアプローチである．しかし，プライバシーなどの観点から研究者や専門家は必ずしも十分な実グラフデータを入手できるわけではない．この状況に対して，グラフ生成モデルは少数の実グラフデータを補完するためのアプローチになりうる．さらに，所有するデータセットには含まれない未知のグラフや未来のグラフを生成することで，新薬の分子構造の発見やネットワークの成長予測に用いることができる．

古典的には，あらかじめ定義された確率でエッジやノードを持つグラフを生成する確率モデルが研究されており，グラフの一面的な特徴のみを再現することに着目している．実グラフはときに特徴的な構造を備えていることがあり，ソーシャルネットワークでは，スケールフリー性，スモールワールド性，クラスタ性といった共通の性質が見られることが知られている．古典的な生成モデルは，これらの特徴をシンプルなアルゴリズムで再現することができるが，注目する特徴を除いたその他の特

徴を完全に再現できる保証がないため，多面的な特徴を備えた実グラフデータへの適応は難しい．

機械学習技術を用いたグラフ生成モデルは，グラフデータから直接的に特徴を学習し，その特徴にしたがってグラフを再現しようとするものである．ここ数年，学習ベースのグラフ生成は多くの研究者の注目を集めており，いくつかのアプローチが試みられている．深層学習モデルである Variational AutoEncoder (VAE) [1] やグラフを時系列のようにシーケンスデータとして捉えて学習する Long Short Term Memory (LSTM) [2] などを用いたモデルが存在しており，分子設計や引用グラフなどの多くのアプリケーションに適用可能な生成モデルが提案されている．しかし，既存の機械学習を用いたグラフ生成モデルは実世界のグラフによく似た特徴を持つグラフを生成できるが，それらの多くは，使用者が指定した特徴を持ったグラフの生成を目的の対象外としている．このようなデータセットの特徴を多面的に模倣しつつ，特定の特徴だけを変更するような柔軟な生成は応用上重要であるにも関わらず，あまり検討されていない．

我々はこれまで，連続的な特徴量の調整を可能にするグラフ生成モデル GraphTune [3], [4] を開発し，生成グラフの特徴量を指定値に依存して変化させることに成功したが，その調整能力については精度の点で課題が残されている．機械学習技術を用いた条件付きグラフ生成モデルである GraphTune [3], [4] は，学習や生成段階にデータセットに関連する条件ベクトルを与えることで多面的なデータセットの特徴を捉えながら，条件ベクトルに従った特徴量の調整を行うことが可能である．GraphTune は，Conditional VAE (CVAE) [1] と LSTM [2] を組み合わせることにより，シーケンスデータへ変換されたグラフとそれに関連する条件ベクトルを学習する．このモデルでは，生成されたグラフの特徴量の平均値を指定値に近い値に調整することは可能であるが，生成グラフによっては指定値から大きく外れた値となる場合があり，応用上，十分な調整精度を得ることができない．

本稿では，従来手法の GraphTune を拡張し，GraphTune が再構築したグラフから条件ベクトルに対応する特徴値を推定する Feature estimator を追加することで，その指定値と生成グラフの特徴量の誤差を生成精度の向上に繋げる手法を提案する．Feature estimator は生成グラフが実際にどのような特徴量になっているかを推定し，実際の指定値と推定される特徴量の誤差を計算することで，生成されたグラフが所望の特徴を有しているかを判定することができる．誤差の値は GraphTune にフィードバックされ，GraphTune の生成精度の向上に寄与する．GraphTune と Feature estimator を結合したニューラルネットワークは，通常通り学習するとリークを起こしてしまうため，GraphTune と Feature estimator を一方の重み更新を凍結させながら交互に学習する交互学習アルゴリズムにより学習を進める．本稿で提案するモデルは，Feature estimator により従来手法の多面的な特徴を継承しつつ，特定の特徴量をより高精度に指定することが可能である．

本稿の構成は以下の通りである．第 2 章では，本研究で取り扱うグラフ生成問題を定式化する．次に第 3 章において，提案手法の基盤となる GraphTune の詳細に触れ，第 4 章で，提案手法の構造や学習と生成の手順について説明する．第 5 章では，実グラフデータを GraphTune と提案手法でそれぞれ学習し，生成結果の精度を比較した評価結果を示す．最後に，第 6 章で本稿の結論を述べる．

2. 問題定式化

本稿では，連結である単純無向グラフを取り扱う．グラフ $G = (V, E)$ は，ノードの集合 $V \subseteq \{v_1, v_2, \ldots, v_n\}$ とエッジの集合 $E \subseteq \{(x, y) \mid x, y \in V\}$ で定義される．$\Omega = \{G_1, G_2, \ldots\}$ は，グラフ $G = (V, E)$ の全集合とする．本章では，グラフデータから構造的な特徴を低次元ベクトルとして抽出する写像とそのベクトルからグラフを再構築する逆像によってグラフ生成問題を定式化する．

写像 $F : \Omega \to \boldsymbol{A}$ を考えた時，グラフ $G_i \in \Omega$ は $F(G_i) = A_{G_i} = [\alpha^1_{G_i} \alpha^2_{G_i} \cdots]^T$ によって特徴ベクトル $A_{G_i} \in \boldsymbol{A}$ へ射影される．ある特徴ベクトル A_{G_i} の j 番目の要素 $\alpha^j_{G_i}$ は，グラフ G_i の特徴を表す．そして，全ての特徴は実数 $\alpha^k_{G_i} \in \mathbb{R}$ で表現される．例として，A_{G_i} はノード数やエッジ数，平均最短経路長，平均次数，エッジ密度，モジュラリティ，平均クラスタ係数，次数分布の冪指数，最大コンポーネントサイズといった特徴の値を表現する．

写像 $F(\cdot)$ の逆像を $F^{-1}(\cdot)$ と定義すると，グラフ G_i は逆像 $F^{-1}(A_{G_i}) = \{G_i, \ldots\}$ を計算することで得られる．我々はグラフの全集合 Ω の部分集合 $\mathcal{G}$ を使用して，$F^{-1}(\cdot)$ を高精度に近似する $\hat{F}^{-1}(\cdot)$ を求める推論問題に取り組む．この問題を解決することにより，ベクトル A_{G_i} の任意の要素を編集した特徴ベクトル $A_{G'_i}$ を正確に再現したグラフ G'_i を生成することが可能となる．

3. GraphTune

3.1 DFS コード

GraphTune はグラフをシーケンスデータである DFS コードへ変換する．そのアルゴリズムは，深さ優先探索によってグラフの全てのノードにタイムスタンプを与え，発見したエッジ $e = (u, v)$ を 5-tuple$(t_u, t_v, L(u), L(v), L(e))$ として，探索順に連ねたシーケンスを構築するものである．ここで，t_u はノード u のタイムスタンプ t_u を表し，$L(\cdot)$ はそれぞれエッジまたはノードのラベルを表している．ノードのラベル $L(u), L(v)$ にはそれぞれのノードの次数 $D(u), D(v)$, エッジのラベル $L(e)$ には 0 を割り当てている．

シーケンスの最後には，EOS(End Of Sequence) トークン $(\mathrm{EOS}_t, \mathrm{EOS}_t, \mathrm{EOS}_L, \mathrm{EOS}_L, 1)$ が追加される．ここで，EOS_t と EOS_L はグラフデータセット $\mathcal{G}$ の全ノードの集合 $\boldsymbol{V}$ に対して $1 + max_{v \in \boldsymbol{V}} t_v$ と $1 + max_{v \in \boldsymbol{V}} L(v)$ を表す．この EOS トークンにより，GraphTune は任意の大きさのグラフに対応している．

3.2 モデル化

GraphTune は図 1 に示す, LSTM ベースの Encoder と Decoder を持つ CVAE で構成される．学習時には，グラフデータセット $\mathcal{G}$ から DFS コードへ変換して得られるシーケンスデータの集

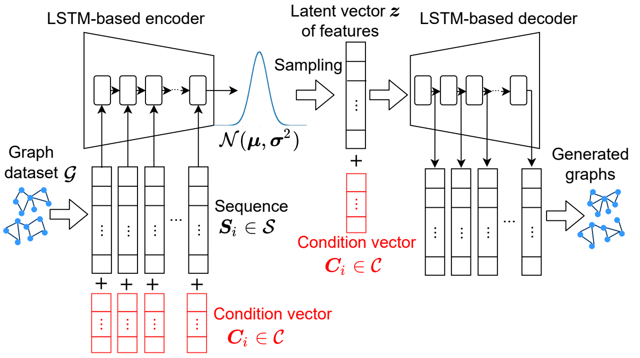

図 1　GraphTune モデル

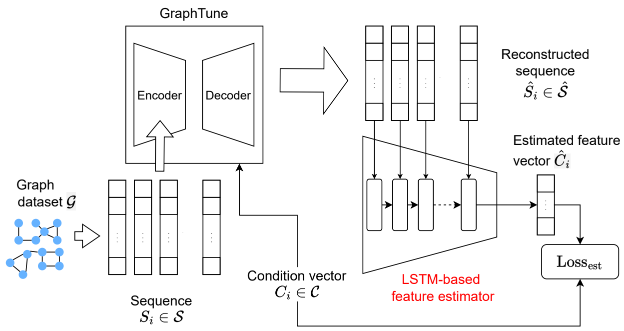

図 2　提案モデル

合 $\mathcal{S}$ と条件ベクトルの集合 $\mathcal{C}$ がモデルへ入力される．この条件ベクトルの集合 $\mathcal{C}$ は，グラフ構造の大域的な特徴量を計算することによって得られる．

Encoder $f_{\mathrm{enc}}(\boldsymbol{S}_i, \boldsymbol{C}_i)$ は，グラフの特徴に従ってシーケンス $\boldsymbol{S}_i$ を学習し，潜在ベクトル z に写像する．Encoder はシーケンスデータを扱うために積層 LSTM を使用しており，出力する潜在ベクトルの母集団には次元 L の多変量の標準正規分布を仮定している．GraphTune の Encoder の処理は以下の式で表せる．

$$z = f_{\mathrm{enc}}(\boldsymbol{S}_i, \boldsymbol{C}_i), \tag{1}$$

$$z \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}^2). \tag{2}$$

Decoder $f_{\mathrm{dec}}(z, \boldsymbol{C}_i)$ は，条件ベクトル $\boldsymbol{C}_i$ と潜在ベクトル z を学習し，入力を再構築したシーケンス $\hat{\boldsymbol{S}}_i$ に写像する．Decoder もまた，積層 LSTM を使用しており，その処理の概要は以下のようになる．

$$\hat{\boldsymbol{S}}_i = f_{\mathrm{dec}}(z, \boldsymbol{C}_i). \tag{3}$$

学習では, DFS コード $\boldsymbol{S}_i \in \mathcal{S}$ と条件ベクトル $\boldsymbol{C}_i \in \mathcal{C}$ をモデルへ入力し，再構築されたシーケンス $\hat{\boldsymbol{S}}_i \in \hat{\mathcal{S}}$ を得る．Encoder からは Kullback-Leibler divergence 損失 (KL 損失)$\mathrm{Loss}_{\mathrm{enc}}$，Decoder からは再構築損失 $\mathrm{Loss}_{\mathrm{dec}}$ が定義されるため，モデル全体の損失 $\mathrm{Loss}_{\mathrm{tune}}$ は次式で表される．

$$\mathrm{Loss}_{\mathrm{tune}} = \beta \cdot \mathrm{Loss}_{\mathrm{enc}} + \mathrm{Loss}_{\mathrm{dec}}. \tag{4}$$

モデル学習後の生成では，指定したい特徴ベクトル $\boldsymbol{C}$ と多変量の標準正規分布から無作為に抽出された潜在ベクトル z を Decoder へ入力する．すると，Decoder は $\boldsymbol{C}$ に従った特徴を持つグラフシーケンス $\hat{\mathcal{S}}$ を生成する．

4. 提　案　法

本稿では，GraphTune を拡張して生成されたグラフの特徴量の値を推定する Feature estimator と呼ばれる LSTM を追加し，その推定値と条件ベクトルの誤差を全体の損失に加えることで，より高精度な特徴量の指定を可能にする生成モデル及びその学習方法を提案する．提案モデルによるグラフ生成は，主に以下に示す 3 つの手順を繰り返すことで高精度に特徴量を調整したグラフ生成を実現する．

（1） グラフデータセットを学習し，GraphTune 部を訓練する．

（2） 学習済み GraphTune で条件付き生成を行う．

（3） GraphTune 部によって生成されたグラフを学習し，Feature estimator 部を訓練する．

本章ではまず，4.1 節で提案モデルの構成について説明し，次に，4.2 節で学習と生成の詳細を説明する．

4.1　モデル構成

提案モデルは，GraphTune と LSTM から成る Feature estimator で構成されており，図 2 のように表される．このモデルは，グラフセット $\mathcal{G}$ から変換されるシーケンスセット $\mathcal{S}$ とグラフの統計的な処理を通して得られる特徴ベクトル集合 $\mathcal{C}$ を用いて学習及び生成を行う．

前処理としては，グラフデータセット $\mathcal{G}$ をシーケンスセット $\mathcal{S}$ に変換する処理と条件ベクトル集合 $\mathcal{C}$ を計算する処理がある．前者は $\mathcal{G}$ を DFS コードによって $\mathcal{S}$ へ変換する処理に該当し，その詳細な手順は従来法である GraphTune と同様に行われる．後者では，$\mathcal{G}$ に対してグラフの統計的な処理によって算出された任意の特徴量を要素とする条件ベクトルの集合 $\mathcal{C}$ を取得する．

GraphTune 部は，3 章で説明した通り，Encoder f_{enc} によるグラフシーケンスの潜在空間 z へのエンコードと Decoder f_{dec} によるデコードの機能を担う．GraphTune 部については，学習時はグラフシーケンス $\boldsymbol{S}_i \in \mathcal{S}$ と条件ベクトル $\boldsymbol{C}_i \in \mathcal{C}$ が Encoder f_{enc} へ入力される．その後，f_{enc} の出力 z は条件ベクトル $\boldsymbol{C}_i$ と共に Decoder f_{dec} へ入力され，再構築されたグラフシーケンス $\hat{\boldsymbol{S}}_i \in \hat{\mathcal{S}}$ が出力される．生成時には，指定したい特徴量の値を含む条件ベクトル $\boldsymbol{C} \in \mathcal{C}$ と多変量の標準正規分布 $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}^2)$ から無作為に抽出された潜在ベクトル z を f_{dec} へ入力することで，新規のグラフシーケンスを生成する．以上を要約すると，提案モデルにおける GraphTune 部の処理は以下のようになる．

$$\hat{\boldsymbol{S}}_i = f_{\mathrm{dec}}(f_{\mathrm{enc}}(\boldsymbol{S}_i, \boldsymbol{C}_i), \boldsymbol{C}_i). \tag{5}$$

一方生成時は，

$$\hat{\mathcal{S}} = f_{\mathrm{dec}}(f_{\mathrm{enc}}(z, \boldsymbol{C}), \boldsymbol{C}). \tag{6}$$

ここで，$\hat{\mathcal{S}}$ は生成グラフのシーケンスである．

Feature estimator では，GraphTune 部により生成されたグラ

フの特徴量を推定する．生成されたグラフの特徴量をニューラルネットワークで推定させることで，生成されたグラフの特徴量と指定したい特徴量の値の差を計算し，後述の特徴量損失 $\mathrm{Loss}_{\mathrm{est}}$ によりフィードバックすることで，より高精度な調整能力を獲得することができる．Feature estimator は，LSTM とその前後に全結合層が配置された構成となっている．Feature estimator の入力となる GraphTune によって再構築されたシーケンス $\hat{\boldsymbol{S}}_i \in \hat{\mathcal{S}}$ は，いずれかの要素で EOS が出現した 5-tuple をシーケンスの終了とみなし，すべての要素を EOS に置き換える．加えて，EOS 以降の部分シーケンスをゼロベクトル $\mathbf{0}$ で置換する．その後，$\hat{\boldsymbol{S}}_i \in \hat{\mathcal{S}}$ の j 番目の要素 $\hat{\boldsymbol{s}}_j$ は，全結合層 f_{emb} を通して各 LSTM ブロック f_{est} へ入力される．LSTM について，初期の隠れ状態ベクトル $\boldsymbol{h}_0$ はゼロベクトル $\mathbf{0}$ で初期化されており，長さ $k = |\hat{\boldsymbol{S}}_i|$ のシーケンス $\hat{\boldsymbol{S}}_i$ に対して，LSTM ブロック f_{est} を隠れ状態ベクトル $\boldsymbol{h}_j$ に適用する．最後の LSTM ブロックの出力 $\boldsymbol{h}_k$ は全結合層 f_{feat} へ入力することで，推定された条件ベクトル $\hat{\boldsymbol{C}}_i$ を出力する．ここで，Feature estimator は学習時にのみ利用され，生成時には利用されないことに注意する．以上を要約すると，提案モデルにおける Feature estimator の処理は以下のようになる．

$$\boldsymbol{h}_0 = \mathbf{0}, \tag{7}$$

$$\boldsymbol{h}_{j+1} = f_{\mathrm{est}}(\boldsymbol{h}_j, f_{\mathrm{emb}}(\hat{\boldsymbol{s}}_j^T)) \quad (j = 0, 1, \ldots, k-1), \tag{8}$$

$$\hat{\boldsymbol{C}}_i = f_{\mathrm{feat}}(\boldsymbol{h}_k). \tag{9}$$

4.2 学習と生成

DFS コードによって与えられたシーケンスデータと統計的処理によって計算された特徴ベクトルを使用して，4.1 節で示した関数を提案モデルは学習する．学習過程において，シーケンスデータ $\boldsymbol{S}_i \in \mathcal{S}$ と条件ベクトル $\boldsymbol{C}_i \in \mathcal{C}$ が提案モデルに入力され，再構築されたシーケンス $\hat{\boldsymbol{S}}_i = \{\hat{\boldsymbol{s}}_j | j = 0, 1, \cdots, k\}$ 及び推定された条件ベクトル $\hat{\boldsymbol{C}}_i$ を得る．

提案法では，GraphTune で使われる再構築損失と KL 損失に加え，Feature estimator により計算される特徴量の情報をフィードバックするために特徴量損失 $\mathrm{Loss}_{\mathrm{est}}$ を導入する．提案法で使われる特徴量損失 $\mathrm{Loss}_{\mathrm{est}}$ は，次式で表される．

$$\mathrm{Loss}_{\mathrm{est}}(\hat{\boldsymbol{C}}_i, \boldsymbol{C}_i) = ||\hat{\boldsymbol{C}}_i - \boldsymbol{C}_i||_2^2. \tag{10}$$

$\mathrm{Loss}_{\mathrm{est}}$ は，条件ベクトル $\boldsymbol{C}_i$ と推定された条件ベクトル $\hat{\boldsymbol{C}}_i$ との L^2 ノルムであり，指定したい値と生成されたグラフの特徴量の誤差量を表している．そして，提案モデル全体における損失 $\mathrm{Loss}_{\mathrm{all}}$ は GraphTune 部の損失 $\mathrm{Loss}_{\mathrm{tune}}$ を用いて次式で表される．

$$\mathrm{Loss}_{\mathrm{all}} = \mathrm{Loss}_{\mathrm{tune}} + \gamma \cdot \mathrm{Loss}_{\mathrm{est}}(\hat{\boldsymbol{C}}_i, \boldsymbol{C}_i). \tag{11}$$

全体の損失 $\mathrm{Loss}_{\mathrm{all}}$ では，$\mathrm{Loss}_{\mathrm{est}}$ と重み係数 γ の積を $\mathrm{Loss}_{\mathrm{tune}}$ に加えることで，Feature estimator の損失のスケールを制御する．この損失 $\mathrm{Loss}_{\mathrm{all}}$ をモデルへ誤差逆伝播し，確率的勾配降下法によってモデルを最適化する．

提案法の学習は，1)GraphTune 部の学習，2)GraphTune によるグラフ生成，3)Feature estimator の学習の 3 つの段階に分かれており，その詳細を Algorithm 1 に示す．GraphTune 部の学習と Feature estimator の学習は同時に実施するとリークにより学習が適切に行えないため，3 つの段階を N 回繰り返すことで両者の学習を交互に実施する交互学習アルゴリズムを採用している (1～32 行目)．最初の段階では，GraphTune の学習を行う (2～13 行目)．モデル全体の損失はあらかじめ 0 で初期化され (3 行目)，GraphTune はすべてのシーケンスデータセット $\mathcal{S}$ を繰り返し処理する (4～11 行目)．GraphTune へ入力されたシーケンス $\boldsymbol{S}_i$ と条件ベクトル $\boldsymbol{C}_i$ は，Encoder f_{enc} を通して Decoder f_{dec} へ入力される．f_{dec} への入力には再び $\boldsymbol{C}_i$ が使用され，再構築されたシーケンス $\hat{\boldsymbol{S}}_i$ が出力される (5 行目)．N が 1 である場合には，GraphTune の損失 $\mathrm{Loss}_{\mathrm{tune}}$ のみを計算して損失に加えるが，それ以外の場合には，Feature estimator の損失 $\mathrm{Loss}_{\mathrm{est}}$ と重み γ の積も追加して計算し，損失に加える (6～10 行目)．GraphTune の各関数 f_{enc},f_{dec} は，損失を逆伝播することで更新される (12 行目)．このとき，f_{emb},f_{est},f_{feat} の重みは凍結し，更新しないことに注意する．次の段階では，学習された f_{dec} と条件ベクトル $\boldsymbol{C}$ を使用して条件付き生成を行う (14～19 行目)．n 個の新規のシーケンスを生成するために，繰り返し生成処理が

Algorithm 1 提案モデルの学習

Input: グラフデータセット $\mathcal{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots\}$,
条件ベクトル集合 $\mathcal{C} = \{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots\}$
Output: 学習済み関数 $f_{\mathrm{enc}}, f_{\mathrm{dec}}, f_{\mathrm{emb}}, f_{\mathrm{est}}, f_{\mathrm{feat}}$

1: **for** N times **do**
2: **repeat**
3: Loss $\leftarrow 0$
4: **for** i from 1 to $|\mathcal{S}|$ **do**
5: $\hat{\boldsymbol{S}}_i \leftarrow f_{\mathrm{dec}}(f_{\mathrm{enc}}(\boldsymbol{S}_i, \boldsymbol{C}_i), \boldsymbol{C}_i)$
6: **if** $N = 1$ **then**
7: Loss $\leftarrow$ Loss + $\mathrm{Loss}_{\mathrm{tune}}$
8: **else**
9: Loss $\leftarrow$ Loss + $\mathrm{Loss}_{\mathrm{tune}} + \gamma \cdot \mathrm{Loss}_{\mathrm{est}}$
10: **end if**
11: **end for**
12: Back-propagate Loss and upate weights to GraphTunef_{enc},f_{dec}
13: **until** stopping criteria
14: **for** $\boldsymbol{C}$ in $\mathcal{C}$ **do**
15: **for** i from 1 to n **do**
16: $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}^2)$
17: Add $f_{\mathrm{dec}}(\boldsymbol{z}, \boldsymbol{C})$ into $\mathcal{S}$
18: **end for**
19: **end for**
20: **repeat**
21: Loss $\leftarrow 0$
22: **for** i from 1 to $|\mathcal{S}|$ **do**
23: $\boldsymbol{h}_0 \leftarrow \mathbf{0}$
24: **for** j from 0 to $k-1$ **do**
25: $\boldsymbol{h}_{j+1} \leftarrow f_{\mathrm{est}}(\boldsymbol{h}_j, f_{\mathrm{emb}}(\boldsymbol{s}_j^T))$
26: **end for**
27: $\hat{\boldsymbol{C}}_i \leftarrow f_{\mathrm{feat}}(\boldsymbol{h}_k)$
28: Loss $\leftarrow$ Loss + $\mathrm{Loss}_{\mathrm{est}}(\hat{\boldsymbol{C}}_i, \boldsymbol{C}_i)$
29: **end for**
30: Back-propagate Loss and upate weights to Feature Estimatorf_{emb}, f_{est}, and f_{feat}
31: **until** stopping criteria
32: **end for**

行われる (15〜18 行目)．生成では，多変量の標準正規分布から無作為に潜在ベクトル z を抽出し (16 行目)，f_{dec} へ $\boldsymbol{C}$ と共に入力することで新規のシーケンスを獲得し，$\mathcal{S}$ へ追加する (17 行目)．最後の段階では，Feature estimator の学習を行う (20〜31 行目)．まず，損失を 0 で初期化し (21 行目)，すべてのシーケンスセット $\mathcal{S}$ を繰り返し処理する (22〜29 行目)．LSTM の初期の隠れ状態ベクトル $\boldsymbol{h}_0$ はゼロベクトル $\boldsymbol{0}$ で初期化される (23 行目)．各シーケンスは要素 $\boldsymbol{s}_j$ 毎に処理され，全結合層 f_{emb} を通して各 LSTM ブロック f_{est} へ再帰的に入力される (24〜26 行目)．最後の LSTM ブロックの出力 $\boldsymbol{h}_k$ は全結合層 f_{feat} へ入力され，出力として推定された条件ベクトル $\hat{\boldsymbol{C}}_i$ が得られる (27 行目)．$\boldsymbol{C}_i$ と $\hat{\boldsymbol{C}}_i$ を用いて計算された $\mathrm{Loss}_{\mathrm{est}}$ を損失に加え (28 行目)，それを逆伝播し，最適化することで各関数 f_{emb},f_{est},f_{feat} は更新される (30 行目)．このとき，f_{enc},f_{dec} の重みは凍結し，更新しないことに注意する．GraphTune と Feature estimator の学習は，既定の回数行われる (13,31 行目)．

学習後は，GraphTune と同様に，潜在ベクトル z を乱数として与えることで条件付き生成を行う．詳細な手順は Algorithm 1 で示した，2 つ目の段階の通りである (14〜19 行目)．

5. 実　　験

5.1 評価条件

提案法によるグラフ生成精度向上への効果を検証するため，実グラフを用いた学習と生成を行った．提案法のベースとなっている GraphTune と調整能力を比較することで，Feature estimator からのフィードバックによる調整能力の向上効果を定量的に評価する．

比較評価のためのデータセットには，GraphTune と同様に Higgs Twitter データセットにおけるフォロー・フォロワー関係のグラフ [5] からサンプリングされた部分グラフを使用した．フォロー・フォロワー関係のグラフからノード数が 50 となるようにランダムウォークによってノードがサンプリングされる．データセットに含まれる部分グラフは，その 50 個のノードにより与えられる誘導部分グラフである．データセットに含まれる誘導部分グラフの数は 2000 である．

提案法のモデルにおけるハイパーパラメータは以下のように設定する．Feature estimator の前に配置された全結合層 f_{emb} のサイズは 256，LSTM の隠れ状態ベクトルの次元は 512，最後に配置され全結合層 f_{feat} のサイズは 1 とした．GraphTune 関連のハイパーパラメータは，提案法の GraphTune 部分，及び比較手法としての GraphTune の両方において GraphTune の提案論文 [3], [4] で評価されたものと同じ値を使用した．最適化関数は Adam を使用しており，学習率は 0.001，weight decay は 0.0，勾配クリップの閾値は 1.0，係数 γ は 1000 とした．ミニバッチサイズは 37，Feature estimator 学習時のエポックは 10000 とした．また，GraphTune 部と Feature estimator 部の交互学習の反復回数 N は 2，Feature estimator 部の学習のために GraphTune 部で生成するグラフ数 n は 2000 とした．

条件ベクトルとしては，平均最短経路長を小数点以下 4 桁で四捨五入した値を学習時に与えた．平均最短経路長は，全てのノードペア間の最短経路長の平均であり，グラフ構造の大域的な特徴量である．生成時は，条件ベクトル $\boldsymbol{C}$ として 3，4，5 のそれぞれの値を指定し，検証を行った．

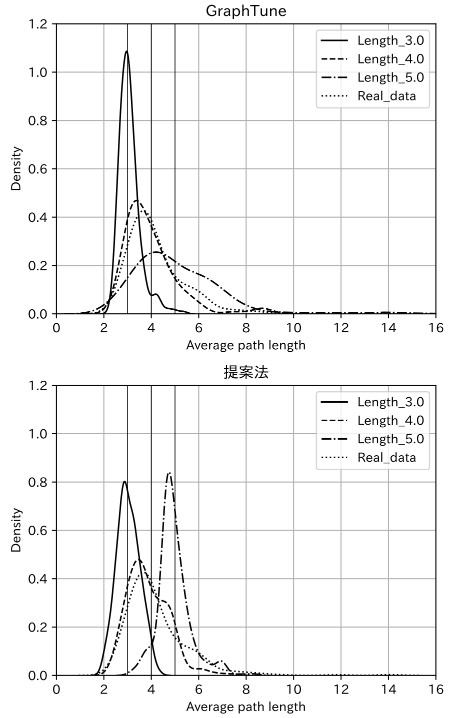

図 3　特徴量として平均最短経路長を指定した生成結果のカーネル密度推定プロット

生成精度の評価は，平均最短経路長の平均値，及び条件ベクトルで指定した値 $\boldsymbol{C}$ に対する 2 乗平均平方根誤差 (Root Mean Squared Error; RMSE) によって行う．前者は，生成されたグラフ集合の平均最短経路長をグラフ毎に計算し，平均を取ることによって算出される．後者は，生成されたサンプル集合の平均値 $\boldsymbol{C}_i$ と指定値 $\boldsymbol{C}$ により，次式で表される．

$$L^2 = \sum_{i=1}^{|\hat{S}|} \sqrt{(c_i - c)^2} \tag{12}$$

$$= ((\mathrm{E}[c_{\hat{S}} - c])^2 + \mathrm{Var}[c_{\hat{S}}])^{\frac{1}{2}} \tag{13}$$

ここで c_i 及び c は，生成グラフのシーケンス $\boldsymbol{S}_i$ から計算される特徴量 $\boldsymbol{C}_i$ と条件ベクトルにより指定された特徴量 $\boldsymbol{C}$ の成分の値を表す．これにより，生成されたサンプルの特徴量の分布の広がりを定量的に評価することが可能となる．

5.2 性能評価

本章では，学習時に与えた特徴量の値を指定することで，提案法は通常の GraphTune よりも生成精度が向上することを示す．提案法と GraphTune を比較することで，Feature estimator の追加による効果を前述の評価指標で定量的に評価する．

提案法と従来手法である GraphTune について，学習時に与えた特徴量である平均最短経路長を条件ベクトルを使用して 3,4,5 と 3 つの値を指定し，それぞれ 300 個のサンプルを生成した．図 3 に，生成されたサンプル集合における平均最短経路長のカーネル密度推定プロットを示す．図 3 の上の図は GraphTune,

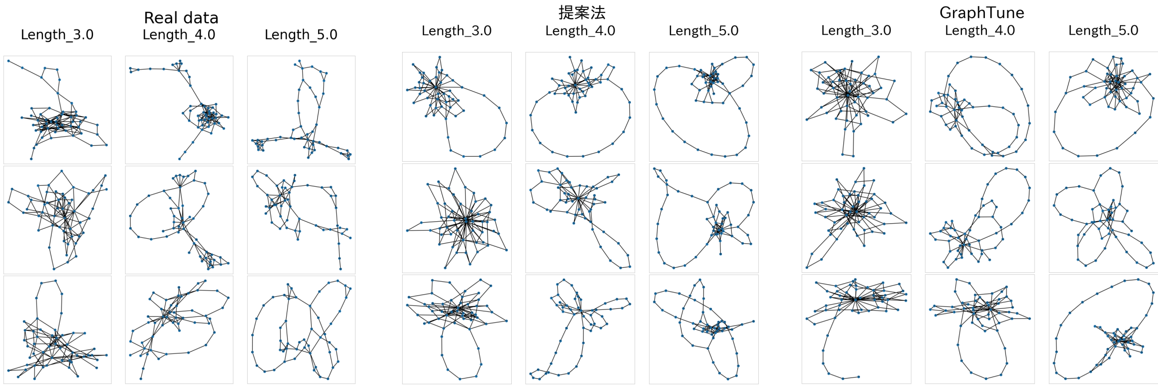

図 4　実グラフデータセット及び提案法と GraphTune によって生成されたグラフの一部

表 1　特徴量として平均最短経路長を指定した生成結果の定量的評価

指標	Condition	提案法	GraphTune	Real data
平均値	3.0	**3.03**	3.09	4.23
	4.0	3.86	**3.97**	
	5.0	**4.96**	5.06	
RMSE	3.0	0.476	**0.463**	–
	4.0	**0.861**	1.16	
	5.0	**0.705**	1.91	

下の図は提案法の結果であり，それぞれに条件付き生成されたグラフの平均経路長の分布と学習データセット $\mathcal{G}$ から 300 個無作為にサンプリングされたグラフの平均経路長の分布がある．

図 3 より，提案法と GraphTune はいずれも指定した値に依存して生成結果を変化させることに成功しているが，提案法の方が指定値ごとの分布がより明確にシフトしていることが確認できる．特に，平均経路長として 5.0 を指定した場合では，GraphTune は平均経路長の値が広い範囲に分布しているのに対して，提案法は指定値の近くに集中して分布している．このことは，表 1 に示す生成されたサンプル集合の平均値と RMSE の結果を見てもわかる．表 1 においては，最も良い結果を太字で強調しているが，全ての条件，指標において，提案法が GraphTune と同等もしくはより良い結果となっていることがわかる．特に，5.0 を指定した時の RMSE は，提案法は GraphTune よりも劇的に小さな値となっており，図 3 において，提案法の平均経路長の分布の広がりが小さくなっていることと対応して，調整精度が大きく改善していることがわかる．図 4 は，提案法と GraphTune によって生成されたグラフ及びデータセットの一部を可視化したものであり，提案法は実グラフに似た構造を保持しつつ，構造を変化させていることがわかる．これらのことから，提案した Feature estimator 及び交互学習は既存のグラフ生成モデルである GraphTune の特徴量の調整能力を強化するフレームワークであるといえる．

6. おわりに

本稿では，従来手法の GraphTune を拡張し，GraphTune が再構築したグラフから特徴量の値を推定する Feature estimator を追加することで，より高精度に特徴量を指定してグラフ生成可能な手法を開発した．提案手法では，Feature estimator が GraphTune により生成されたグラフの特徴量を推定し，指定したい値と推定値の差をフィードバックすることで高精度化を実現している．また，GraphTune 部と Feature estimator がリークすることなく協調して学習するための交互学習アルゴリズムを開発し，指定したい値と推定値の差を適切にフィードバック可能にした．

実グラフを用いた実験では，GraphTune と提案法における平均最短経路長を指定した生成を行い，提案法は GraphTune よりも高精度にグラフの特徴量を調整可能であることを示した．実験を通して，提案法により生成されたグラフの平均経路長の値は，指定値の周辺に集中して分布することを確認している．同時に，定量的な評価指標として特徴量の平均値と RMSE により，提案法は GraphTune よりも生成グラフの特徴量と指定したい値の差が小さくなることを確認した．

本稿での実験を踏まえた上で，提案法にはいくつかの展望がある．まず，平均最短経路長以外の特徴量を条件ベクトルとして採用し，生成精度が向上するか検証する必要がある．また，本稿では交互学習の反復回数を 2 としたが，これを増加させた場合の生成精度への効果を測定する必要がある．さらに，複数の特徴量の同時指定を検証することで生成モデルや生成可能なグラフの限界を探ることは，興味深い未探索領域である．

7. 謝　　辞

本研究の一部は科研費 20H04172 の助成を受けたものである．

文　　献

[1] D.P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” Proc. of the 2nd International Conference on Learning Representations (ICLR 2014), 2014.
[2] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol.9, no.8, pp.1735–1780, 1997.
[3] S. Nakazawa, Y. Sato, K. Nakagawa, S. Tsugawa, and K. Watabe, “A Tunable Model for Graph Generation Using LSTM and Conditional VAE,” Proc. of the 41st IEEE International Conference on Distributed Computing Systems (ICDCS 2021) Poster Track, 2021.
[4] S. Nakazawa, Y. Sato, S. Tsugawa, K. Nakagawa, and K. Watabe, “Graphtune: A learning-based graph generative model with tunable structural features,” CoRR, vol.abs/2201.11494, , 2022. https://arxiv.org/abs/2201.11494
[5] M.D. Domenico, A. Lima, P. Mougel, and M. Musolesi, “The Anatomy of a Scientific Rumor,” Scientific Reports, vol.3, no.2980, , 2013.