

動的グラフを対象としたリンク予測手法の動向調査

中西 宏和[†] 前川 政司[†] 佐々木 勇和[†] 鬼塚 真[†]

[†] 大阪大学大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

E-mail: †{nakanishi.hirokazu,maekawa.seiji,sasaki,onizuka}@ist.osaka-u.ac.jp

あらまし 動的グラフは多くの実世界における動的なシステムを表現することができる。動的グラフにおいて、リンクの時間的かつ構造的な近傍構造を捉えることは、将来のリンクの発生を予測するために必要不可欠である。近年、動的グラフにおける近傍構造を捉えることを目的とした手法が数多く提案されている。本論文では、動的グラフを対象としたリンク予測手法の包括的な概観を提供する。我々は、最先端のリンク予測手法を、代表的な Message Passing (MP) ベース, Walk Aggregation (WA) ベース, ハイブリッド (MP+WA) の3つのカテゴリに分類し、これらの手法の利点欠点を明らかにする。また、既存の研究では評価軸として表現力に注目しているものが多いが、より広い視野でこの分野の調査を行うために、本論文では効率性の評価軸を追加して評価する。最後に、この分野で今後解くべき課題と研究の方向性を示す。

キーワード 動的グラフ, グラフニューラルネットワーク, リンク予測

1 はじめに

動的グラフは現実世界の様々な複雑なシステムを抽象化することができるため、幅広く活用されている。動的グラフは相互作用する実体としてのノード、相互作用自体としてのリンク、相互作用がいつ起こるかを示すタイムスタンプを持つ。この動的グラフを対象とした重要なタスクに、リンク予測がある。これは、動的グラフが時間の経過とともに変化する性質を考慮し、将来のイベントの発生を予測するタスクである。例えば、SNS ネットワークにおける友人推薦 [1] や将来の消費者の購買予測 [2] などが応用事例として挙げられる。

近年、グラフデータを分析するためにグラフニューラルネットワーク (GNN) [3-5] が登場している。GNN は、注目ノードの近傍情報を集約することで、さまざまな応用事例に適用可能な embedding を生成することができる。しかし、GNN は静的なグラフにおいて大きな成功を収めたが、動的なグラフにおいては未開拓な部分が多い。それは、既存の静的グラフを対象とした GNN の手法を動的なグラフにそのまま適用することが効果的でないためである [6]。

最近、動的なグラフに対して効果的な手法が提案され始めている。まず、動的グラフをある一定の時間間隔で表現されるグラフのスナップショットとして扱うもの [7-9] がある。このようなグラフは離散時間動的グラフ (DTDG: Discrete Time Dynamic Graphs) と呼ばれ、このグラフを扱う手法では、動的グラフをある一定の時間間隔により表現するため、この時間間隔よりも細かい情報が欠落する。そのため、より先進的な方法に、相互作用する2ノードとそのタイムスタンプのシーケンスを動的グラフとして表現する方法がある。このようなグラフは、連続時間動的グラフ (CTDG: Continuous Time Dynamic Graphs) と呼ばれ、このグラフを扱う手法 [6, 10-17] では、属性情報のほか、時間的パターンと構造的パターンのどちらか、

もしくは両方を捉えることで高いリンク予測精度を達成することができる。このように CTDG を用いた動的リンク予測手法が最近広く注目を集めている。しかし、我々の知る限り、これら手法を包括的にまとめた研究は存在しない。

本論文では、近年の研究の動向を踏まえた動的グラフを対象とするリンク予測手法の包括的な概観を提供する。我々は、最先端のリンク予測手法を、リンク予測へのアプローチの方法により、Message Passing (MP) ベース [6, 10-13], Walk Aggregation (WA) ベース [14, 15], ハイブリッド (MP+WA) [16, 17] ベースの3つのカテゴリに分類する。MP ベースは、リンクの発生毎に近傍ノードへノード情報とその相互作用の情報を伝搬することで動的グラフにおけるグラフの時間的な変化を捉えたノード embedding を生成し、これを用いてリンクの発生傾向を捉える。WA ベースは、リンク予測対象の2ノードを始点とする多数ウォークから相対位置情報を抽出することで構造的パターンを捉えたリンク embedding を生成し、リンクの予測を行う方法である。ハイブリッドは、MP ベースと WA ベースの方法を組み合わせることにより、高精度化を達成している。これらカテゴリごとや各手法ごとに、それぞれの関係性や利点・欠点を明らかにする。MP ベースと WA ベースについて、それぞれ共通する基本アーキテクチャについてそれぞれ特定し、一般的な定式化を行う。さらに、表現力と効率性の2つの観点をベースに、複数の項目について分析することで、各手法が持つ特性を明らかにする。これらの分析を踏まえて、当該分野における現状の課題と今後の方向性について言及する。

本稿の構成は以下の通りである。2章で事前準備を行い、3章で各手法について詳細に述べた後、4章でそれら手法を2つの観点で分析する。また、5章では今後のこの分野における課題と方向性について述べる。最後に、6章で結論を述べる。

ベース	手法名	属性情報		時間的パターン		構造的パターン	処理速度	使用記憶領域
		低次近傍	高次近傍	周期	非周期			
MP ベース	TGAT [6]	✓		✓				✓
	TGN [10]	✓		✓	✓			
	APAN [11]	✓		✓	✓		△	
	TGL [12]	✓		✓	✓		○	✓
	TPGNN [13]	✓	✓	✓	✓		△	
WA ベース	CAW-N [14]	✓	✓			✓		✓
	NeurTWs [15]	✓	✓		✓	✓		✓
ハイブリッド	PINT [16]	✓		✓	✓	✓		
	NAT [17]	✓	✓	✓	✓	✓	○	

表 1: 各手法の分類と 2 つの観点による分析. 表現力は属性情報と時間的パターン, 構造的パターンの 3 つに分解し, 効率性は処理速度と使用記憶領域の 2 つに分解する. 属性情報は低次近傍と高次近傍に, 時間的パターンは周期と非周期にさらに分解し, それぞれ捉えられていれば ✓ をつけている. 処理速度については, 各論文内において実験により, 各ベース方法において最も基本的な手法である TGN [10] および CAW [14] に比べて処理速度の改善が確認でき, かつ, そのうち SOTA であるものには ○, SOTA ではないが改善が確認できるものには △をつけている. 使用記憶領域については, 記憶機構を用いていないもの, もしくは CSR 行列ベースのデータ構造を用いているものは使用記憶領域がローコストであると考えられるために ✓ をつけている.

2 事前準備

ここでは, 動的グラフやリンク予測, リンク予測に利用されるグラフ特徴の定義を行う.

2.1 動的グラフ

動的グラフ中でも本論文で扱う CTGD は, 相互作用する 2 ノードとそのタイムスタンプのシーケンスとして表現される. ノード集合を V , タイムスタンプ集合を T とすると, CTGD \mathcal{E} は以下の式により定義される.

$$\mathcal{E} = \{(u_1, v_1, t_1), (u_2, v_2, t_2), \dots\} \quad (1)$$

ただし, $u_i, v_i \in V$ は相互作用する 2 ノードを表し, $t_i \in T$ は各相互作用の発生時刻を表し, $t_1 \leq t_2 \leq \dots$ を満たす. また, データセットによっては各相互作用に属性情報がリンク属性として付与されている場合がある. 例えば, 掲示板ネットワークの Reddit データセット¹などは, 編集内容のテキストデータが 512 次元のベクトルに変換され, リンク属性として各相互作用に付与されている. この場合, リンク属性の集合を E とすると, 属性付き CTGD \mathcal{E}_e は以下の式により定義される.

$$\mathcal{E}_e = \{(u_1, v_1, t_1, e_1), (u_2, v_2, t_2, e_2), \dots\} \quad (2)$$

ただし, $e_i \in E$ である.

2.2 問題定義 (リンク予測)

動的リンク予測は, ある時刻 t より前の動的グラフ $\{(u', v', t') \in \mathcal{E} | t' < t\}$ から時刻 t に 2 ノード間にリンク (u, v, t) が発生するかどうかを予測するタスクである. このタスクを高精度かつ高効率で予測することが目的である. 以降, リンク予測対象である 2 ノードのことを注目ノード, リンクのことを注目リンクと表現する.

2.3 リンク予測に利用されるグラフ特徴

リンク予測タスクにおいて高い精度を実現するために, 属性情報と時間的パターン, 構造的パターンの 3 つの情報を CTGD から抽出し, embedding に反映する必要がある.

属性情報. Message Passing ベースの GNN は, 注目ノードの近傍の属性情報を集約することにより, 汎用的なノード embedding を生成する. ただし, これらの手法では 3-hop 以上離れた属性情報を集約しようとすると, 多くの場合 embedding が似通ってしまう過剰平滑が生じる [18]. ここで, 注目ノードから 2-hop 以内の属性情報を低次近傍属性情報, 3-hop 以上の属性情報を高次近傍属性情報と定義する. この高次近傍属性情報を活用するためには, 過剰平滑の課題を解決する必要がある.

時間的パターン. 動的グラフにおけるリンクの発生を長期的な視点で観察したときに見られる, ささまざまな時間的特性を時間的パターンと定義する. 本稿では, 時間的パターンを周期的パターンと非周期的パターンに分けて考える. 周期的パターンとは, ある 2 ノード間で周期的に相互作用が発生するパターンのことであり, E-メールのネットワークなどに頻繁に見られる. 非周期的パターンについては, ブラックフライデーには瞬時的に相互作用の発生頻度が全体的に増加するという瞬時性やある動画が投稿日から日数が経過したことでアクセス数が減衰する減衰性といったパターンを指す.

構造的パターン. ある制限された時間内に現れるリンクにより構成される連結サブグラフの類似パターンを構造的パターンと定義する. Message Passing ベースの手法では, 属性情報の集約時に平均の計算を行うために注目ノードに対する相対的な位置情報が失われ, 構造的パターンを抽出することができない [16]. 構造的パターンはリンク予測を行う上で重要な情報であるとされており [14], 後述の WA ベースの手法はこの情報を捉えることを目的に手法が提案されている. 構造的パターンの代表的な例として, 友人の友人は友人になりやすいという仮定を裏付ける Triadic Closure というサブグラフパターンなどがある.

¹: <http://snap.stanford.edu/jodie/reddit.csv>

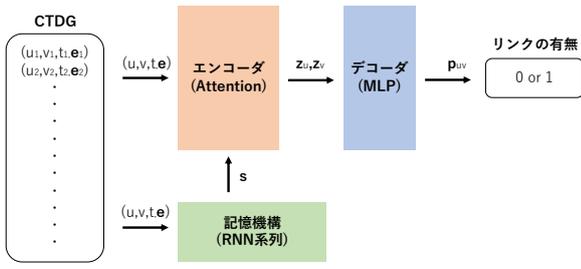


図 1: MP ベースのモデルアーキテクチャの概要図. リンクの発生に伴い記憶機構により状態が更新される, エンコーダでは, 注目リンクの情報とその時刻における状態が入力され, リンク予測対象の 2 ノードの embedding が出力される. デコーダでは, その embedding が入力され, 最終的なリンクの有無が出力される. なお, s は記憶機構が保持する状態のベクトル, z_u はノード u のノード embedding, p_{uv} はノード u, v 間のリンクの発生確率の予測値である.

3 手法の分類

本章では, CTDG を対象とする動的リンク予測手法を 3 つに分類し, それぞれのベースとなる考え方について述べた後, 各手法について詳細な説明を行う. なお, Neural Networks を用いた手法が支配的な精度を示しているため, 本章で紹介する手法は, Neural Networks を用いた手法に絞っている. 本稿で紹介する手法についての分類と表現能力と効率性の分析を表 1 に示す.

3.1 Message Passing ベース

概要. Message Passing (MP) ベースは, リンク発生毎に近傍ノードへ発生したリンクに関する情報を伝搬することで動的グラフにおけるグラフの時間的変化を捉えたノード embedding を生成する手法である. MP ベースは, 静的グラフにおいて成功している GCN [3] の考え方を動的グラフにも適用することで, ノード embedding を獲得することを目的としている. 相互作用に関する情報の伝播履歴を記憶しておくために, RNN などの系列モデルで構成される記憶機構を持ち, これにより時間的に変化する傾向をノード embedding に反映することができる. このノード embedding は, リンク予測タスクだけでなく, 動的クラス分類などのさまざまなタスクに活用可能である.

MP ベースは, 主に, 記憶機構, エンコーダ, デコーダの 3 つの構成要素からなり, その構成の概要は図 1 のように表される. 記憶機構は, 過去のリンク発生履歴を要約する各ノードの状態を保持する. エンコーダは, 注目 2 ノードの近傍情報と記憶機構の情報を集約することでノード embedding を生成する. デコーダはリンク予測対象の 2 ノードの embedding を MLP に入力することで最終的なリンクの有無を予測する.

記憶機構. 記憶機構は, 過去のリンク発生履歴を要約する各ノードの状態を保持する. リンクが発生すると, そのリンクに関わるノードの状態が系列モデル (RNN [19], GRU [20] など) により更新される. ある時刻 t にノード u, v 間に相互作用が発

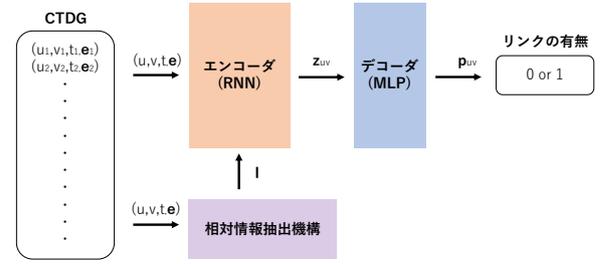


図 2: WA ベースのモデルアーキテクチャの概要図. リンクの発生に伴い相対情報抽出機構により構造的パターンを捉えた相対情報が算出される. エンコーダでは, 注目リンクの情報と相対情報が入力され, 注目する 2 ノードのリンクの embedding が出力される. デコーダでは, その embedding が入力され, 最終的なリンクの有無が出力される. なお, I は相対情報, z_{uv} はノード u, v からなるリンクの embedding, p_{uv} はノード u, v 間のリンクの発生確率の予測値である.

生した場合, u と v に関する次のメッセージ $m_u(t), m_v(t)$ が生成される.

$$m_{uv}(t) = (s_u(t^-) || s_v(t^-)) \Phi(t - t^-) || e_{uvt} \quad (3)$$

$$m_{vu}(t) = (s_v(t^-) || s_u(t^-)) \Phi(t - t^-) || e_{uvt} \quad (4)$$

ここで, $||$ は concatenate を表し, $s_v(t)$ はノード v の時刻 t における状態である. $s_v 1(t^-)$ は時刻 t の直前の状態であり, 初期値 $s_v(0)$ にはノード属性かランダム値を用いる². $\Phi(\cdot)$ は過去の相互作用と現在の相互作用の時間差を入力とする関数であり, identity function もしくは, Xu ら [6] が提案したタイムエンコーディング関数を用いられる. e_{uvt} は時刻 t においてノード u, v 間で発生した相互作用のリンク属性を表す.

時刻 t に同時に発生したをリンクの集合のうち, リンクにノード v を含んでいるものを $\mathcal{J}(v, t)$ と表す. リンクの発生時に, そのリンクに関わるノードの状態を系列モデルにより更新するために, メッセージの集約が次の式により行われる.

$$\bar{m}_v(t) = \text{MESAGG}_{(v, u, t) \in \mathcal{J}(v, t)}(m_{vu}(t)) \quad (5)$$

MESAGG 関数は集約関数であり, 平均が広く用いられる. ノード v の状態は, 直前の自身の状態と生成した集約後のメッセージを用いて次の式により更新される.

$$s_v(t) = \text{UPDT}(s_v(t^-), \bar{m}_v(t)) \quad (6)$$

UPDT 関数には RNN や GRU などの系列モデルが用いられる.

エンコーダ. エンコーダは, 注目する 2 ノードに対して, その近傍ノードの情報を集約することで, 各ノード embedding を生成する. 時刻 t におけるノード v の時間的近傍 $\mathcal{N}(v, t) = \{(u, e_{uvt'}, t') | \exists (u, v, t') \in \mathcal{E}, t' < t\}$ という近傍ノードと, そのノードとの相互作用に付与されているリンク属性, その発生時刻の 3 つにより構成されるものの集合と定義する.

²: ノード属性が与えられないデータセットの場合は, ランダム値をノード属性として利用する.

記憶機構	手法	保持内容	更新方法	高次近傍の抽出	近傍問い合わせの有無
メモリ	[6, 10, 16]	自身のノード状態	RNN		✓
メールボックス	[11]	自身のノード状態	FIFO		
k -hop メモリ	[13]	k -hop 毎の自身ノード状態	RNN	✓	
k -hop 辞書メモリ	[17]	自身のノードと k -hop 近傍の状態	RNN と FIFO		✓

表 2: 各記憶機構についての比較表. 保持内容とは, 各記憶機構が各ノードごとに保持する内容のことを指す. 更新方法は保持内容をどのように更新するかを表し, そのうち FIFO は古い情報から優先的に削除する方法である. 近傍問い合わせの有無はエンコーダにおいて近傍ノードの情報を集約するために, 近傍問い合わせをするか否かを表す.

時刻 t におけるノード v の embedding $z_v(t)$ は次のように定式化される.

$$z_v(t) = \text{AGG}_{\text{MP}}_{(i, e_{ijt}, t') \in (v, \text{e}_{uv}, t) \cup \mathcal{N}(v, t)}(s_i(t'), e_{ijt}, t - t') \quad (7)$$

AGG_{MP} は近傍ノードの情報を集約する学習可能なパラメータを持つ関数であり, Attention や MLP など構成される. この近傍ノードの情報は 1-hop 先の属性情報と時間的情報を含む. 手法によっては, 集約時の近傍ノードの特定にかかる時間を削減するために, エンコーダにおける近傍集約を行わないものもある [11–13, 17]. 代わりに自身の記憶機構に k -hop 先の状態も保持する. これにより自身の記憶を参照するだけで近傍集約を行い, 処理速度の向上を達成している.

デコーダ. デコーダはリンク予測対象の 2 ノードの embedding を MLP に入力することでリンクの有無を予測する. ノード u, v の 2 ノードの間において時刻 t にリンクが発生する確率 p_{uv} は, 以下のように定式化される.

$$p_{uv}(t) = \text{MLP}(z_u(t), z_v(t)) \quad (8)$$

MP ベースの利点としては, 記憶機構やタイムエンコーディング関数の活用により時間的パターンの抽出が可能であることが挙げられる. 欠点としては, Message Passing だけでは, リンク予測において重要な構造的パターンを識別することが難しいことが挙げられる. この課題は, ハイブリッド (MP+WA) の手法 [16, 17] において取り組まれている. また, 複数 hop 先まで伝搬するために層数を増加させると, 2つの問題が生じる. 一つ目は, 過剰平滑であり, グラフ内の全てのノードがマルチホップ近傍を共有することにより, 学習された全てのノード embedding が類似してしまうという問題である. この課題は, [13] において取り組まれている. 二つ目は伝搬するノード数が hop 数の増加とともに指数関数的に増加するため, 計算量的が大きくなるという課題である. この課題は, [17] において取り組まれている. また, 大規模データセットの場合, そのノード数の多さから, メモリを保持するために莫大な記憶容量を要求する.

各手法. TGAT [6] は, 学習時に未登場のノード表現を学習することが難しいという従来の手法の課題を解決するために, 静的グラフにおいて成功を取めた Graph Attention (GAT) を動的グラフに対しても適用できるように拡張した手法である. また, 古典的調和解析の Bochner の定理を活用したタイムエンコーディングを用いることで, 時間的な情報を MP に組み込む

ことを提案した最初の手法である. この手法では, 記憶機構は使われておらず, 豊富な時間的パターンを捉えられない欠点がある.

TGN [10] は, 過去の相互作用の履歴を考慮するために, 各ノードの状態をメモリに保持することで, より豊富な時間パターンを捉えた embedding を獲得することを可能にした. メモリとは, 各ノードごとに過去の相互作用の履歴を要約する状態をベクトルとして保持する, 最も基本的な記憶機構である. また, JODIE [21] や DyRep [22], TGAT などの TGN 以前の CTDG を対象とした手法を一般化することができるフレームワークとなっている. 最近, PyTorch Geometric において TGN の並列計算をより高速化した方法が公開されており, より大規模なグラフに対しても効率的に計算が行える. TGN の課題としては, 層数を増やすと, 過剰平滑の問題が生じ, 高次情報を embedding に反映できないことや, エンコーダの集約時に問い合わせるノードの数が指数関数的に増大することで計算効率が悪いという点が挙げられる.

APAN [11] は, CTDG による手法の実世界応用を目的に, 実行速度の改善を行なった手法である. TGAT や TGN では, エンコーダでのノード embedding の生成の際に近傍ノードに問い合わせをする必要があり, ここが計算時間のボトルネックとなっている. これを解決するために APAN では, 各ノードではメモリの代わりにメールボックスという固定長の k -hop 近傍のメッセージを保持する記憶機構を用いてノード embedding を生成する. これにより, ノード embedding の生成の際に近傍ノードに問い合わせをする必要がなくなり, 実行時間を大幅に短縮している. メールボックスでは, 新しいメッセージを受け取ると, 最も古いメッセージを消去する FIFO の方式がとられている. このため, リンク発生が頻繁なノードは, 長期的な時間パターンを捉えられないことが課題として考えられる.

TGL [12] は, 大規模グラフに適用可能かつ, TGN よりも多くの手法を一般化した汎用フレームワークである. 近傍ノードへの高速な問い合わせを可能にする CSR 行列ベースのデータ構造と, 様々な近傍探索アルゴリズムをサポートする並列実行可能なサンプラーにより, 計算時間を大幅に削減することに成功している. また, 記憶機構を有する手法においてバッチサイズが大きい場合に生じるバッチ内の時間的な依存関係が失われる問題を解決するために, ランダムチャンクスケジューリングアルゴリズムを提案している.

TPGNN [13] は, 伝搬されるメッセージの hop 数ごとに記憶状態を分けて保持し, Transformer [23] を用いて重要な hop

数を特定することで、TGNにおける課題である高次情報の抽出と過剰平滑の解決を達成している手法である。APANと同様に、エンコーダでの embedding の生成時には、自身のノードが保持する記憶ベクトルのみ用いるため、近傍ノードに問い合わせをする必要がなく、計算効率も向上している。ただし、各ノードごとに記憶状態を伝搬する hop 数分の状態を保持する必要があるため、ノード数の多い大規模なグラフを対象とする場合には空間的計算量が課題となると考えられる。

3.2 Walk Aggregation ベース

概要. Walk Aggregation(WA) ベースは、注目ノードを始点とする時間的な制限を設けた多数ウォークから相対位置情報を抽出し、RNN に入力することで構造的パターンを捉えたリンク embedding を生成し、リンクの予測を行う方法である。このウォークは Wang ら [14] によって提案された Causal Anonymous Walks (CAW) と呼ばれるウォークをベースとしている。CAW は時間的因果関係の条件の下、最近のリンク発生を優先的にサンプリングすることで生成されるウォークであり、動的グラフにおける構造的パターンを捉えることができる。MP ベースが CTDG における汎用的なノード embedding の生成を目的としている一方で、WA ベースではリンク予測タスクの精度の向上を目的としている。そのため、リンク予測タスクを解く上で重要となる構造的パターンを捉えることに主眼を置いている。WA ベースは、主に、相対情報抽出機構、エンコーダ、デコーダの3つの構成要素からなり、その構成は図2のように表される。相対情報抽出機構は、注目ノードを始点とする多数ウォークを用いて、注目ノードに対する各近傍ノードの相対位置情報を生成する。

相対情報抽出機構. 相対情報抽出機構は、注目ノードを始点とする多数ウォークを用いて、注目ノードに対する各近傍ノードの相対位置情報を生成する。ここでは、代表的な手法である CAW-N において提案されている CAW について定式化する。まず、CTDG における長さ L の一般的なウォークを $W = ((w_0, t_0), (w_1, t_1), (w_2, t_2), \dots, (w_L, t_L))$ 、と定義する。ただし、 $t_0 > t_1 > \dots > t_L$ かつ $(w_{i-1}, w_i, t_i) \in \mathcal{E}, 1 \leq i \leq L$ とする。時刻 t_0 においてノード u_0 と v_0 にリンクが発生したとき、 u_0 と v_0 を始点とする長さ m のウォークをある時間的因果関係の条件の下、それぞれ M 個生成し、その集合を S_{u_0}, S_{v_0} とする。この時間的制約とは、最近の相互作用ほど重要であるという特徴を捉えるため、 $\exp(-\alpha(t - t_p))$ に比例する確率で次の遷移先ノードを決定する。ただし、 α は正の定数であり、 t は現在注目しているリンクの発生時刻、 t_p は時刻 t 以前の遷移候補リンクの発生時刻を表す。このウォークの集合を用いて、 $S_{u_0} \cup S_{v_0}$ に少なくとも一度は登場したノード w に対して、 $I_{CAW}(w; \{S_{u_0}, S_{v_0}\})$ を生成する。これは、匿名化された相対位置情報を表すベクトルであり、注目2ノードのペアの形式とすることにより、注目2ノードの相関を捉えることができる。以下にその定義式を示す。

$$I_{CAW}(w; \{S_{u_0}, S_{v_0}\}) = \{g(w; S_{u_0}), g(w; S_{v_0})\} \quad (9)$$

$$g(w; S_u)[i] = |\{W | W \in S_{u_0} : w = W[i][0]\}|, i \in \{0, 1, \dots, m\} \quad (10)$$

$g(w; S_{u_0})[i]$ はウォーク集合 S_{u_0} の内、ノード w が i 番目の位置するウォークの数をカウントしたものとされており、これにより動的グラフの構造的情報を捉えることができる。

こうして得られた I_{CAW} をウォークのノード ID と置き換えることで CAW を獲得することができ、次のように定式化される。

$$\hat{W} = ((I_{CAW}(w_0), t_0), (I_{CAW}(w_1), t_1), \dots, (I_{CAW}(w_m), t_m)) \quad (11)$$

ただし、表記の簡単のため、 $I_{CAW}(w; \{S_{u_0}, S_{v_0}\})$ を $I_{CAW}(w)$ と表現している。こうして得られた CAW の集合をそれぞれ $\hat{S}_{u_0}, \hat{S}_{v_0}$ とする。

エンコーダ. エンコーダは、生成した CAW を入力として注目リンクの embedding を以下の式により生成する。

$$z_{u_0 v_0}(t_0) = \text{AGG}_{\hat{W} \in \hat{S}_{u_0} \cup \hat{S}_{v_0}} (\text{enc}(\hat{W})) \quad (12)$$

ここで、 enc は、CAW を系列処理する学習可能なパラメータを持つ関数であり、次のように定式化される。

$$\text{enc}(\hat{W}) = \text{RNN}(\{f_1(I_{CAW}(w_i))\}_{i=0,1,\dots,m} || \{f_2(t_{i-1}, t_i)\}_{i=0,1,\dots,m}) \quad (13)$$

ただし、 $t_{-1} = t_0$ である。ここで、RNN 関数は系列モデルである RNN などに入力することを表し、 f_1 は CAW を処理するための MLP、 f_2 はタイムエンコーディング関数である。このようにして CAW を系列モデルにより処理することで、リンク予測タスクにおいて重要である構造的パターンを抽出することが可能となる。

デコーダ. デコーダは、予測対象のリンクの embedding を MLP に入力することで最終的なリンクの有無を予測する。ノード u_0, v_0 の2ノードの間において時刻 t_0 にリンクが発生する確率 $p_{u_0 v_0}$ は、以下のように定式化される。

$$p_{u_0 v_0}(t_0) = \text{MLP}(z_{u_0 v_0}(t_0)) \quad (14)$$

WA ベースの利点としては、 I_{CAW} が相対位置情報を保持しているため、リンク予測タスクにおいて重要である構造的パターンを抽出することが可能となる。また、生成した相対位置情報をウォークの各要素として RNN などの系列モデルに入力しているため、注目ノードからの hop 数を考慮可能であり、豊富な構造的パターンを抽出できると考えられる。

欠点としては、WA ベースの手法はリンク予測タスクを解くことを目的として設計されており、動的クラス分類などの他の下流タスクを解くことを想定していないという点で、汎用的ではないということが挙げられる。また、各リンクの予測に対して莫大な数のウォークを生成し、またその数だけ系列モデルにする必要があるため、莫大な計算時間がかかる。

各手法. CAW-N [14] は、上記の CAW を用いることで、CTDG における構造的パターンを明示的に捉えた最初の手法である。MP ベースが主流であった本分野において全く異なる

アプローチにより最先端の精度を達成した手法であり、後の研究に大きな影響を与えている。CAW-N ではリンク embedding に埋め込む情報として、 I_{CAW} のほか、時間的情報として、タイムエンコーディングを用いることで時間的パターンを捉えることを試みている。しかし、実際には時間的に近いイベントが優先的にサンプリングされるウォーク戦略をとっているため、タイムエンコーディングと記憶機構の両方を活用している MP ベースと比較すると、時間的パターンの抽出は効果的に行われていない。また、一つのリンク予測を実行するために多くのウォークを生成し、それらを系列モデルにより処理する際にかかる時間は莫大なものとなる。

NeurTWs [15] は、複数ウォークの作成時に既出のウォークのサンプリング確率を下げたり、高次ノードの影響を強調することによって、CAW に比べてより豊富な近傍情報をリンク embedding に埋め込むことができる手法である。また、タイムエンコーディングの代わりに Ordinary Differential Equations (ODE) を用いることで、より豊富な時間的パターンを明示的に抽出することができる。通常のリンク予測タスクではネガティブサンプルが単純にランダムな 2 ノードをサンプリングしたものであるという問題があり、それに対処するために、コンピュータビジョンの領域において成功した Contrastive Learning を活用して embedding を生成する。しかし、WA ベースの手法の課題であるウォークの生成とそれらを RNN に入力して計算する際の時間的コストの課題は解消されていないため、本手法も大規模グラフへの適用性は低い。

3.3 ハイブリッド

概要. ハイブリッド (MP+WA) は、文字通り MP ベースと WA ベースの方法を組み合わせた手法である。具体的には、MP ベースを基本設計とし、Message Passing もしくはエンコーダにおける集約時に相対位置情報を付与することで、本来捉えられなかった構造的パターンを活用可能にしている。

各手法. PINT [16] は、MP ベースのエンコーダにおける集約関数に理論的裏付けのある工夫を施し、さらに相対位置情報を付与することでノード embedding の表現力を向上させている手法である。ノードエンベディング生成時に、CAW における遷移確率 $\exp(-\alpha(t - t_p))$ を時間的制約として集約関数に乗じることで、最近発生したリンクに明示的に注目し、TGN や CAW が捉えることのできなかった構造的パターンを抽出可能にしている。また、ウォークに基づく相対位置情報を使うことで、従来の MP ベースの手法が捉えることができなかった構造的パターンを抽出可能にしている。この時作成するウォークは行列により計算されるため、処理速度も向上している。一方、集約時に近傍間い合わせを行うことや、ウォークに基づいて相対位置情報を用いることから処理速度が遅く、大規模グラフへの適用性は低い。

NAT [17] は、 k -hop 辞書メモリと、シンプルな相対位置情報を活用することで、表現力と効率性の双方において最先端の性能を示す手法である。 k -hop 辞書メモリは直近の相互作用の履歴からダウンサンプリングした k -hop 以内の近傍ノードの状態

をそれぞれ保持する。これにより、TPGNN と同様に高次情報の抽出が可能となり、最終的なノード embedding の表現力が向上する。本稿にて登場した 4 つの記憶機構について表 2 にまとめる。また、静的グラフにおいて成功した Distance Encoding (DE) を相対位置情報として用いることで、構造的パターンを捉えることも可能である。他にも、キャッシュとハッシュ関数をうまく組み合わせることで、近傍ノードへの並列で高速なアクセスを実現することができ、近傍情報を構築を高速化している。処理速度は TGL と同等程度の現在最先端の実行速度を達成しており、大規模グラフへの適用性は高い。一方で、各ノードごとに自身含めた複数の近傍ノードの状態を保持する必要があるため、使用するメモリは莫大なものとなると考えられる。

4 分 析

この章では、ここまで紹介した手法を表現力と効率性の 2 つの観点で分析する。表 1 に各手法に対するこの 2 つの観点による分析をまとめる。

4.1 表現力

ある手法の embedding がリンク予測を行う上で重要な情報をどの程度表現可能であるかの程度を表現力と定義する。高い表現力を有する手法は様々なパターンの CTDG が識別可能であるため、リンク予測精度は向上する。2 章で定義した 4 つのグラフ特徴についてそれぞれ分析を行う。

属性情報. 属性情報のうち、低次近傍属性情報については MP ベース、WA ベース共に活用可能である。MP ベースでは 2-hop 近傍までであれば過剰平滑の課題が生じないため、低次近傍情報を問題なく捉えることができる。WA ベースでは、ウォークの長さを 2 以上とすることで低次近傍属性情報が獲得可能である。この情報はリンク予測において最も基本的な情報であり、全ての手法において獲得できていることがわかる。

一方、MP ベースでは大半の手法が過剰平滑のため高次近傍属性情報を活用できない。しかし、TPGNN [13] と NAT [17] は記憶状態を hop 数ごとに分けて保持し、Transformer を用いて重要な hop 数を特定することで、過剰平滑を回避し、高次近傍属性情報の抽出を可能にしている。WA ベースでは、ウォーク長を 3 以上に設定することで高次近傍情報が獲得可能である。

時間的パターン. MP ベースの手法は、Xu らにより提案された、三角関数の重ね合わせを用いたタイムエンコーディングを用いることで周期的パターンを捉えている [6]。記憶機構を持つ手法 [10–13, 16, 17] では、系列モデルによる状態の更新により、状態ベクトルが過去のイベント発生時の時間的傾向を要約して保持することができるため、非周期的も含めた時間的パターンも捉えている。このように MP ベースの手法では積極的に時間的パターンを捉えようとする試みが見られる。一方で、WA ベースの手法は記憶機構は使っておらず、またタイムエンコーディングは活用しているものの、最近のイベントが優先的に選択されるウォークの戦略をとっているため、長期的な時間的パターンを捉えられないと考えられる。ただし、WA ベースの中でも

NeurTWs [15] は、時系列処理において強力である Ordinary Differential Equations (ODE) を用いることで、より豊富な時間的パターンを抽出している。

構造的パターン. CAW-N [14] では、CAW を用いて系列モデルに入力することで構造的パターンを捉えており、NeurTWs [15] では、既出ウォークを再サンプリングする確率を下げ、高次ノードの影響を強調する工夫をウォークに施すことで、より豊富な構造的パターンを効果的に捉えている。こうした WA ベースの流れを受けて、PINT [16] や NAT [17] は MP ベースに対して相対位置情報を付与することで構造的パターンを捉えている。PINT が WA ベースと同様にウォークにより相対位置情報を算出する一方で、NAT ではシンプルな Distance Encoding を活用し、最先端の精度を達成している。

4.2 効率性

実アプリケーションへの応用を目標としたときに、処理速度や使用記憶領域の2つの観点において、各手法が効率的であるかどうかを評価する。処理速度が速ければ、大規模グラフに対しても適用可能となり、即時性の求められる実アプリケーションへの応用が現実的となる。使用記憶領域が少なく済む手法であれば、実アプリケーションとして実装がローコストで済んだり、大規模グラフを適用する際に記憶領域が不足することを回避したりすることができる。

処理速度. MP ベースにおいては、エンコーダでの近傍問い合わせが処理速度のボトルネックとなっているが、APAN [11] や TGL [12]、TPGNN [13] では k -hop 近傍の情報を自身のノードのメモリに保持することで、近傍問い合わせをする必要がなくなり、処理速度を向上させている。さらに、TGL では、CSR 行列ベースのデータ構造を活用することで Message Passing における近傍問い合わせを高速化し処理速度を向上させている。一方、WA ベースの手法は、一つのリンク予測を実行するために多くのウォークを生成し、それらを系列モデルにより処理するためにかかる時間が莫大なものとなるため、処理速度の観点で効率であるとは言えない。また、ハイブリッドの NAT [17] は、キャッシュとハッシュ関数をうまく組み合わせることで、近傍ノードへの並列で高速なアクセスを実現している。このように、最近多くの手法が処理速度の向上を目的に新しい手法が提案されていることがわかる。

使用記憶領域. MP ベースの手法は TGAT [6] 以外はノード数に比例する数の状態を保持する必要がある。そのため、ノード数の多い大規模グラフを扱う場合、莫大な記憶領域が必要である。しかし、TGL [12] では、CSR 行列ベースのデータ構造を活用することで CTDG の使用記憶領域を削減することができる。一方、WA ベースの手法は状態を保持する必要がないため、使用記憶領域の観点では効率的な手法であると言える。このように、使用記憶領域の観点では、最近提案された手法において改善点は TGL 以外は見受けられない。

5 今後の課題と方向性

ここでは、今後のこの分野における課題と今後の方向性について述べる。

5.1 効率性の向上のための方向性

効率性に関しては、WA ベースの手法において全体的に処理速度の改善がなされていない。WA ベースで処理速度を高めるためには、現状とは全く異なるアルゴリズムで最終的なリンク embedding を生成する必要があると考えられる。WA ベースの処理速度のボトルネックは、ウォークの数に依存する処理である。具体的には、ウォークから相対位置情報を抽出する部分と、相対位置情報で構成されたウォークをエンコーダにより集約する部分である。この部分の処理速度を改善する方法としては、ウォークを生成せずに、ウォーク生成の際に用いる確率 $\exp(-\alpha(t - t_p))$ を直接用いて相対位置情報を作成することが考えられる。エンコーダにより集約する部分については、RNN を用いずに、グラフ分類タスクに帰着する方法があると考えられる。これは静的グラフのリンク予測において成功を納めた SEAL [24] から着想を得ており、リンク予測 2 ノードの近傍ノードで構成されるサブグラフに相対位置情報を付与し、グラフ分類用の GNN に入力するというものである。こうすることで、既存の WA ベースの手法の課題である処理速度の課題を解決できると考えられる。

5.2 新しい観点での評価

現状、精度に注目した手法が多く、効率性も考慮するものも提案され始めている。一方で、識別可能性や公平性、説明可能性に関してはまだ未開拓である。識別可能性とは、各手法が構造的パターンをどの程度まで識別可能であるかを表す用語である。PINT [16] の論文では、それ以前の手法である TGAT や TGN、CAW を含めて識別可能性の検証をしている一方で、その他の論文では、こうした理論的な検証は取り組まれていない。公平性とはある特定のグループに対して偏見を持たないようにすることである。公平性を重視する場合、例えば、動的リンク予測においては高次数ノードが低次数ノードに比べて推薦されやすい状況を取り除いたり、動的クラス分類では特定の属性を持つグループに対して評価が緩くなったりしないようにする必要がある。このような特定のグループに対して偏見を持つかどうかを検証する研究は、我々の知る限り見つからない。また、近年 AI の実用化に伴い、説明可能性を求められる場面が増加しているが、そのような観点での研究も我々の知る限りでは見つからない。このように、新しい観点で既存手法を見つめ直し、今後の方向性を考えることが重要であると考えられる。

5.3 適切な評価方法

既存のデータセットは SOTA で性能評価をすると、AP や AUC では 90 % 後半の精度を示すことが多く、新しい手法の精度改善の余地が小さい。これを解消するための方法が 2 つ考えられる。一つ目は、より複雑なデータセットを用いること

である。予測の難しいパターンが含まれたデータセットを検証に用いることで、精度の改善が認識しやすくなると考えられる。二つ目は、評価指標の見直しである。AP や AUC はリンクの存在の有無、つまり、0, 1 を判定する指標である。一方で、Hits@10 や Mean Rank は、多数のリンクの候補に対する正解リンクのランキングを評価値に用い、より細かい粒度での予測を行うため、詳細な分析が可能かつ、実用上の評価に適切であると考えられる。そのため、これらの評価指標を用いることで手法を適切に評価できると考えられる。

5.4 データセットの拡充

小規模なデータセットに加えて、大規模なものも公開され始めているが、属性を持つグラフは少ない。既存手法では、主にリンクの発生履歴をベースに embedding を生成し、属性情報は embedding に連結するアプローチしかされていない。属性付きグラフが豊富に存在すれば、効果的にノード属性を活用するアプローチの重要性が増す。また、様々なカラムを含む属性付きグラフを用いることができれば、特定の属性によりノードをグルーピングすることで、公平性の検証も容易に可能になると考えられる。

6 終わりに

本論文では、近年の動向を踏まえ、CTDG を対象とするリンク予測手法の包括的な外観を提供した。最先端のリンク予測手法を、MP ベース、WA ベース、ハイブリッドの3つのカテゴリに分類し、手法間のつながりを整理した。また、表現力と効率性の2つの観点で各手法を比較し、利点欠点を明らかにした。今後の方向性として識別可能性や公平性といった新しい観点で研究をすることや、データセットや評価指標を見直すことが考えられる。

謝 辞

本研究は JSPS 科研費 JP20H00583 の助成を受けたものです。

文 献

- [1] Mohammad Al Hasan and Mohammed J Zaki. A survey of link prediction in social networks. In *Social network data analytics*, pp. 243–275. 2011.
- [2] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *The TKDD*, Vol. 1, No. 1, p. 2–es, 2007.
- [3] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the ICLR*, 2017.
- [4] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *ICLR*, 2018. accepted as poster.
- [5] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the NeurIPS*, 2017.
- [6] da Xu, chuanwei ruan, evren korpeoglu, sushant kumar, and kannan achan. Inductive representation learning on temporal graphs. In *Proceedings of the ICLR*, 2020.
- [7] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *Proceedings of the ICONIP*, 2017.
- [8] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Canedo. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *The Knowledge-Based Systems*, Vol. 187, p. 104816, 2020.
- [9] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the WSDM*, p. 519–527, 2020.
- [10] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *Proceedings of the ICML*, 2020.
- [11] Xuhong Wang, Ding Lyu, Mengjian Li, Yang Xia, Qi Yang, Xinwen Wang, Xinguang Wang, Ping Cui, Yupu Yang, Bowen Sun, et al. Apan: Asynchronous propagation attention network for real-time temporal graph embedding. In *Proceedings of the SIGMOD*, pp. 2628–2638, 2021.
- [12] Hongkuan Zhou, Da Zheng, Israt Nisa, Vasileios Ioannidis, Xiang Song, and George Karypis. Tgl: A general framework for temporal gnn training on billion-scale graphs. *arXiv preprint arXiv:2203.14883*, 2022.
- [13] Zehong Wang, Qi Li, and Donghua Yu. Tpgnn: Learning high-order information in dynamic graphs via temporal propagation. *arXiv preprint arXiv:2210.01171*, 2022.
- [14] Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. In *Proceedings of the ICLR*, 2021.
- [15] Ming Jin, Yuan-Fang Li, and Shirui Pan. Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs. In *Proceedings of the NeurIPS*, 2022.
- [16] A. H. Souza, D. Mesquita, S. Kaski, and V. Garg. Provably expressive temporal graph networks. In *Proceedings of the NeurIPS*, 2022.
- [17] Yuhong Luo and Pan Li. Neighborhood-aware scalable temporal network representation learning. *Learning on Graphs Conference*, 2022.
- [18] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI*, 2018.
- [19] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the NIPS*, 2014.
- [20] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of the NIPS*, 2014.
- [21] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD*, pp. 1269–1278, 2019.
- [22] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *Proceedings of the ICLR*, 2019.
- [23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, Vol. 30, , 2017.
- [24] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *Proceedings of the NeurIPS*, 2018.