

修士論文の和文要旨

研究科・専攻	大学院 情報理工学研究科 情報・ネットワーク工学専攻 博士前期課程		
氏名	橋本 陸	学籍番号	2031117
論文題目	動的グラフに対する回転を用いたノードエンベディングにおける 基準ノード選択の検討		
<p>要 旨</p> <p>グラフデータ、つまりはノード（頂点）とエッジ（辺）で表現されたデータの分析に対して、ノードエンベディングは有効な手法の一つである。ノードエンベディングはグラフのノードを低次元空間でベクトル表現する手法である。この時、グラフの構造を反映させる必要があり一般的にノード間に辺が存在する、共通の多くの隣接ノード持つ等の関係が強いノード同士の距離が小さくなるようにベクトル表現する。このように低次元空間でベクトル表現することにより既存の機械学習のアルゴリズムを用いた分類器や予測器をノード分類やリンク予測などのグラフで表現される情報に対するタスクに効果的に使用することが可能になり、今日までに様々な手法が提案されている。</p> <p>また、近年では時間変化するグラフ、つまり動的グラフに対するノードエンベディングの手法に関する研究も行われている。こちらも様々な手法が提案されているが多くの手法で動的グラフに対するノードエンベディングではベクトル表現に時間情報を反映させるため他時刻の同一ノードのベクトル表現に対してアライメントをとる。具体的には、各時刻のベクトル表現に対して制約を加えるまたは処理を加えることで、前の時刻でベクトル表現された近くで現時刻のベクトル表現を行う。このアライメントを取る方法として先行研究で提案されたプロクラステス回転を用いた手法は高い効果を示している。プロクラステス回転は各データ点が一対一対応をする時、2つのデータ点群が合致するように回転する。</p> <p>本研究ではより効果的にプロクラステス回転を用いるためにノードの時間変化を定義し時間変化の小さいノードを基準にプロクラステス回転を行うことを提案する。まず、時間変化が小さい場合よりそれ以前の時刻のベクトル表現と近い表現になると考えた。そして、そのようなノードを基準にすることでより精密なアライメントが可能となる。結果的にグラフで表現されるデータに対するタスクに対しより効果的に機械学習のアルゴリズムを適用することができると考えた。以上の考えをもとに、いくつかの実世界データを用いた数値実験を行った。実験により得た結果をもとに既存の全ノードを基準にプロクラステス回転を適応する場合と比較することで動的グラフに対する回転を用いたノードエンベディングにおける基準ノード選択がグラフで表現される情報に対するタスクに及ぼす効果を示す。</p>			

令和3年度 修士論文

動的グラフに対する回転を用いた
ノードエンベディングにおける
基準ノード選択の検討

電気通信大学 情報理工学研究科
情報・ネットワーク工学専攻

学籍番号：2031117

橋本 陸

指導教員

劉志准教授

令和4年1月28日

目次

第1章	序論	3
1.1	研究背景・目的	3
1.2	研究成果	5
1.3	論文構成	5
1.4	表記法	6
第2章	関連研究	7
2.1	Node embedding	7
2.1.1	概要	7
2.1.2	既存手法	9
2.2	動的グラフに対する Node Embedding	11
2.2.1	概要	11
2.2.2	既存手法	12
2.3	直交プロクラステス回転	13
2.3.1	概要	13
2.3.2	数学的定式化とその解法	13
2.4	グラフ分析と機械学習	14
2.4.1	グラフの分析タスク	14
2.4.2	分析タスクのための機械学習の手法	16
2.4.3	評価指数	18
第3章	基準ノード選択制回転を用いた動的Node Embedding:SPRDNE	22
3.1	直交プロクラステス回転の時系列データへの適応の考察	22
3.2	提案手法	23
3.2.1	概要	23
3.2.2	回転基準ノードの選択	25

第4章 実験と評価	29
4.1 実験概要	29
4.1.1 実験内容とその目的	29
4.1.2 データセット	30
4.1.3 実験条件	35
4.2 実験結果および考察	36
4.2.1 collegeMsg	36
4.2.2 ia-reality-call	41
4.2.3 SFHH-conf-sensor	46
第5章 結論	53
謝辞	54
参考文献	55

第1章 序論

1.1 研究背景・目的

実世界では、繋がり(リンク)を持つものが多数存在する。リンクには、人や組織等の社会的リンクや分子構造などの物理的リンクなど様々なものがある。このようなリンクを持つデータを表現する方法としてグラフがある。グラフはリンクに着目し頂点(ノード)と辺(エッジ)によりその様なデータを簡潔に表現することができる。分子構造を例にあげるならばノードが分子、結合部が辺にあたる。

このようなノードと辺を使用し繋がりを図示するというグラフの概念は、1736年ケーニヒスベルクの七つの橋問題に関連しLeonhard Euler (1707–1783)が考案したものが起源の一つだとされている。[1] ケーニヒスベルクの七つの橋問題はケーニヒスベルクという都市に流れるプレーゲル川にかかる7つの橋を題材とした問題である。この7つの橋全てを1度ずつ通り元の場所に帰ることができるかというのがこの問題の概要である。これは一筆書きに関連する問題であり、オイラーは図1.1に置き換えこの問題を解決した。図1.1では辺を橋、川により隔てられた陸地をノードとしている。グラフは起源となったとされるケーニヒスベルクの七つの橋問題をはじめ様々な問題の解決に使われてきた。特に代表的なものとして、巡回セールスマン問題や四色問題等がある。

現代ではインターネットの普及等の技術の進歩により、巨大なグラフが多数生成、あるいは認知されている。そのようなグラフには多くの場合、潜在的に重要な特徴や有用な情報が存在しており、グラフの分析に対する注目度はますます高まっている。一方で、特に巨大なグラフを分析することは困難である。そこでNode embeddingが提案された。Node embeddingはノードを低次元空間に射影しベクトルで表現することを試みる手法である。低次元空間に射影しベクトルで表現することで既存の機械学習のアルゴリズムの効果的な適応が可能となる。これによりグラ

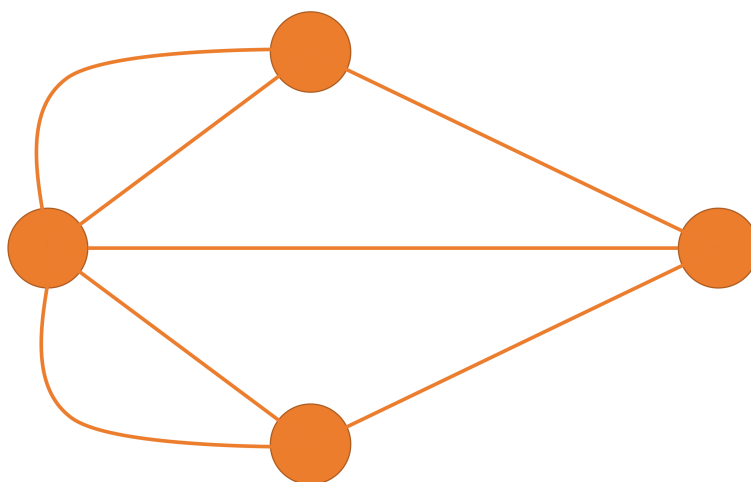


図 1.1: ケーニヒスベルクの七つの橋問題のグラフ表現

フの潜在的情報や特徴を明らかにすることが可能となる. Deepwalk [2] や Node2vec [3] を始め今日までに様々な手法が提案されている.

また, 近年では時間情報を持つグラフに対する分析も注目されている. 実世界でも SNS など時間によって変化するグラフは多数存在している. 一般的に動的グラフに対する Node embedding では時刻間に何らかの適切な制約を与えるまたは処理をすることで時間変化をベクトル表現に反映させる. 最も一般的なものは短い時刻間の同一ノードの距離を近づけるような制約や距離である. これは同一ノードは短い時刻間の場合に比較的特徴が類似するとされており, 直前の時刻と近い距離に位置すると考えられる為である. tNodeEmbed [4] ではその処理として, 直交プロクラステス回転 [5] を用いている.

プロクラステス回転は全単射の対応を持つ2つの点集合が存在した場合, 一方を他方に集合内の対応する点間のユークリッド距離の合計が最小になるように回転する. これを用いて近い二つの時刻間の両方で辺を持つノードを対象に同一ノードのベクトル表現を距離の合計が最小となる回転を求め, 後の時刻のベクトル表現に適応する. この処理により, 異なる時刻間のベクトル表現を類似した空間で表現することが可能となり, 時間変化をベクトル表現に反映させることができる.

しかし, 隣接関係の変化のみに注目しても時間変化の中で新たに隣接するノードが増加するノードや逆に他のノードとの隣接関係を失うノードも存在する. また, 隣接関係の変化しないノードも存在する. このよ

うに各ノードの隣接関係はそれぞれ多様に変化し、時間が経つにつれて特徴や性質が多く変化するノードや、時間によってあまり性質や特徴が変わらないノードが存在すると予想でき、変化の度合いはノードそれぞれで異なると考えられる。そこで、本稿ではこのようなノードの変化の度合いの違いに注目し、tNodeEmbedのように動的グラフに対するNode embeddingに直交プロクラステス回転を利用する際に、より効果的な回転を行うために回転基準となるノードの選択を行うことを提案する。さらに、回転基準となるノードの選択のために必要となるノードの時間変化の定義を提案し、実世界データを用いた数値実験によって回転基準となるノードの選択および提案したノードの時間変化の定義の有効性を明らかにした。

1.2 研究成果

本研究で得られた主な成果を示す。

- プロクラステス回転を用いた動的グラフにおけるNode embeddingにおいて回転基準となるノードを選択することを提案した。
- 基準選択のために必要なノードの時間変化に対し、最短経路の長さの変化を用いて定義することを提案した。
- 幾つかの実世界データセットにおいて動的グラフに対する時系列リンク予測においてROC曲線のAUCの向上を確認した。

1.3 論文構成

本論文の構成について述べる。本章では研究目的とその背景について述べ、本稿で使用する表記について説明する。2章では、本研究に関連するいくつかの研究について紹介する。3章では、提案手法の基準ノード選択制回転を用いた動的Node Embedding (SPRDNE:Node Embedding by Selective Procrustes Rotation for Dynamic Graph)について説明する。4章では、本研究の手法についての評価実験を行い提案手法を評価する。また、評価実験とその他諸実験に基づいて結果について考察する。最後に、5章でまとめと今後の課題について述べる。

1.4 表記法

スカラーを小文字で a, b, \dots と表す. d を任意の自然数とするとき, \mathbb{R}^d は d 次元のユークリッド空間を表し, \mathbb{R}_+^d はそのうちの非負の部分空間を表す. ベクトルを太字の小文字で $\mathbf{a}, \mathbf{b}, \dots$ と表す. ベクトル $\mathbf{a} \in \mathbb{R}^d$ の第 i 成分を $[\mathbf{a}]_i$ と表す. また, p -ノルムは $\|\mathbf{a}\|_p = \sqrt[p]{\sum_i |[\mathbf{a}]_i|^p}$ と定義される. 行列を太字の大文字で $\mathbf{A}, \mathbf{B}, \dots$ と表す. m, n を任意の自然数とするとき, 行列 $\mathbf{A} \in \mathbb{R}^{m \times n}$ の第 i 行 j 列での要素を $[\mathbf{A}]_{i,j}$ と表す. また, フロベニウスノルムは $\|\mathbf{A}\|_F = \sqrt{\sum_{m,n} ([\mathbf{A}]_{m,n})^2}$ と定義される. $\text{Tr}(\cdot)$, $(\cdot)^T$ は, それぞれ行列の跡と転置を示す. ベクトルまたは行列を含む演算において, \odot は要素ごとの積をとる. また, $/$ は要素ごとの除算をとる. その他主要な表記法を表 1.1 に記す.

表 1.1: 主な表記法

表記法	定義, 説明
$G_t = (\nu_t, \varepsilon_t)$	時刻 t の無向または有向グラフ, $1 \leq t \leq T, t$: 自然数
ν_t	時刻 t において辺を持つノードの集合
ε_t	G_t の辺集合
$n_t := \nu_t $	G_t のノード (頂点) 数
$ \varepsilon_t $	G_t のエッジ (辺) 数
d	ノードのベクトル表現の次元数
$\mathbf{A}_t \in \mathbb{R}_+^{n_t \times n_t}$	G_t の隣接行列
\mathbf{V}'_t	ν_t に属するノードの回転行列適応前のベクトル表現を行とする行列
\mathbf{V}_t	ν_t に属するノードの回転行列適応後のベクトル表現を行とする行列
\mathbf{R}_t	\mathbf{V}'_t に作用させる d 次元回転行列
$\hat{\nu}_t$	$\hat{\nu}_t = \nu_t \cap \nu_{t-1}$, すなわち時刻 $t-1$ と t で辺を持つノードの集合
$s_t^{v,u}$	時刻 t におけるノード v からノード u への最短経路の長さ
$W(x)$	重み関数, $W : x \mapsto y$ s.t. $x \in [0, 1], 0 \leq y$

第2章 関連研究

この章ではNode embedding, tNodeEmbed, 及びtNodeEmbedに使用されている直交プロクラスレス回転 [5] およびその他関連研究について説明する。

2.1 Node embedding

2.1.1 概要

実世界には様々なグラフで表現できるデータが存在している。例として、道路や鉄道などの交通インフラや分子構造, SNSなどがあげられる。そのようなデータには多くの場合, 潜在的に存在する特徴や情報を明らかにする事で現実の様々な問題に役立てることができる。例として, SNSを分析することで, ターゲティング広告などマーケティングへ応用することができる。具体的なグラフの分析タスクの例として, ノードのクラス分類, クラスタリング, リンク予測などがある。これらの分析タスクを効果的に処理しグラフを分析するための手法の一つとしてNode embeddingが提案されている。Node embeddingはグラフの構造等を基に各ノードを低次元のベクトルで表現する手法である。これにより図2.1に示すようにグラフの分析に既存の機械学習のアルゴリズムを効果的に使用することが可能となる。

図2.2はグラフの構造から各ノードをベクトル表現した例である。この図のうち(a)は元の入力情報, (b)は(a)のグラフの各ノードのベクトル表現を表している。また, このベクトル表現が満たすべき一般的条件として以下の3つが挙げられる。

1. 元の隣接行列での表現に比べ十分に低次元であること。
2. グラフの構造等の情報が反映されている。
3. ベクトルの要素は実数値をとる。

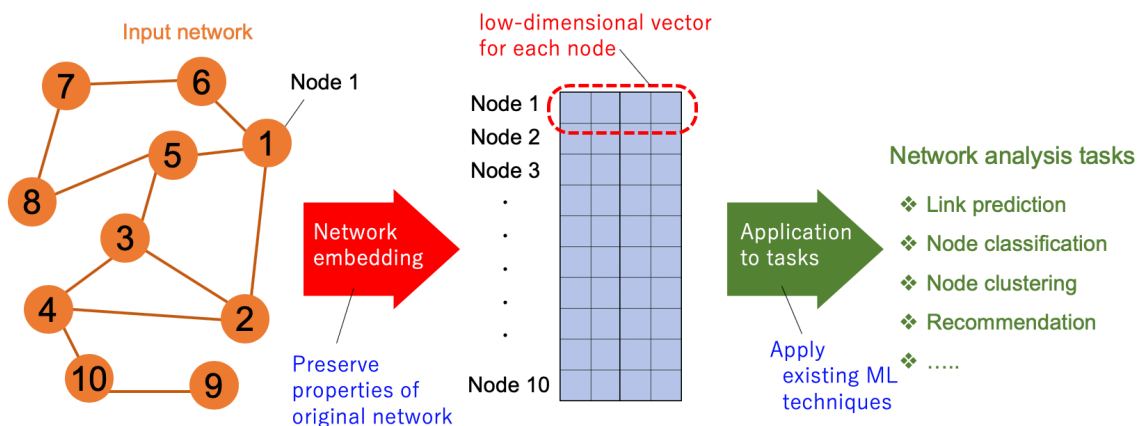


図 2.1: 概要図 : Node embedding

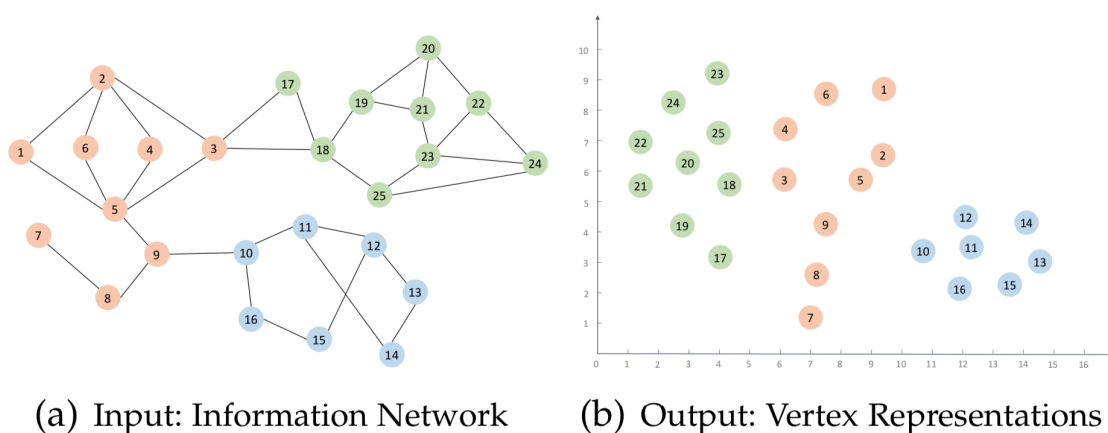


図 2.2: Node embedding の例 [6]

3つの条件のうち2について具体例をあげる．文献[6]では反映されるグラフの構造，つまりベクトル表現をした時のノード同士の類似度が高くなる要素として次の四つを定義している．

1. 一次近接

直接隣接しているノード同士．図2.2ではノード7，ノード8が当てはまる．

2. 二次近接及び高次近接

多くの隣接ノードにより間接的に隣接しているノード同士．図2.2ではノード2とノード5が二次隣接に当てはまる．ノード20とノード25は高次隣接に当てはまる．

3. 構造的役割による近接

ノード間の距離が離れた場合であっても，多くのノードに隣接している等，構造的役割が似ているノード同士．図2.3のノード4とノード

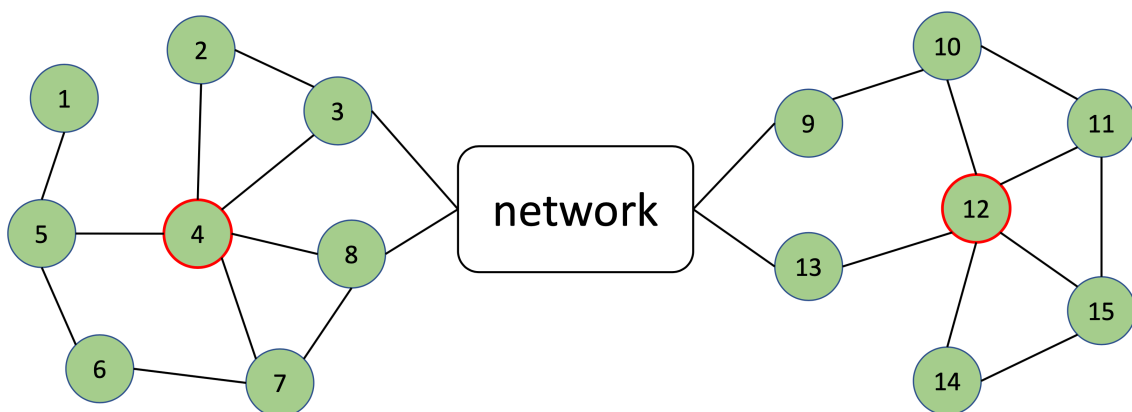


図 2.3: 同じ構造的役割の例 [6]

ド12が当てはまる.

4. コミュニティ内近接

同じコミュニティ内, つまりエッジの密度の高い塊内のノード同士. 図2.2ではノード10からノード16が当てはまる.

2.1.2 既存手法

文献[6]によると Node embedding の手法は表2.1に示す通り, 4つとそれらを組み合わせたハイブリットな手法に分類することができる.

表 2.1: 手法の分類 ([6] より抜粋)

方法論	代表的アルゴリズム
行列分解	Social Dim. [7][8] GraRep [9], NetMF [10]
ランダムウォーク	Deepwalk [2], Node2vec [3]
エッジモデリング	LINE [11]
深層学習	DNGR [12]
ハイブリット	DP [13]

1. 行列分解

行列分解を元にした手法では各ノードの接続性を行列で表し, 行列分解を用いて各ノードを低次元空間でベクトル表現を行う. 接続性を表す行列は様々な種類があり, 例として隣接行列や各ノード同士のコサイン類似度で評価した行列などがある. 長所として全体的な構成を取得できる. 一方短所として時間, メモリのコストが高い.

2. ランダムウォーク

ランダムウォークを元にした手法では、ランダムウォークを用いてグラフのサンプリングを行う。ランダムウォークの足跡内での隣接の発生頻度からノードのベクトル表現を求める。発生頻度の高いノード同士は関係が近いとされベクトル表現した空間においても近接関係が保たれやすい。比較的効果的な手法ではあるがノードのベクトル表現にグラフ上で比較的近い距離内の構造しか反映することができない。

3. エッジモデリング

エッジモデリングを元にした手法では各ノードの接続性から直接ベクトル表現を得る。接続性の情報として隣接ノードだけではなく隣接ノードの隣接ノードを用いる手法もある。効果的な手法ではあるがノードのベクトル表現にグラフ上で比較的近い距離内の構造しか反映することができない。

4. 深層学習

複雑で非線形なベクトル表現を深層学習を利用することで取得する。非線形な構成を取得できる一方で時間的コストが大きい。

以下で Node embedding の代表的な手法をいくつか紹介する。

Deepwalk [2]

Deepwalk は自然言語処理のうち単語をベクトルで表す手法を応用し各ノードのベクトル表現を得る手法であり、Word2vec [14] という手法が元になっている。Word2vec では単語の意味はその周辺の単語、つまりは文脈によって決まるとの考えに基づき、Skip-Gram [15][16] というアルゴリズムを用いて、与えられた文章からその単語の周りに出現する単語によってベクトル表現を求める。DeepWalk はノードを単語の代わりとして扱う。また、ランダムウォークを用いてグラフからサンプリングを行う。これは文章の代わりにランダムウォークの足跡を用いる為である。最終的には Skip-Gram を用いてランダムウォークの足跡から周辺に出現するノードによってベクトル表現を求める。

Node2vec [3]

DeepWalkを発展させたランダムウォークに基づいた手法。特徴は通常のランダムウォークではなく遷移確率にバイアスをかける点である。また、そのバイアスの強さはパラメータによって制御が可能となっている。バイアスのかけ方として深さ優先探索、幅優先探索がある。深さ優先探索では起点となるノードからより遠いノードに遷移しやすくなり、構造の類似したノード同士がベクトル表現された空間上でも近接しやすくなると主張されている。幅優先探索では起点となるノードの近くのノードに遷移しやすくなり、グラフ上で近いノード同士やグラフの構造上で同じクラスターに属すノード同士がベクトル表現された空間上でも近接しやすくなると主張されている。

LINE [11]

LINEではノードの接続性のうち一次近接および二次近接に着目する手法である。具体的には一次近接および二次近接に関連した目的関数を定める。これら2つの目的関数を最小化するノードのベクトル表現それぞれ求める。最終的なベクトル表現はこの二つのベクトル表現を直列結合したものとする。

2.2 動的グラフに対するNode Embedding

2.2.1 概要

グラフで表現されるデータの中にはノードやエッジに時間情報を持ち時間変化する場合がある。通常のグラフに対するNode Embeddingに求められる一般的条件に加えて時間変化をベクトル表現に反映させる必要がある。つまり、 G_1, G_2, \dots, G_T が与えられた時、ノードのベクトル表現を $\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_T$ とした場合、ある時刻 t のノードのベクトル表現 \mathbf{V}_t はそれ以前のグラフ G_1, G_2, \dots, G_{t-1} やそのノードのベクトル表現 $\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_{t-1}$ の影響を受ける。また、 \mathbf{V}_t はそれ以降のグラフ $G_{t+1}, G_{t+2}, \dots, G_T$ のノードのベクトル表現 $\mathbf{V}_{t+1}, \mathbf{V}_{t+2}, \dots, \mathbf{V}_T$ に影響を与える。したがって、多くの手法で適切な影響を与える制約を取り入れている。例として、近傍の別時刻、同一ノードの距離を小さくする制約がある。これは別時刻であっ

でも近傍かつ同一ノードならばその部分空間を共有するという仮定に基づき与えられる制約である。最も一般的な方法としてベクトル表現の差の ℓ_2 ノルム制約が挙げられる。また、同様な考えに基づきプロクラステス回転を利用することが可能である。

2.2.2 既存手法

DynmicTri [17]

この手法の互いに隣接した3つのノードつまり閉じた三角関係が共通の隣接を持つ2つの非隣接ノードつまり開いた三角関係からどのように発展したかをモデル化することで、グラフの時間変化をベクトル表現に反映させた手法。開いた三角関係が閉じた三角関係に変化する確率、および変化しない確率を定義し時間変化するグラフのノードのベクトル表現の学習に利用した。

tNodeEmbed [4]

動的グラフに対する Node embedding に初めてプロクラステス回転を用いた手法。得られたベクトル表現から LSTM を用い時系列リンク予測とマルチラベルクラス分類を行い特にクラスタリング系数が低いグラフに対して高い効果を示した。

FILDNE [18]

本研究と同様に tNodeEmbed を元にプロクラステス回転を行う際に基準ノードを選択するのが特徴。この手法については本研究と同時期に行われていた研究であり、偶然にもノードの選択を行うという点において方向性が一致した。一方で、本研究と異なり選択基準となるノードの時間変化として隣接関係の変化を用いている。また、本研究では tNodeEmbed と同様に各時刻で得たベクトル表現を LSTM により処理したが、FILDNE ではいくつかのベクトル表現を重みつけた上で足し合わせることを提案している。

2.3 直交プロクラステス回転

2.3.1 概要

直交プロクラステス回転は因子軸の回転法の一つであり，点集合を表す2つの行列 \mathbf{A}, \mathbf{B} が与えられたとき， \mathbf{A} が \mathbf{B} に可能な限り合致するように回転する直交行列 \mathbf{R} を求める．

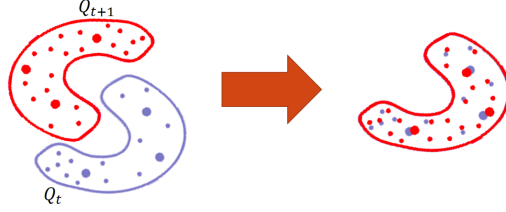


図 2.4: Illustration of the alignment between 2 consecutive time-steps [19]

2.3.2 数学的定式化とその解法

数学的定式化

直交プロクラステス問題は次のように定式化される．

$$\mathbf{R} = \arg \min_{\Omega} \|\mathbf{A}\Omega - \mathbf{B}\|_F^2 \text{ subject to } \Omega^T \Omega = \mathbf{I}_N \text{ (}\mathbf{I}: \text{identity matrix)} \quad (2.1)$$

解法

式(2.1)は $\mathbf{B}^T \mathbf{A}$ の特異値分解を $\mathbf{U}\Sigma\mathbf{V}^T$ とすると $\Omega = \mathbf{V}\mathbf{U}^T$ とすることで最小化される．文献[20]により以下のように証明がされている．

$$\begin{aligned} \mathbf{R} &= \arg \min_{\Omega} \|\mathbf{A}\Omega - \mathbf{B}\|_F^2 \\ &= \arg \min_{\Omega} \text{tr}\langle (\mathbf{A}\Omega - \mathbf{B})(\mathbf{A}\Omega - \mathbf{B})^T \rangle \\ &= \arg \min_{\Omega} \|\mathbf{A}\Omega\|_F^2 + \|\mathbf{B}\|_F^2 - 2\text{tr}\langle \mathbf{A}\Omega\mathbf{B}^T \rangle \end{aligned}$$

第一項，第二項が定数となる為第三項の最大化問題に帰着できる．

$$\mathbf{R} = \arg \max_{\Omega} \text{tr}\langle \mathbf{B}^T \mathbf{A}\Omega \rangle \quad (2.2)$$

$$= \arg \max_{\Omega} \text{tr}\langle \mathbf{U}\Sigma\mathbf{V}^T \Omega \rangle \quad (2.3)$$

$$= \arg \max_{\Omega} \text{tr}\langle \mathbf{V}^T \Omega \mathbf{U}\Sigma \rangle \quad (2.4)$$

式(2.2)から式(2.3)への変換では、 $\mathbf{B}^T \mathbf{A}$ を $\mathbf{U}\Sigma\mathbf{V}^T$ のように特異値分解している。式(2.3)から式(2.4)への変換では、行列の跡の循環性を利用し、順番を変更している。ここで $\mathbf{V}^T \Omega \mathbf{U} = \mathbf{X}$ とすると $\mathbf{X}\mathbf{X}^T = \mathbf{I}$ となるので \mathbf{X} は正方な直交行列である。従ってその対角成分は必ず1以下となる。よって $\text{tr}\langle \mathbf{X}\Sigma \rangle$ が最大となる時 $\mathbf{X} = \mathbf{I}$ となり

$$\mathbf{I} = \mathbf{V}^T \Omega \mathbf{U} \Leftrightarrow \Omega = \mathbf{V}\mathbf{U}^T$$

特異値分解

ここで前述の解法に用いられた実行列に対する特異値分解について説明する。全ての行列 $\mathbf{X} \in \mathbb{R}^{m \times n}$ は以下の式を満たす。

$$\mathbf{X} = \mathbf{V}\Sigma\mathbf{U}^T \tag{2.5}$$

このとき $\mathbf{V} \in \mathbb{R}^{m \times m}$, $\mathbf{U} \in \mathbb{R}^{n \times n}$ は直交行列、 $\Sigma \in \mathbb{R}^{m \times n}$ は、

$$\Sigma = \left(\begin{array}{c|c} \Sigma_1 & \mathbf{O} \\ \hline \mathbf{O} & \mathbf{O} \end{array} \right)$$

という形の行列になる。 \mathbf{O} は零行列を表す。この時 $\Sigma_1 \in \mathbb{R}^{r \times r}$ は対角行列となり、その成分を特異値と呼ぶ。また、 r は \mathbf{X} の階数となる。このような行列の特性を用いて式(2.5)の様に行列を分解することを特異値分解と呼ぶ。

2.4 グラフ分析と機械学習

本節ではグラフの分析タスクやそのための機械学習の手法、また評価指数について述べる。

2.4.1 グラフの分析タスク

グラフの分析タスクにはノードのクラスタリングや分類、リンク予測、グラフ分類などがある。本節ではそのうちのノードのクラス分類およびリンク予測について紹介する。また、Node embeddingによりベクトル表現を直接評価する指標は存在しない。そのため多くの場合、得られたベクトル表現を使用したグラフの分析タスクを評価基準としている。ノードのクラス分類やリンク予測そのような評価基準としても多く使われている。

ノードのクラス分類 (Node Classification)

グラフのノードを既知のクラスに分類するタスク。概要図を図2.4.1に示す。図2.4.1は3つのクラスラベルA,B,Cがありクラスが既知のノードを元にクラスが未知のノードのクラスを推定する。

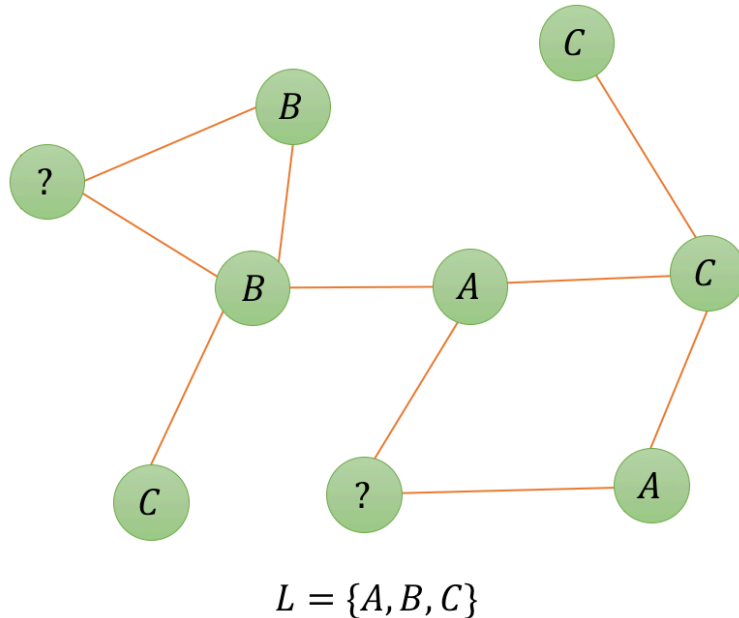


図 2.5: 概要図：ノードのクラス分類

リンク予測 (Link Prediction)

グラフのエッジ(リンク)の存在を予測するタスク。概要図を図2.6に示す。単一の不完全なグラフが与えられたとき既知のエッジから未知のリンクを予測する。または、ノードあるいはエッジの特徴量が与えられたときエッジの存在を予測する。

時系列リンク予測 (Temporal Link Prediction)

時間変化するグラフがあるとき、その後のエッジの存在を予測する。図2.7に概要図を示す。図2.7では時刻 $t=1$ から $t=T$ のグラフが与えられており、それ以降の $t=T+1$ のグラフのエッジを予測している。

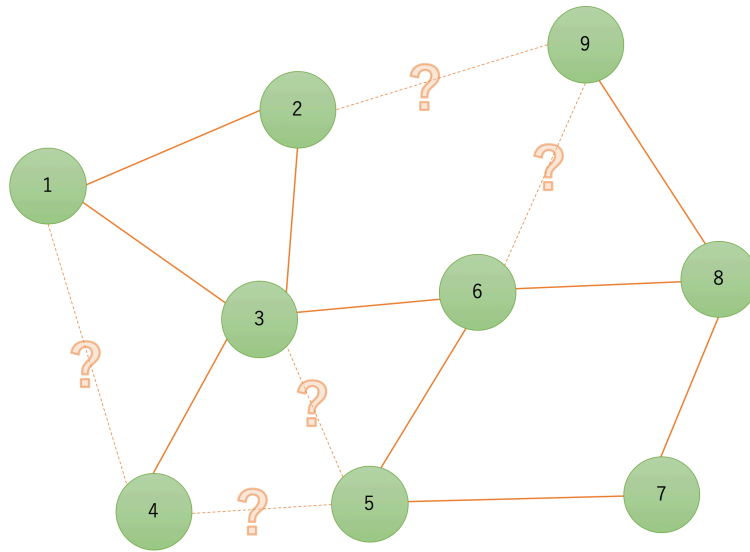


図 2.6: 概要図: リンク予測

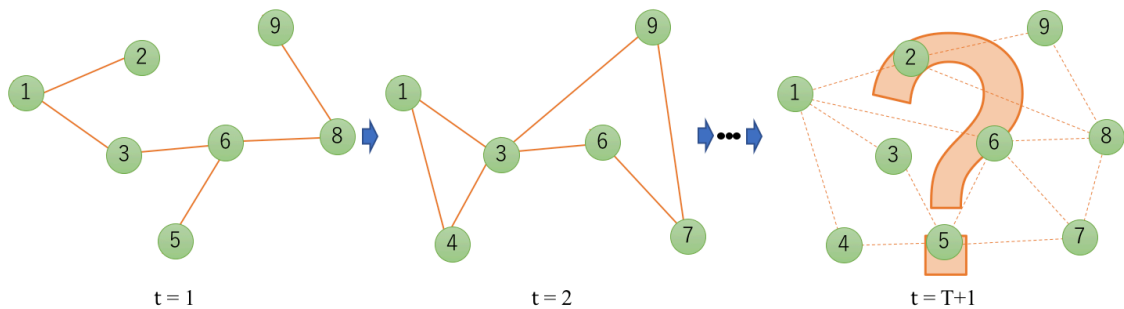


図 2.7: 概要図: 時系列リンク予測

2.4.2 分析タスクのための機械学習の手法

ここではグラフの分析タスクに予測器や分類器として利用される機械学習の手法について述べる。

ロジスティック回帰 [21]

ロジスティック回帰は多変量解析の一種であり、結果が2値で表される時に使用される教師あり学習の一種である。予測器や分類器として利用される。ロジスティック回帰では活性化関数にロジット関数(シグモイド関数)を使用することで事象に対する確率が出力される。ロジット関数は次のように定義される。

$$y(x) = \frac{1}{1 + \exp(-x)} \quad (2.6)$$

また x は特徴量 (説明変数) と重みで表される. 特徴量が n 個あるとする. 特徴量 i を x_i , 重みの定数項を a_0 , 特徴量 i に対応する重みを a_i とすると次のように表される.

$$x = a_0 + \sum_{k=1}^n a_k x_k \quad (2.7)$$

この重みは教師データを使用した最尤法, あるいは最小二乗法により求めることができる.

LSTM(Long Short-Term Memory)

LSTM [22] は脳の神経回路の仕組みを模したモデルのニューラルネットワーク (NN) の内, 時系列データを扱うことが可能な再起型ニューラルネットワーク (RNN) の一種. LSTM は長期依存の変化を学習できることが特徴. 一方で当初は短期間での大きな変化に対応できないという点があったがこの問題に対して文献 [23] が忘却ゲートを導入することを提案された. これを始めとし, その後も様々な研究がされており, 多様な分野で活用されている. 図 2.8 に LSTM の中間層の概要図を示す.

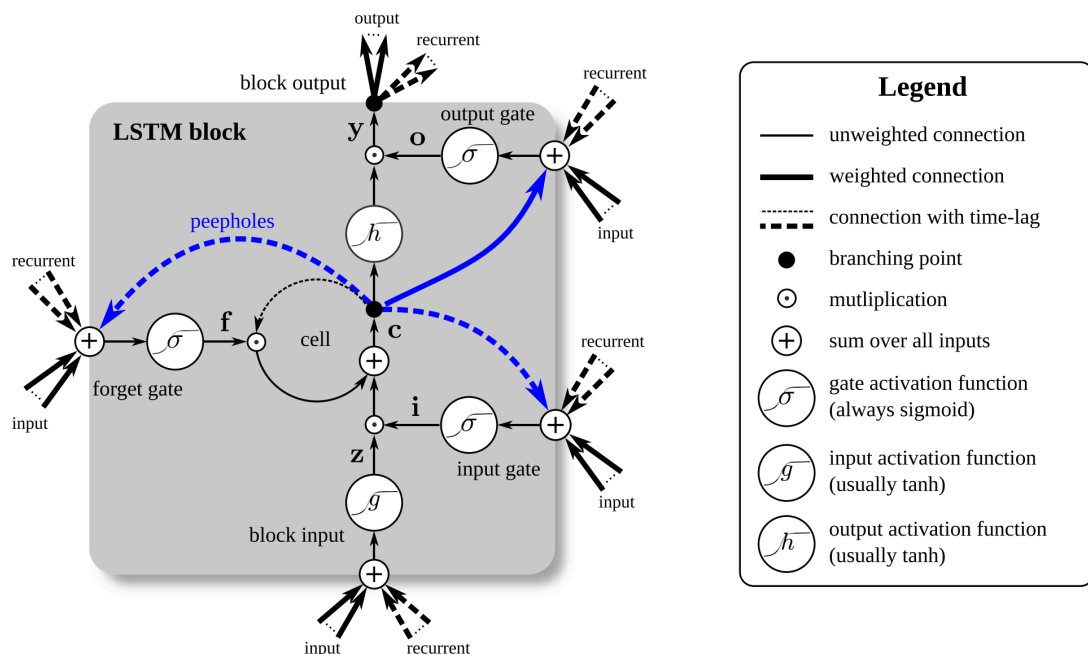


図 2.8: 概要図: LSTM [24]

特徴量のスケールリング

特徴量のスケールリングとは特徴量の取りうる値の範囲を変えることであり、スケールの異なる特徴量を揃えることができる。スケールの異なる特徴量ではうまく学習できない場合があり、特にデータ間の距離がアルゴリズムで重要な場合に特徴量のスケールリングは効果的である。主なスケールリングの方法として下記の2つがある。

- 正規化

特徴量を任意の最小値, 最大値の範囲に変換する。主に $[0, 1]$ や $[-1, 1]$ の範囲を使用する。分布は変化せず、画像のRGBの強さなど特徴量の最大値と最小値が決まっている場合などに有効である。一方で外れ値の影響によって偏ったスケールリングになる場合がある。

- 標準化

特徴量を平均0, 標準偏差1になるように変換する。外れ値に強い。

具体的には以下の数式を用いて変換される。式(2.8)は範囲を $[m, M]$ とする場合の正規化の変換式である。式(2.9)は標準化の変換式である。

$$x_{norm,i} = \frac{x_i - x_{min}}{x_{max} - x_{min}}(M - m) + m. \quad (2.8)$$

$$x_{std,i} = \frac{x_i - \mu_x}{\sigma_x}. \quad (2.9)$$

表記	定義
x	特徴量
x_i	特徴量 x サンプル i の値
$x_{norm,i}$	正規化された x_i
$x_{std,i}$	標準化された x_i
x_{min}	特徴量 x のうち最小の値
x_{max}	特徴量 x のうち最大の値
μ_x	x の平均
σ_x	x の標準偏差

2.4.3 評価指数

ここでは予測器や分類器の結果に対する評価指数について述べる。まず、混合行列を表2.2に示す。この4種の結果について以下で説明する。

表 2.2: 混合行列

		予測値	
		Positive	Negative
真値	Positive	TP(True Positive)	FN(False Negative)
	Negative	FP(False Positive)	TN(True Negative)

- 真陽性数 (True Positive)
実際は陽性であるもののうち正しく陽性と判定された数.
- 偽陽性数 (Fales Positive)
実際は陰性であるもののうち誤って陽性と判定された数
- 偽陰性数 (False Negative)
実際は陽性であるもののうち誤って陽性と判定された数
- 真陰性数 (True Negative)
実際は陰性であるもののうち正しく陰性と判定された数

この4種の数値を使用した基本的な評価指数を紹介する.

- 正解率 (accuracy)

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}.$$

正解の程度を評価する. 数値が高いほど良い. 一方でクラス内の要素数が異なる時, 要素数が多いクラスに予測値が偏ると高い数値を示すため評価指数として不十分な場合がある.

- 真陽性率, 再現率 (recall)

$$\text{Recall} = \frac{TP}{TP + FN}.$$

実際に陽性であるもののうち予測値で陽性であった割合. 数値が高いほど良い.

- 偽陽性率

$$\text{FalesPositiveRate} = \frac{FN}{TP + FN}.$$

実際は陰性であるもののうち予測値で陽性であった割合。数値が低いほど良い。

- 真陰性率，特異度 (specificity)

$$\text{Specificity} = \frac{TN}{FP + TN}.$$

実際に陰性であるもののうち予測値で陰性であった割合。数値が高いほど良い。

- 偽陰性率

$$\text{FalesNegativeRate} = \frac{FP}{FP + TN}.$$

実際に陰性であるもののうち予測値で陽性であった割合。数値が低いほど良い。

- 適合率 (precision)

$$\text{Precision} = \frac{TP}{TP + FP}.$$

予測値で陽性としたもののうち真に陽性であった割合。数値が高いほど良いとされる。

次に2値分類の評価指数として利用されるROC曲線，PR曲線，およびf1値について説明する。

- ROC曲線 (Receiver Operatorating Characteristic Curve)

予測値を閾値で陽性と陰性を判定した時，縦軸を真陽性率，横軸を偽陽性率とし閾値毎にプロットした曲線。テストデータ全体を評価できる。一方でテストデータの内実際に陽性であるデータが極端に少ない場合等クラスに属するデータ数の差が大きい場合には評価指数として機能しない場合がある。xy平面の第一象限においてこの曲線と $y = 0, x = 1$ に囲まれた領域の面積をAUC(Area Under the Curve)と呼ぶ。単にAUCと呼ばれる場合は多くの場合でROC曲線のAUCを示す。AUCが1に近いほど良い結果とされる。陽性，陰性の判定をランダムに行った場合結果はAUCは0.5に近づく。

- PR 曲線 (Precision Recall Curve)

予測値を閾値で陽性と陰性を判定した時縦軸を再現率、横軸を適合率とし閾値毎にプロットした曲線。特にテストデータの内実際に陽性であるデータが極端に少ない場合に有効。ROC 曲線の AUC のみで評価することができない場合などに使用される。ROC 曲線と同様に AUC が 1 に近いほど良い結果とされる。

第3章 基準ノード選択制回転を用いた動的Node Embedding:SPRDNE

本章では提案手法である SPRDNE:Node Embedding by Selective Procrustes Rotation for Dynamic Graph について具体的に説明する.

3.1 直交プロクラステス回転の時系列データへの適応の考察

プロクラステス回転は2つの点集合が合致するような回転を行うが各点それぞれの特性は考慮されていない. また, 既存手法 (tNodeEmbed) では全てのノードを基準に通常のプロクラステス回転を用いている. よって各々のノードの時間変化の度合いは考慮されていない. 図3.1では各ノードを表す点の大きさはノードの時間変化の大きさを表している. 図3.1の上部分はそのような, 各々のノードの時間変化の度合いは考慮されていない場合の例となっており, 点の大きさは関係なくただ対応する点の距離によって回転のさせ方が決まる. しかしながらグラフで表現される時系列データの場合, 隣接関係の変化のみに注目した場合であっても時間変化の中で隣接関係の変化しないノードがある一方, 他のノードと新たなエッジ持つノードやエッジを失うノードも存在する. このように各ノードはそれぞれ多様に変化し, 時間が経つにつれて特徴や性質が大きく変化するノードや, 時間によってあまり性質や特徴が変わらないノードが存在すると予想でき, 変化の度合いはノードそれぞれで異なる. 従って, プロクラステス回転を用いるとき, ノードそれぞれの特性, つまり今回の場合時間変化の大きさを考慮し回転をする必要があると考えた. つまり, 図3.1の下部分のような回転である. 上部分の回転に比べ時間変化の小さいノードがより近接するように回転されている. ここで課

題となるのは、そもそもノードの時間変化をどのように定義するかである。本研究ではノードの時間変化を定義するためにノード間の最短経路の変化を利用した。また、時間的変化の少ないノードがより近くなるような回転にするため回転の基準となるノードを時間変化の大きさを元に選択することを提案する。

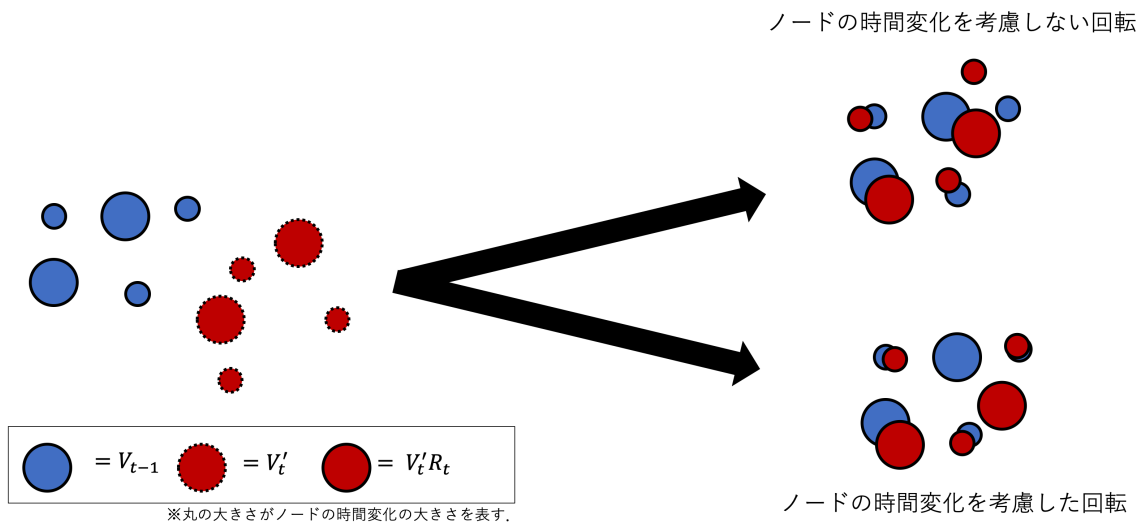


図 3.1: 概要図:時系列データに適した回転

3.2 提案手法

3.2.1 概要

提案手法の概要図を図3.2に示す。基本的流れは既存手法の tNodeEmbed と同様である。提案手法は以下のように大きく3つのステップに分かれる。

- 1: データを分割する。
- 2: 分割したデータごとに静的なノードエンベディングを行う。
- 3: 以下の二つのステップを $t=2$ から T まで繰り返す。
 - 3.1: プロクラステス回転に使用するノードを選択し回転行列を求める。
 - 3.2: 回転行列を用いて最終的なベクトル表現を得る。

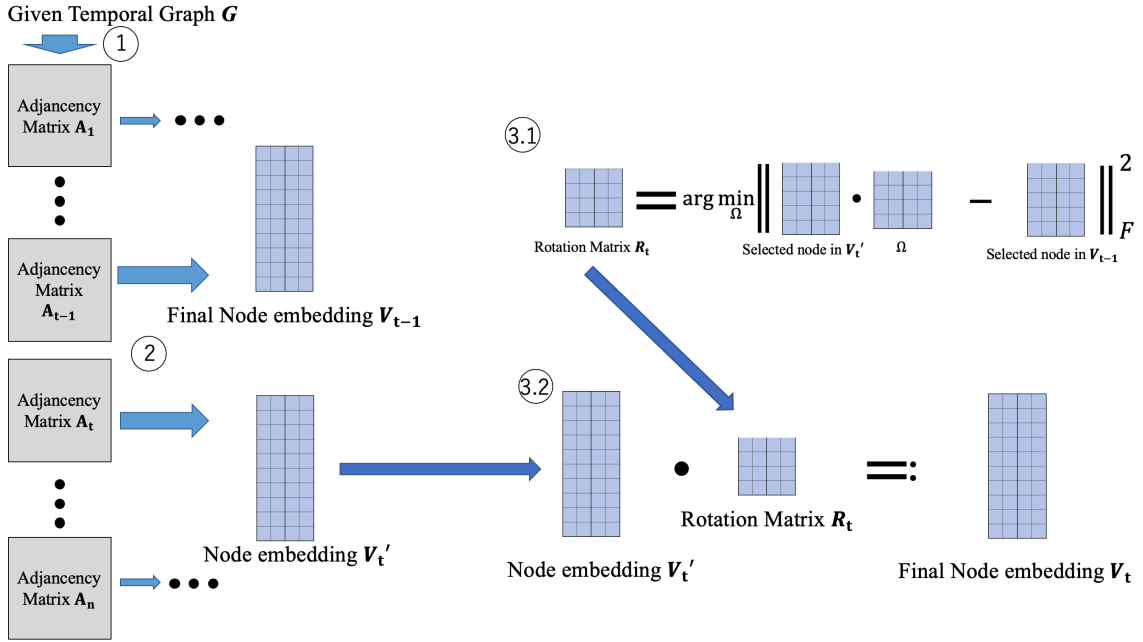


図 3.2: 概要図:提案手法

以降で各ステップについての詳細を説明する．ステップ1ではグラフのリンクの時間情報を元に各時刻のエッジの数または時間が可能な限り均一になるように分割し各時刻の隣接行列 A_1 から A_T までを生成する．時間を均一にする場合にはエッジの数が極端に偏る可能性がある為対象とするグラフによっては注意が必要である．ステップ2ではステップ1で生成した隣接行列を元に各時刻で静的な Node embedding を行い回転適応前のノードのベクトル表現 V_1 および V_2 から V_T を得る．この時，Node embedding の手法は任意のものを使用することができる．静的なグラフに対する Node embedding は動的なものとは比べ数多く提案されており，すでにある静的なグラフに対する手法を動的なグラフに応用できるという点は既存手法の tNodeEmbed の大きな利点である．本研究での実験等には Node embedding の代表的手法である Node2vec を採用した．最後にステップ3を $t=2$ から T まで繰り返す．ステップ3.1ではステップ2で得たベクトル表現に適応するための回転行列 R_t を求める．この回転行列を求める際には基準となるノードを定義した時間変化の大きさを元に選択し，選択したノードのみを基準に回転行列を求める．この回転基準となるノードの選択が本研究の主眼であり，新たに試みた部分となる． R_t は以下の式で表される．

$$R_t = \arg \min_{\Omega} \|\tilde{V}_t' \Omega - \tilde{V}_{t-1}'\|_F^2 \text{ subject to } \Omega^T \Omega = I_N \text{ (I:identity matrix)}$$

ここで $\tilde{\mathbf{V}}'_t, \tilde{\mathbf{V}}'_{t-1}$ はそれぞれ $\mathbf{V}'_t, \mathbf{V}'_{t-1}$ の中から選択されたノードのベクトル表現を表す. ステップ3.2では時間変化を反映させるために回転させた最終的なベクトル表現 \mathbf{V}_t を得る. \mathbf{V}_t は以下の式で表される.

$$\mathbf{V}_t = \mathbf{V}'_t \mathbf{R}_t$$

実際に時系列リンク予測等グラフの分析タスクを行う場合はこの \mathbf{V}_t を用いてLSTMなどの手法を適応する.

3.2.2 回転基準ノードの選択

本節では, 本研究の中心となる回転基準ノードの選択について述べる.

ノードの時間変化の定義

ノードの時間変化を定義するにおいて本研究では最短経路を利用した. 最短経路を利用することでノードの関係の変化のうち, 一次近接だけではなく二次近接及び高次近接の変化を捉えることができる. 数学的に次のように定式化した.

$$(s_{\text{dif}})_v := \frac{1}{|\hat{\nu}_t| - 1} \sum_{u \in \hat{\nu}_t \setminus \{v\}} \left| \frac{1}{s_t^{v,u}} - \frac{1}{s_{t-1}^{v,u}} \right| \quad (3.1)$$

比較には時刻間で共通のノードを使用する. 最短経路の値域は隣接ノードの1から経路が存在しない場合の無限大となる. そのため逆数を用いることで値域を $[0, 1]$ とした. 各要素の差の絶対値の和をノードの時間変化の大きさと定義した.

重みの検討

式(3.1)の定義の場合, ノード間の最短経路がどこで変化したと場合も同じ変化量として扱われる. 例として, あるノード間の最短経路の長さが1から2に変化した場合と10から11に変化した場合は同じ変化量1として扱われる. しかしながら, グラフ構造をもつデータでは一般的に近くに存在するノードが重要とされる. 従って, ノードの近傍での変化をより強調することが必要だと考えた. そこで最短経路の長さに対して重み

付けを行った．以降表記の簡略化の為 $\frac{1}{s_t^v} = s$ とする．例として半径1の円の右下部分をあげる．数式では以下のように表される．

$$W_c(s) = -\sqrt{1 - (s)^2} + 1.$$

式(3.2)の接線の傾きは $[0,1]$ の区間で単調増加のため最短距離が小さい数の時間変化は大きく，大きい値の時間変化は小さくする重みとなっている．これにより，ノードの近傍での変化をより強調することができる．また，一般的に最適な重みはデータごとに異なる為，パラメータで調整できた方がよい．従って，パラメータで調整できる重みとして次の二つを使用した．

$$W_G(s) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(s-1)^2}{\sigma^2}\right). \quad (\text{平均1の正規分布})$$

$$W_e(s) = e^{a(s-1)}. \quad ((1,1) \text{ を通過する指数関数})$$

W_G , W_e のパラメータはそれぞれ σ (分散), a となる． W_e の特徴としては最大値の上限が1となっているおり，パラメータによっては w_c に近い挙動をする． W_G の特徴としては最大値が変動し，また分散が大きい場合接線の傾きが変曲点まで単調増加する為，1次近接と2次近接の差が他の重みに比べ小さくなる．つまり，共通の隣接ノードを持つノードが多い場合等に有効と考えられる．また重みを含めたノードの時間変化の大きさは次のように定義される．

$$(s_{\text{dif}})_v := \frac{1}{|\hat{v}_t| - 1} \sum_{u \in \hat{v}_t \setminus \{v\}} \left| W\left(\frac{1}{s_t^{v,u}}\right) - W\left(\frac{1}{s_{t-1}^{v,u}}\right) \right| \quad (3.2)$$

重みがない場合と3種の重みに関する比較図を図3.3に示す．横軸は最短距離の長さを表しており縦軸は重みで処理後の出力である $W(s)$ を示す． W_G , W_e に関してはパラメータで調整可能な重みのため一例となる．

ノードの選択方法

本節では，定義したノードの時間変化を元にどのように選択するかについて述べる．具体的な方法として下の3つをあげる．

- 閾値

一定の閾値を設け，式(3.2)で定義したノードの時間変化が閾値以下

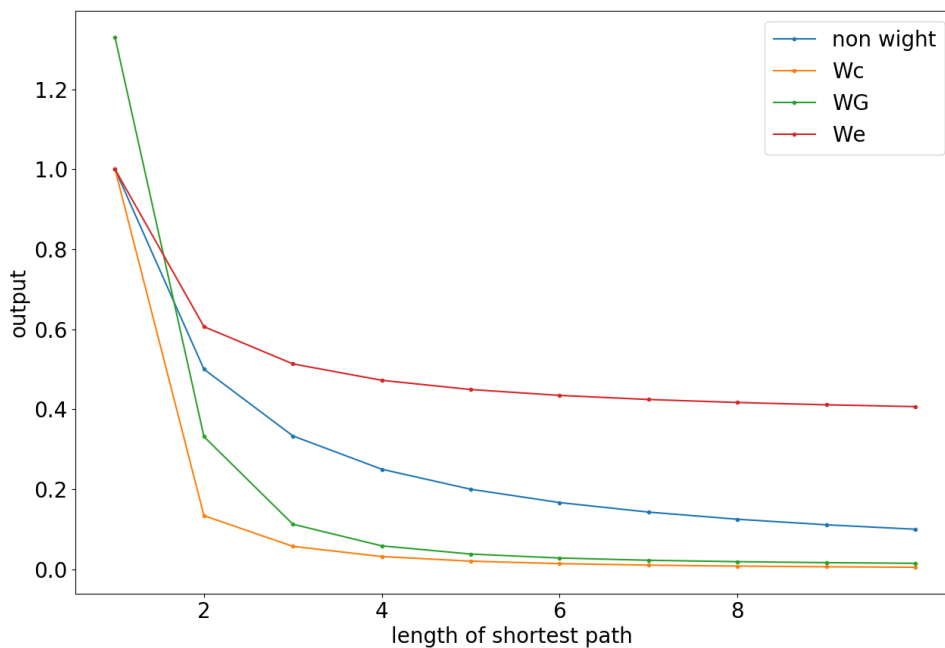


図 3.3: 重み関数の比較

の場合そのノードは選択される．閾値の値がパラメータとなる．3つの中で最も重みの影響を受けやすいが時間変化が0になる場合は重みの違いによって影響が出ない．

- 割合

一定の割合を設け，式(3.2)で定義したノードの時間変化が少ないノードを割合の分選択する．割合の値がパラメータとなる．閾値と比較して重みの影響を受けにくい．ノードの時間変化の大小関係が設定した割合付近で変化があった場合影響が出る．2つの連続した時刻間で共通に存在するノードの数によって選択されるノードの個数が変化する．

- 個数

任意の個数を決め，式(3.2)で定義したノードの時間変化が少ないノードを決めた数選択する．個数の値がパラメータとなる．同様に閾値と比較して重みの影響を受けにくい．ノードの時間変化の大小関係が設定した個数付近で変化があった場合影響が出る．2つの連続した時刻間で共通に存在するノードの数によって選択される

ノードの個数が変化しない．一方で割合は大きく変化する．

以降では上記の3つを選択方法，それぞれのパラメータを選択値と呼称する．

第4章 実験と評価

本章ではいくつかの数値実験を行い，提案手法の効果を示す．

4.1 実験概要

4.1.1 実験内容とその目的

性能評価実験

本実験は提案手法の性能を評価することを目的とし，基準ノードの選択を行わない場合と比較した．具体的には各手法で得たノードのベクトル表現を用いた時系列リンク予測により後の時刻の隣接行列を予測し，その結果を比較する．予測器として，時系列データの処理に優れたLSTMを使用した．ROC曲線のAUCの平均を評価指数とし，参考値としてPR曲線のAUCの平均を併記する．重み関数として，まず W_c を使用し，必要な場合追加実験として W_G ， W_e を使用し最も結果の数値が高い結果を記載した．1度の実験をベクトル表現を取得し予測タスクを3度行うこととし，各データセットに対して5度実験を行なった． W_c, W_G, W_e に関してはその式を以下に再記する．

$$\begin{aligned}W_c(s) &= -\sqrt{1-(s)^2} + 1. \text{ (半径1の円の右下部分)} \\W_G(s) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(s-1)^2}{\sigma^2}\right). \text{ (平均1の正規分布)} \\W_e(s) &= e^{a(s-1)}. \text{ ((1,1)を通過する指数関数)}\end{aligned}$$

その他実験

性能評価実験結果の考察の為，以下の項目についての値を実験で取得した．

- 各時刻の $|\hat{v}_t|$
 t および $t-1$ のどちらの時刻でもエッジを持つノードの数を取得した．

- 選択されたノードの割合
 $|\hat{v}_t|$ のうち回転の基準に使用されたノードの割合を取得した.
- 各時刻での \hat{v}_t に属すノードの時間変化の平均
式(3.2)で定義した $(s_{\text{dif}})_v$ の各時刻での平均を取得した.
- 単一時刻の各ノードの時間変化
ノードごとの $(s_{\text{dif}})_v$ の値を取得した. サンプルとして各データセットの中から一時刻分を示す.
- エポック数に対する訓練データおよび検証データの精度と損失
訓練データのうち1割を検証データとして利用し, 精度と損失を確認することで訂正なエポック数について検証した. ノードの選択を行わない既存手法で行った.

4.1.2 データセット

本実験では以下の実世界のデータセットを利用した. またデータセット”SFHH-conf-sensor”は2つのパラメータで実験した.

- collegeMsg [25] [26]
このデータセットは, カリフォルニア大学アーバイン校のオンラインソーシャルネットワークで送信されたプライベートメッセージで構成されている. ユーザーはネットワークで他のユーザーを検索し, プロファイル情報に基づいて会話を開始することができる. データセットには, 少なくとも1つのメッセージを送受信したユーザーが含まれる. ユーザーがノード, メッセージの送受信がリンクとして扱われる. テストデータのグラフを図4.1, サンプルとして時刻 $t=2$ のグラフを図4.2に示す.
- ia-reality-call [27] [28]
このデータセットは, マサチューセッツ工科大学の少数のコアユーザー間の人間の携帯電話の通話イベントで構成されている. これらのユーザーには, すべての通話が収集された携帯電話が実際に割り当てられている. データには, この少数のユーザーからアクティブに監視されていない他の個人の電話への呼び出しも含まれている.

そのため、これらのノードは通常、実験に参加して電話が割り当てられたMITの少数のユーザー内のノードよりもエッジが少ない。データは、RealityCommonsプロジェクトの一環として2004年に実施されたRealityMining実験によって100台の携帯電話を使用して9か月にわたって収集された。この実験の目的は、社会学者が従来の調査ベースの方法論または従来のシミュレーションベースの方法論を超えて人間の相互作用を調査できるようにするスマートフォンの機能を調査することで、MITメディアラボの75人の学生または教職員と、メディアラボに隣接するMITスローンビジネススクールの25人の新生を対象とした。75人のメディアラボ参加者のうち、20人は修士課程の学生で、5人はMITの新生で、残りは少なくとも1年間メディアラボに留まった。ノードは人を表し、エッジは、2人のユーザー間の電話またはボイスメールを示す。テストデータのグラフを図4.3、サンプルとして時刻 $t=2$ のグラフを図4.4に示す。このデータセットは選択方法の影響を受けやすいため割合と閾値の二つの選択方法で実験を行った。

- SFHH-conf-sensor [29] [28]

このデータセットは、フランスのニースで開催されたカンファレンス、SFHH(2009年6月4~5日)への405人の参加者の対面のやり取りを示している。これは、分散RFIDセンサーネットワークからの人と人の相互作用のダイナミクスと、会議出席者の動的接触ネットワークでのSEIR感染症モデルのシミュレーションの出版物で使用されており、出版物の補足データセットとしてリリースされた。人をノードとし、対面のやり取りをリンクとしている。テストデータのグラフを図4.1、サンプルとして時刻 $t=3$ のグラフを図4.2に示す。

その他データセットの詳細について表(4.1)に記す。

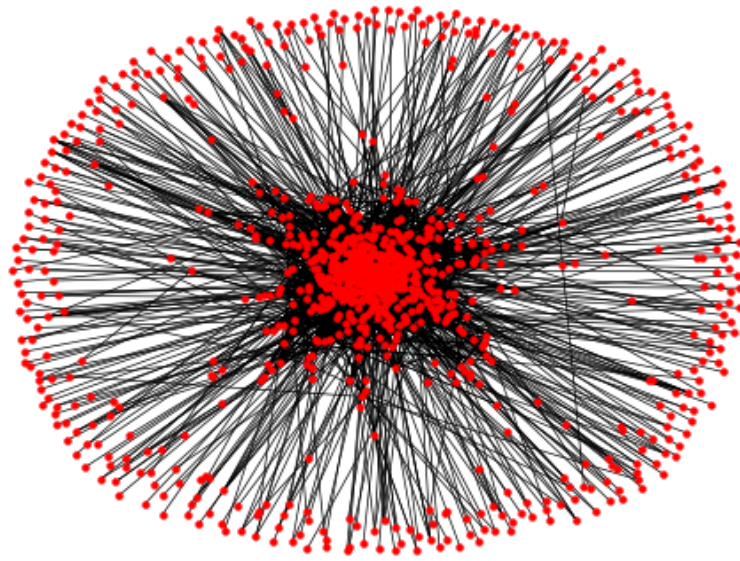


図 4.1: collegeMsg におけるテストデータのグラフ

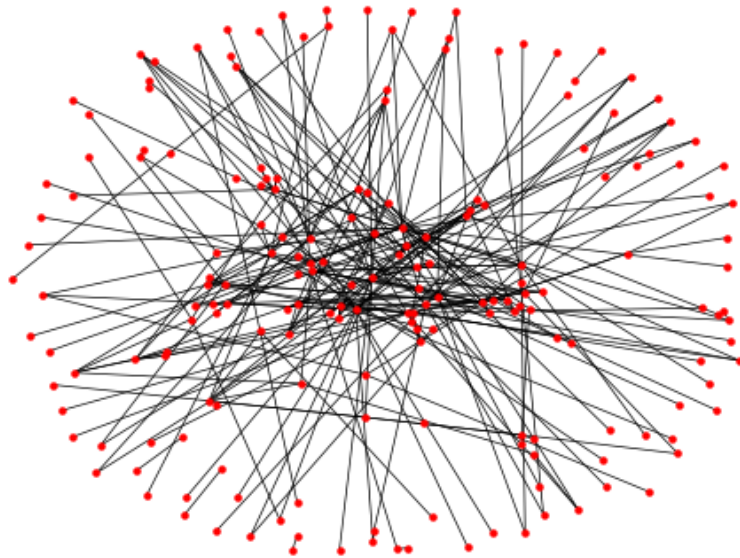


図 4.2: collegeMsg における時刻 $t = 2$ のグラフ

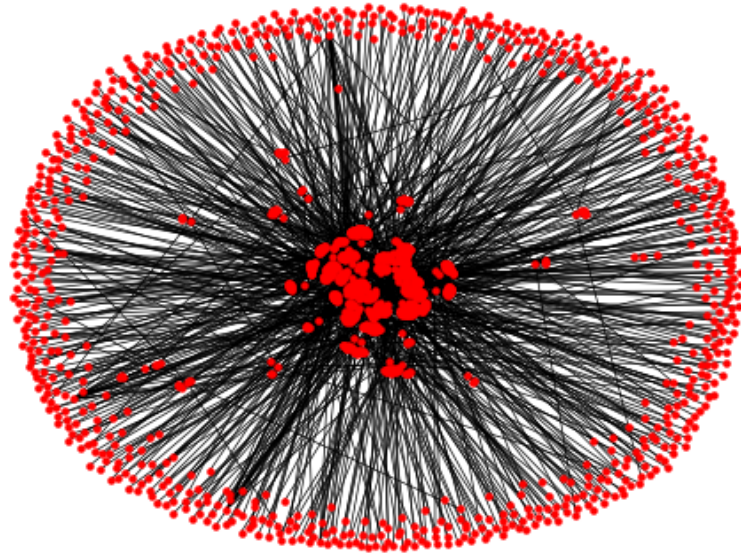


図 4.3: ia-reality-call におけるテストデータのグラフ

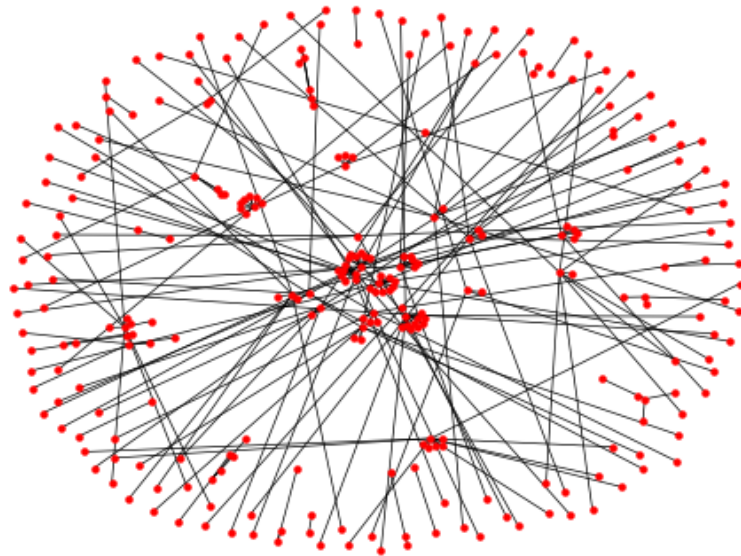


図 4.4: ia-reality-call における時刻 $t = 2$ のグラフ

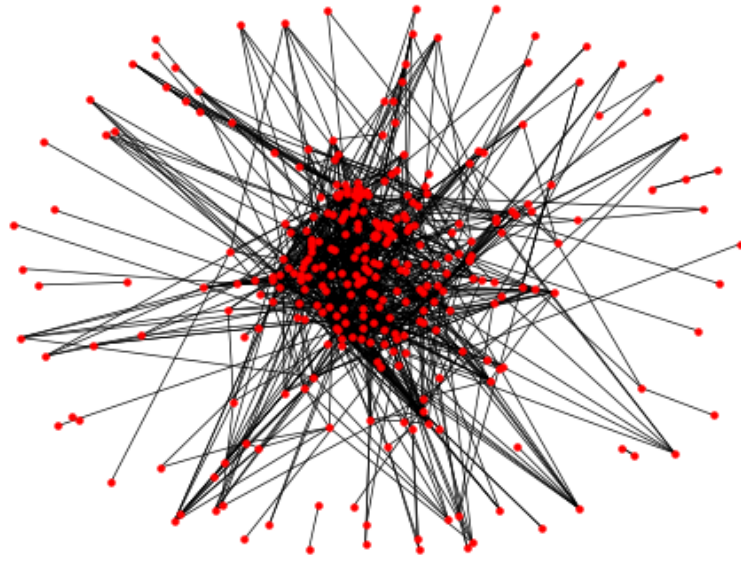


図 4.5: SFHH-conf-sensor におけるテストデータのグラフ

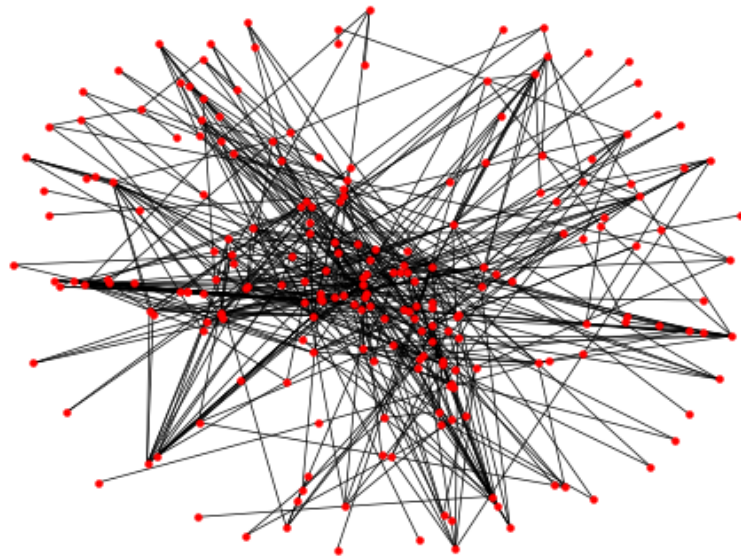


図 4.6: SFHH-conf-sensor における時刻 $t=3$ のグラフ

表 4.1: データセットの詳細

データセット名	ノード数	エッジ数	期間	種類
collegeMsg	1899	59835	193日間	有向
ia-reality-call	6800	52050	9ヶ月間	無向
SFHH-conf-sensor	403	70261	1日	無向

4.1.3 実験条件

表 4.2: 実験パラメータ

データセット名	分割数	次元数	参照時間	選択方法	選択値	エポック数
collegeMsg	50	8	20	個数	8	75
ia-reality-call	30	32	15	閾値	0	20
SFHH-conf-sensor	20	64	10	割合	0.5	15
SFHH-conf-sensor	20	64	10	閾値	0.2	15

表 4.3: 分割後のデータセットの諸数値

データセット名	collegeMsg	ia-reality-call	SFHH-conf-sensor
平均ノード数	248.8	250.2	181.5
平均エッジ数	322.3	204.4	427.7
最大ノード数	320	264	314
最大エッジ数	327	213	1474
最小ノード数	181	231	90
最小エッジ数	320	203	110
テストノード数	1006	1504	313
テストエッジ数	4131	1537	1109

データセットのエッジの後ろの時刻の2割をテストデータの正例とし、負例は同じ数ランダムで選ばれたエッジを使用した。静的なノードエンベディングの手法は確率に基づいた手法(ランダムウォーク)を利用したNode2vec [3]を使用した。この時、Node2vecの入力には各時刻でエッジを持つノードのみとした。また、未出現のノードの特徴的は同じ次元数を持つゼロベクトル、消滅中のノードは最後に存在していた時刻の特徴量で補完した。特徴量をプロクラステス回転適用前の各時刻で $[-1, 1]$ で正規化した。LSTMは単一ノードの参照時間分の特徴量を1つするために

利用し、得られた特徴量を全結合型のニューラルネットワークの入力とし最終的な予測値を求めた。LSTMはバッチサイズを32, 学習率を0.001とした。エポック数は各データセット毎に決定した。テストデータのみが存在するノードは除外した。その他実験に各データセットで使用したパラメータは以下の表4.2に記す。また、表4.2に記載された分割数で分割した場合のデータセットに関するいくつかの数値を以下に示す。

4.2 実験結果および考察

4.2.1 collegeMsg

性能評価実験

結果を表4.4, 表4.5に示す。

表 4.4: collegeMsg の比較実験結果。(平均±標準偏差)を用いた。(Val: Best, Val: Best in one experiment)

回数	重み関数	ROC-AUC	PR-AUC
1	選択なし	0.817 ± 0.009	0.721 ± 0.017
	W_c	<u>0.830</u> ± 0.011	<u>0.739</u> ± 0.029
2	選択なし	0.826 ± 0.002	0.734 ± 0.006
	W_c	0.833 ± 0.005	<u>0.739</u> ± 0.016
3	選択なし	<u>0.831</u> ± 0.010	0.741 ± 0.019
	W_c	0.830 ± 0.005	0.740 ± 0.006
4	選択なし	0.818 ± 0.004	0.726 ± 0.019
	W_c	<u>0.825</u> ± 0.007	<u>0.731</u> ± 0.011
5	選択なし	0.820 ± 0.010	<u>0.727</u> ± 0.018
	W_c	<u>0.823</u> ± 0.010	0.718 ± 0.014

表 4.5: 表4.4の全結果の平均ならびに標準偏差 (Val: Best)

重み関数	ROC-AUC	PR-AUC
選択なし	0.823 ± 0.006	0.730 ± 0.008
W_c	0.828 ± 0.004	0.733 ± 0.009

はじめにROC-AUCの値を比較する。個別では5回のうち4回の実験で提案手法の基準ノードの選択をした場合の値が高かった。また、唯一選択

なしの場合が高い数値を3回目の結果の差は標準偏差が選択をする場合と選択をしない場合でそれぞれ0.01, 0.005に対して0.001だった。さらに最も差がついた1回目の実験では同様に0.009, 0.011に対して0.013となった。加えて最も高い数値は0.833で提案手法が記録した。平均でも僅かながら選択した場合の方が高い数値を記録した。また, PR-AUCはどの場合も極端に低い数値は示さなかった。

その他実験結果

- 各時刻の $|\hat{v}_t|$

図4.7に結果を示す。横軸が時刻 (timestep), 縦軸は $|\hat{v}_t|$ を表す。このデータセットでは僅かながら \hat{v}_t の数は減少傾向にあることがわかる。

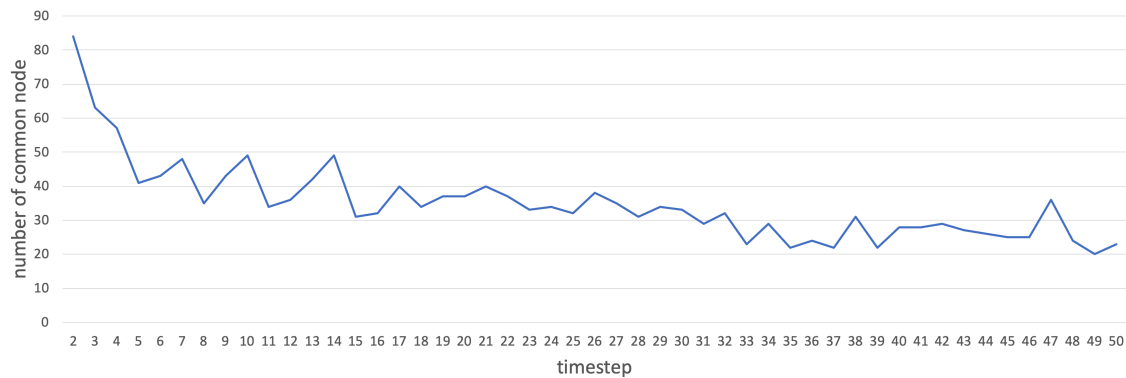


図 4.7: \hat{v}_t の数:collegeMsg

- 選択されたノードの割合

図4.8に結果を示す。横軸が時刻 (timestep), 縦軸は \hat{v}_t うちの選択されたノードの割合を示す。選択方法の特性と \hat{v}_t の数は減少傾向が影響し僅かな増加傾向が見て取れる。

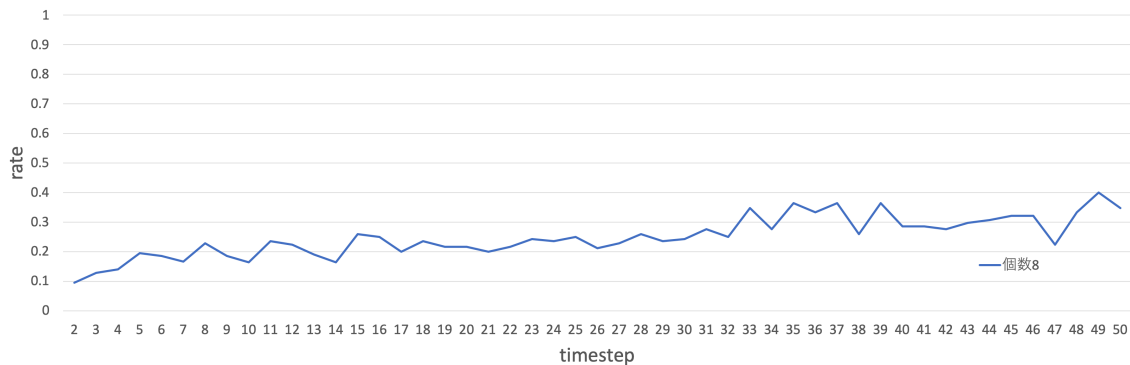


図 4.8: 選択されたノードの割合:collegeMsg(W_c)

- 各時刻での \hat{v}_t に属すノードの時間変化の平均

図 4.9 に結果を示す。横軸が時刻 (timestep), 縦軸がその時刻に存在する全てのノードの式 (3.2) で定義した $(s_{\text{dif}})_v$ の平均を表す。どの時刻でも一定程度の時間変化を持つことがわかる。

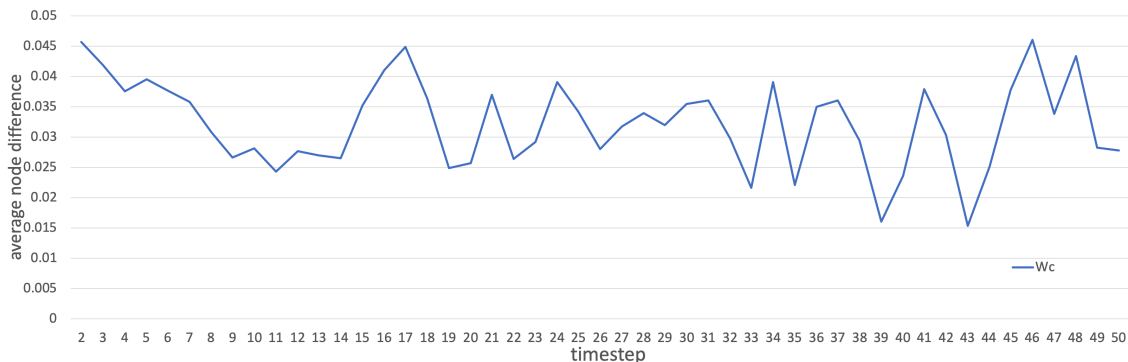


図 4.9: 時間変化の平均:collegeMsg

- 単一時刻の各ノードの時間変化

図 4.10 に結果を示す。横軸がノード, 縦軸が式 (3.2) で定義した $(s_{\text{dif}})_v$ を表す。比較的 time 変化が大きいノードも確認できる。一方で時間変化が 0 となるノードもノードの選択個数以上存在している。

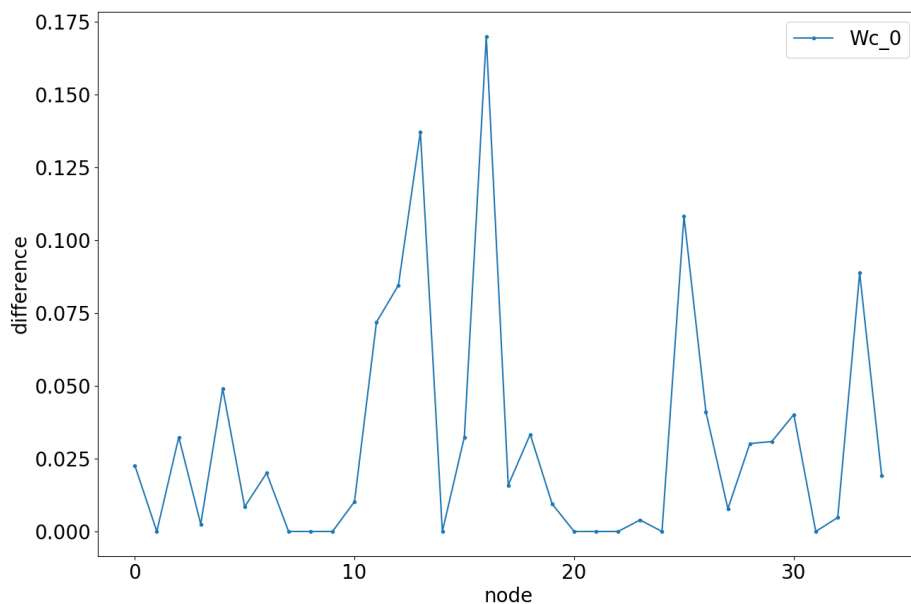


図 4.10: ノード毎の時間変化の比較:collegeMsg(t=8)

- エポック数に対する訓練データおよび検証データの精度と損失
 図 4.11, 図 4.12 に結果を示す. 横軸はエポック数は縦軸はそれぞれ精度と損失の値を表す. 凡例の train が訓練データを表し, validation が検証データを表す. 訓練データでの損失や精度はそれぞれ減少傾向や増加傾向が見られるものの検証データに関しては一定程度で高止まりすることが確認できる.

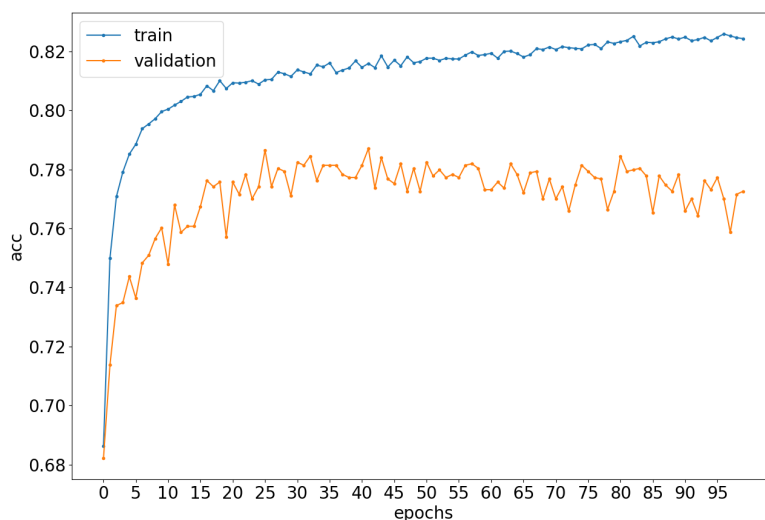


図 4.11: エポック数に対する精度の変化:collegeMsg

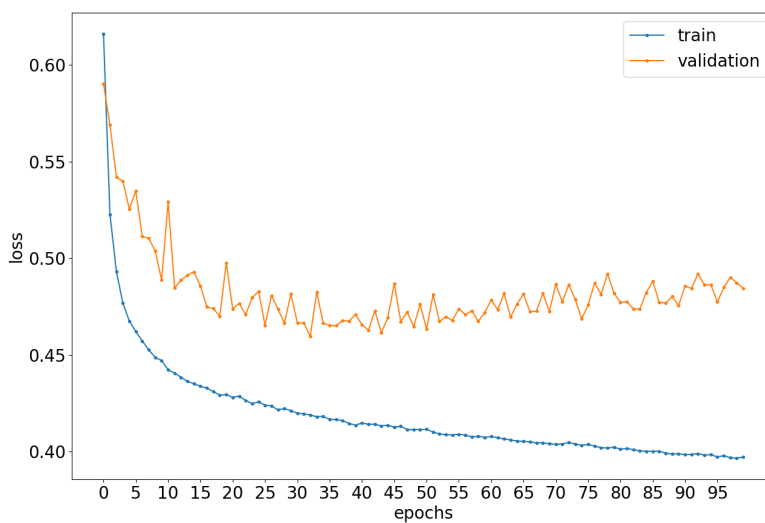


図 4.12: エポック数に対する損失の変化:collegeMsg

考察

性能評価実験での結果から平均値に加え個別に見ても多くの場合で提案手法が高い値を記録している。また今回の実験で唯一既存手法が上回った3回目の実験での差は他の実験よりも小さい。これらのことから提案手法のノードの選択が有効と言える。図4.10から時間変化が0になるノードが11個確認できる。今回の選択方法である個数指定では次元数と同じ8個のノードを指定しており、重みの影響を受けないノードが図4.10のようにほとんどの時刻でそれ以上のため、重み W_c を用いた実験のみ行った。一方でノード毎に比較すると時間変化の大きさの差は顕著であり、時間変化の大きいノードを回転基準とするノードから除外できたことで精度が向上したと考えられる。エポック数に着目すると図4.11, 図4.12から50から80に近いエポック数が必要であることが確認できる。

4.2.2 ia-reality-call

性能評価実験結果

結果を表 4.6, 表 4.6 に示す.

表 4.6: ia-reality-call の比較実験結果. (平均±標準偏差)を用いた. (Val: Best, Val: Best in one experiment)

回数	重み関数	ROC-AUC	PR-AUC
1	選択なし	0.981 ± 0.006	0.978 ± 0.009
	W_c	0.990 ± 0.005	0.993 ± 0.004
2	選択なし	0.977 ± 0.008	0.979 ± 0.003
	W_c	<u>0.987 ± 0.008</u>	<u>0.984 ± 0.016</u>
3	選択なし	0.977 ± 0.011	0.968 ± 0.017
	W_c	<u>0.984 ± 0.005</u>	<u>0.975 ± 0.014</u>
4	選択なし	0.980 ± 0.009	0.963 ± 0.021
	W_c	<u>0.984 ± 0.006</u>	<u>0.973 ± 0.018</u>
5	選択なし	0.977 ± 0.014	0.970 ± 0.017
	W_c	<u>0.978 ± 0.009</u>	<u>0.975 ± 0.022</u>

表 4.7: 表 4.6 の全結果の平均ならびに標準偏差 (Val: Best)

重み関数	ROC-AUC	PR-AUC
選択なし	0.978 ± 0.002	0.972 ± 0.007
W_c	0.985 ± 0.005	0.980 ± 0.009

同様に ROC-AUC の値を比較する. 個別では 5 回のうち全ての実験で提案手法の基準ノードの選択をした場合の方が高い値を記録し, 平均ではその差は 0.007 となった. 最も高い数値を記録したのは提案手法の 1 回目の実験で 0.990 を記録した. 最も低い数値を記録したのは選択なしの場合の 2, 3, 5 回目で 0.977 を記録した. また, 最も差が大きいのは 2 回目の実験で標準偏差がそれぞれ 0.008 なのに対して 0.01 の差があった. また, PR-AUC はどの場合も極端に低い数値は示さなかった.

その他実験結果

- 各時刻の $|\hat{\nu}_t|$

図 4.13 に結果を示す. 横軸が時刻 (timestep), 縦軸は $|\hat{\nu}_t|$ を表す. どの

時刻でも一定のノードが存在していることがわかる．局所的には僅かな増加傾向や減少傾向は見られるが全体として一定の傾向は確認できない．

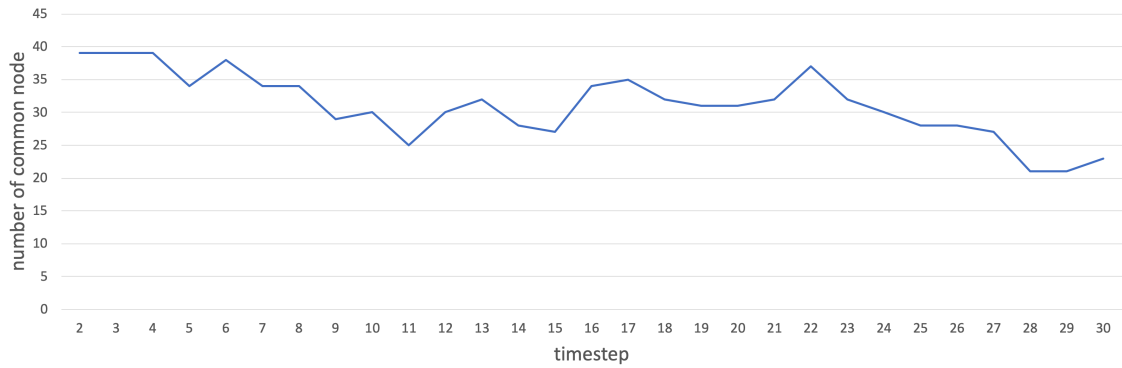


図 4.13: \hat{v}_t の数:ia-reality-call

- 選択されたノードの割合

図 4.14 に結果を示す．横軸が時刻 (timestep), 縦軸はでは \hat{v}_t うちの選択されたノードの割合を示す．選択方法として閾値を用い値を 0 事からこのデータセットでは時間変化のないノードが多くを占める事がわかる．

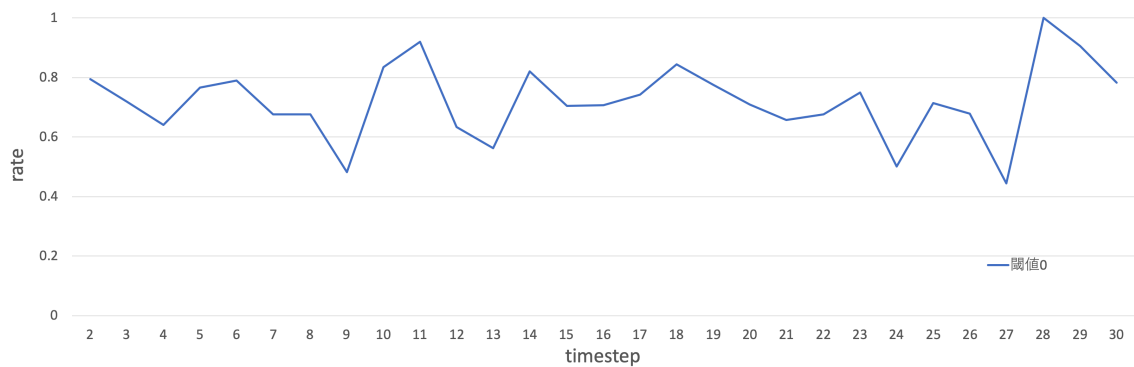


図 4.14: 選択されたノードの割合:ia-reality-call(W_c)

- 各時刻での \hat{v}_t に属すノードの時間変化の平均

図 4.15 に結果を示す。横軸が時刻 (timestep), 縦軸がその時刻に存在する全てのノードの式 (3.2) で定義した $(s_{\text{dif}})_v$ の平均を表す。時間変化の平均が全体的に 0 に近く、時間変化の小さいノードが多いことがわかる。

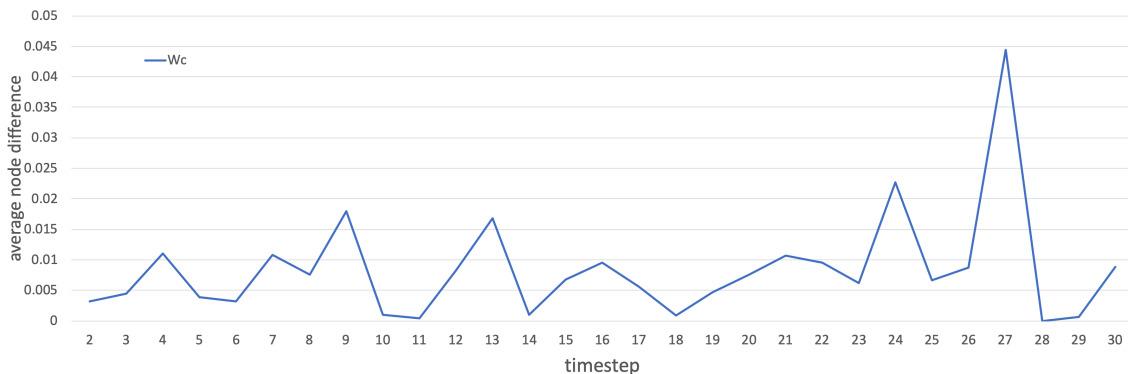


図 4.15: 時間変化の平均:ia-reality-call

- 単一時刻の各ノードの時間変化

図 4.16 に結果を示す。横軸がノード, 縦軸が式 (3.2) で定義した $(s_{\text{dif}})_v$ を表す。

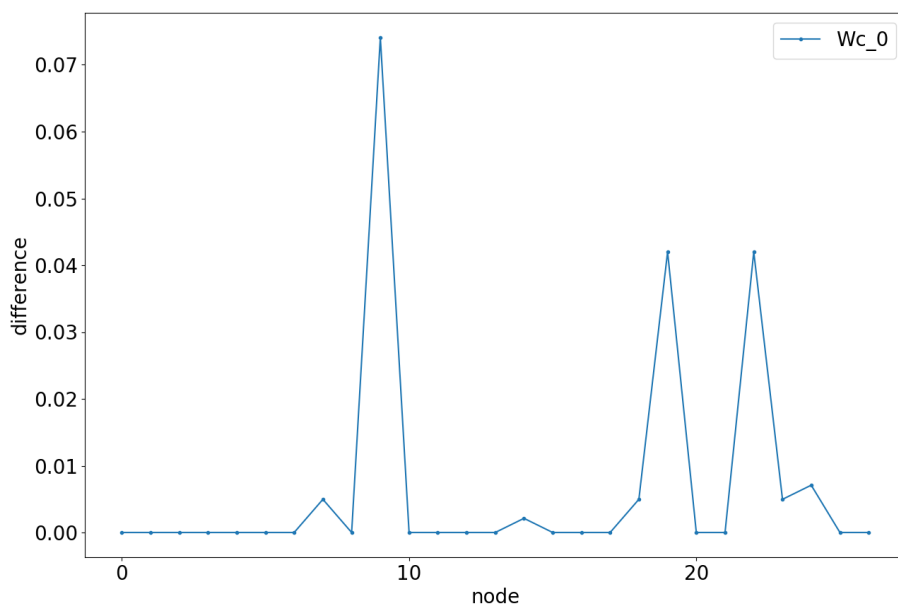


図 4.16: ノード毎の時間変化の比較:ia-reality-call(t=15)

- エポック数に対する訓練データおよび検証データの精度と損失
 図 4.11, 図 4.12 に結果を示す. 横軸はエポック数は縦軸はそれぞれ精度と損失の値を表す. 凡例の train が訓練データを表し, validation が検証データを表す.

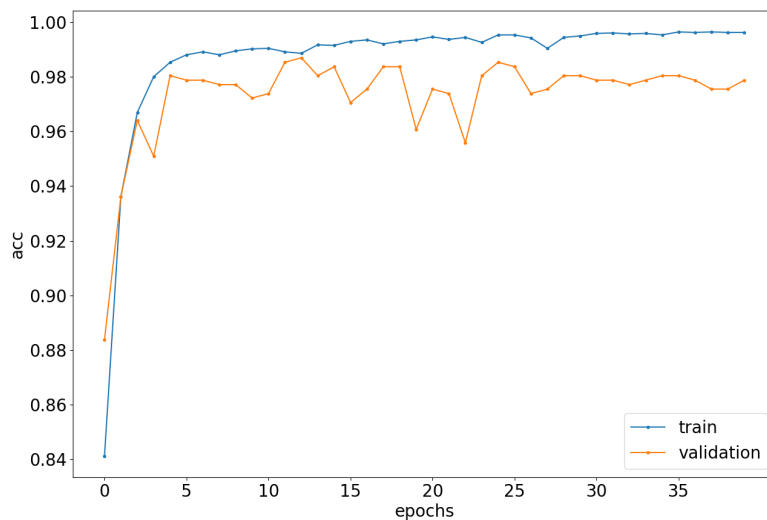


図 4.17: エポック数に対する精度の変化:ia-reality-call

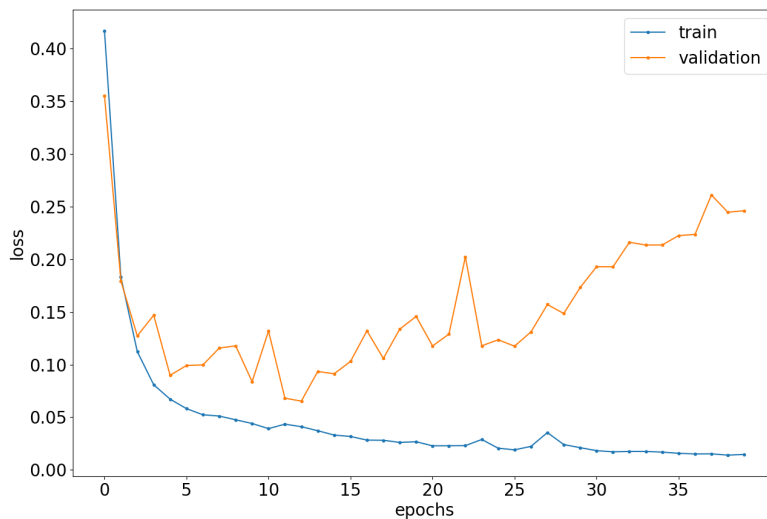


図 4.18: エポック数に対する損失の変化:ia-reality-call

考察

性能評価実験での結果から平均値に加え個別に見ても提案手法が高い値を記録している。よって提案手法のノードの選択が有効と言える。図4.16から時間変化が0になるノードが多く確認できる。今回の選択方法で閾値を採用しその値を0としていることから重みは影響を与えないことがわかる。そのため重み W_c を用いた実験のみ行った。ノード毎に比較すると時間変化が0となるノードが多く、図4.15を他のデータセットと比べてもわかるように全体的に時間変化の少ないデータセットであることがわかる。よって式(3.2)で定義した時間変化のないノードを回転基準とするノードから除外できたことで精度が向上したと考えられる。エポック数に着目すると図4.17, 図4.18から15から25に近いエポック数が必要であることが確認できる。

4.2.3 SFHH-conf-sensor

性能評価実験

選択方法に割合を用い割合を0.5とした場合の結果を表4.8, 表4.9に, 閾値を用い割合を値を0.2とした場合の結果を表4.10, 表4.11に示す.

表 4.8: SFHH-conf-sensor(割合 0.5) の比較実験結果. (平均±標準偏差)を用いた. (**Val**: Best, Val: Best in one experiment)

回数	重み関数	ROC-AUC	PR-AUC
1	選択なし	0.721 ± 0.012	0.700 ± 0.010
	W_c	0.719 ± 0.033	0.689 ± 0.038
	W_e	<u>0.736</u> ± 0.016	0.689 ± 0.026
	W_G	0.729 ± 0.013	<u>0.701</u> ± 0.020
2	選択なし	0.708 ± 0.006	0.679 ± 0.009
	W_c	0.714 ± 0.022	0.678 ± 0.024
	W_e	0.725 ± 0.015	0.694 ± 0.026
	W_G	<u>0.737</u> ± 0.003	<u>0.697</u> ± 0.007
3	選択なし	0.730 ± 0.005	0.705 ± 0.007
	W_c	0.720 ± 0.019	0.678 ± 0.018
	W_e	<u>0.740</u> ± 0.002	<u>0.720</u> ± 0.004
	W_G	0.730 ± 0.005	0.701 ± 0.008
4	選択なし	0.716 ± 0.012	0.669 ± 0.019
	W_c	0.737 ± 0.016	0.711 ± 0.016
	W_e	0.741 ± 0.007	0.717 ± 0.013
	W_G	0.741 ± 0.022	0.715 ± 0.032
5	選択なし	0.713 ± 0.021	0.678 ± 0.021
	W_c	0.714 ± 0.014	0.673 ± 0.022
	W_e	<u>0.734</u> ± 0.006	<u>0.704</u> ± 0.016
	W_G	0.733 ± 0.009	0.698 ± 0.013

表 4.9: 表 4.8 の全結果の平均ならびに標準偏差 (**Val**: Best, Val: Second best)

重み関数	ROC-AUC	PR-AUC
選択なし	0.717 ± 0.008	0.686 ± 0.015
W_c	0.721 ± 0.009	0.686 ± 0.015
W_e	0.735 ± 0.006	0.704 ± 0.014
W_G	<u>0.734</u> ± 0.005	<u>0.702</u> ± 0.007

表 4.10: SFHH-conf-sensor(閾値0.2)の比較実験結果. (平均±標準偏差)を用いた. (Val: Best, Val: Second best)

回数	重み関数	ROC-AUC	PR-AUC
1	選択なし	<u>0.740</u> ± 0.003	0.707 ± 0.003
	W_c	0.722 ± 0.017	0.685 ± 0.020
	W_e	0.735 ± 0.015	0.708 ± 0.015
	W_G	0.739 ± 0.014	0.718 ± 0.013
2	選択なし	0.715 ± 0.006	0.683 ± 0.005
	W_c	0.711 ± 0.009	0.671 ± 0.020
	W_e	<u>0.730</u> ± 0.013	<u>0.700</u> ± 0.009
	W_G	0.726 ± 0.007	0.699 ± 0.009
3	選択なし	0.717 ± 0.008	0.697 ± 0.004
	W_c	0.723 ± 0.020	0.697 ± 0.025
	W_e	0.725 ± 0.011	0.686 ± 0.026
	W_G	<u>0.734</u> ± 0.005	<u>0.705</u> ± 0.008
4	選択なし	0.715 ± 0.013	0.682 ± 0.017
	W_c	0.730 ± 0.018	0.708 ± 0.016
	W_e	0.721 ± 0.018	0.692 ± 0.030
	W_G	<u>0.735</u> ± 0.006	<u>0.713</u> ± 0.007
5	選択なし	0.712 ± 0.008	0.696 ± 0.017
	W_c	0.724 ± 0.009	0.694 ± 0.008
	W_e	0.726 ± 0.021	0.688 ± 0.025
	W_G	0.743 ± 0.013	<u>0.710</u> ± 0.008

表 4.11: 表 4.10 の全結果の平均ならびに標準偏差 (Val: Best, Val: Second best)

重み関数	ROC-AUC	PR-AUC
選択なし	0.719 ± 0.012	0.693 ± 0.010
W_c	0.722 ± 0.007	0.691 ± 0.014
W_e	<u>0.727</u> ± 0.005	<u>0.694</u> ± 0.009
W_G	0.735 ± 0.006	0.708 ± 0.008

表 4.8, 表 4.9 について述べる. まず, ROC-AUC の値を比較する. 各実験での最高値に着目すると重み W_e が 3 回, W_G が 2 回となった. また, 全体の最高値は 4 回目の W_G が記録し 0.741 であった. 各実験での最低値に着目すると選択なしが 3 回, 重み W_c が 2 回となった. 全体での最低値は 5 回目の選択なしであり 0.713 であった. また平均で比較すると最高値は W_e の 0.735 であり最低値は選択なしで 0.717 となった. 選択なしの場合と W_c ではその差が 0.004 だった. 一方で, 選択なしと W_e の差は 0.018, W_G との差

は0.017であった．次に表4.10, 表4.11 同様に, ROC-AUCの値を比較する．各実験での最高値に着目すると選択なしが1回, 重み W_e が1回, W_G が3回となった．また, 全体の最高値は4回目の W_G が記録し0.743であった．各実験での最低値に着目すると選択なしが3回, 重み W_e が2回となった．全体での最低値は2回目の W_e であり0.711であった．また平均で比較すると最高値は W_e の0.735であり最低値は選択なしで0.719となった．選択なしの場合と W_e ではその差が0.003だった．一方で, 選択なしと W_G の差は0.016, W_e との差は0.008であった．(割合0.5)と比較すると W_e が大きな影響を受けており0.008の低下が見られた．また, どちらもPR-AUCはどの場合も極端に低い数値は示さなかった．

その他実験結果

- 各時刻の $|\hat{v}_t|$

図4.19に結果を示す．横軸が時刻 (timestep), 縦軸は $|\hat{v}_t|$ を表す．

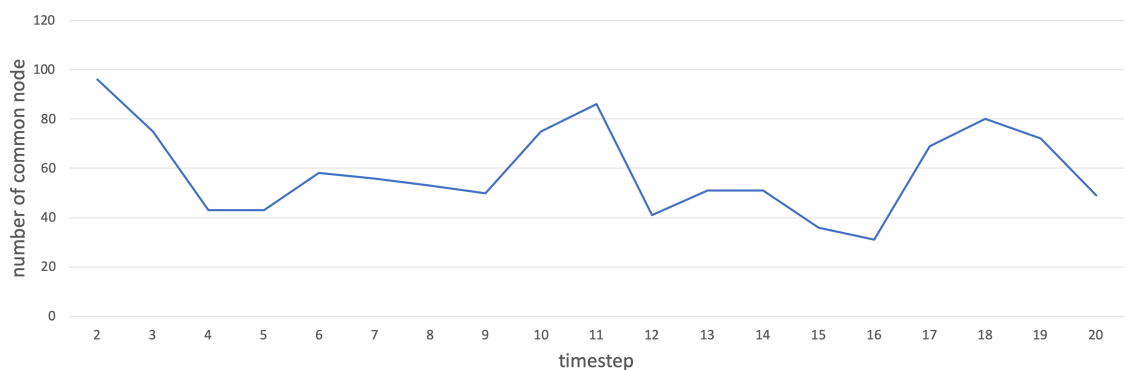


図 4.19: \hat{v}_t の数

- 選択されたノードの割合

図4.20に結果を示す．横軸が時刻 (timestep), 縦軸はでは \hat{v}_t うちの選択されたノードの割合を示す．

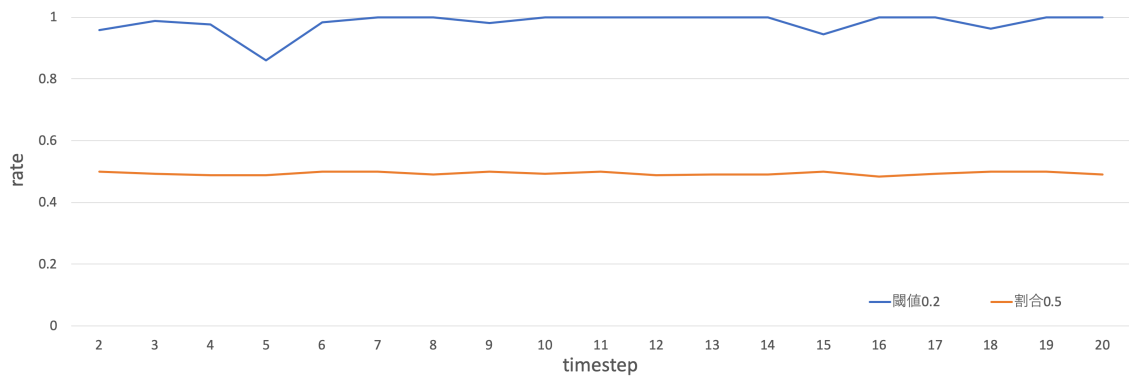


図 4.20: 選択されたノードの割合:SFHH-conf-sensor(W_c)

- 各時刻での \hat{v}_t に属すノードの時間変化の平均

図 4.21, 4.22 に結果を示す. 横軸が時刻 (timestep), 縦軸がその時刻に存在する全てのノードの式 (3.2) で定義した $(s_{\text{dif}})_v$ の平均を表す.

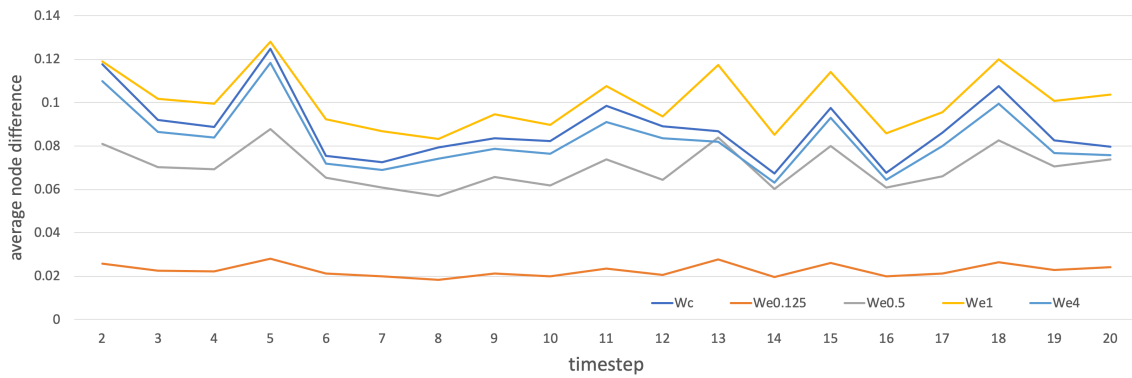


図 4.21: 時間変化の平均 (W_e)

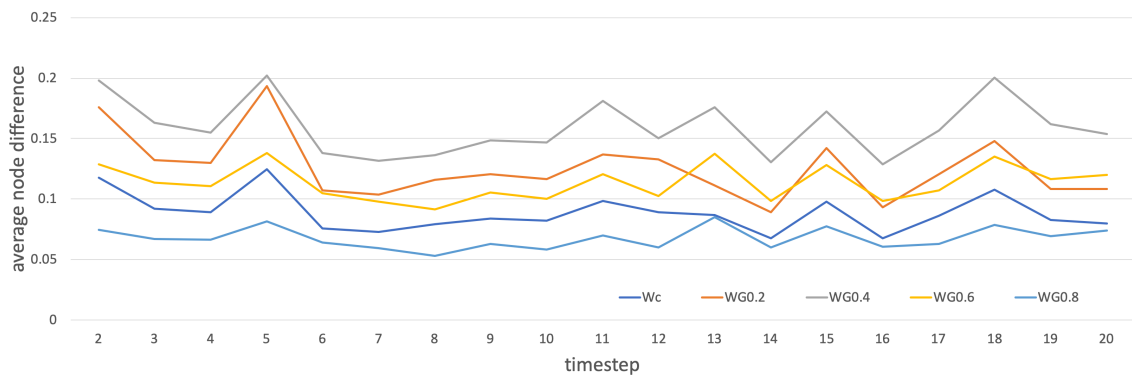


図 4.22: 時間変化の平均 (W_G)

- 単一時刻の各ノードの時間変化

図 4.23, 図 4.24 に結果を示す. 横軸がノード, 縦軸が式 (3.2) で定義した $(s_{\text{dif}})_v$ を表す.

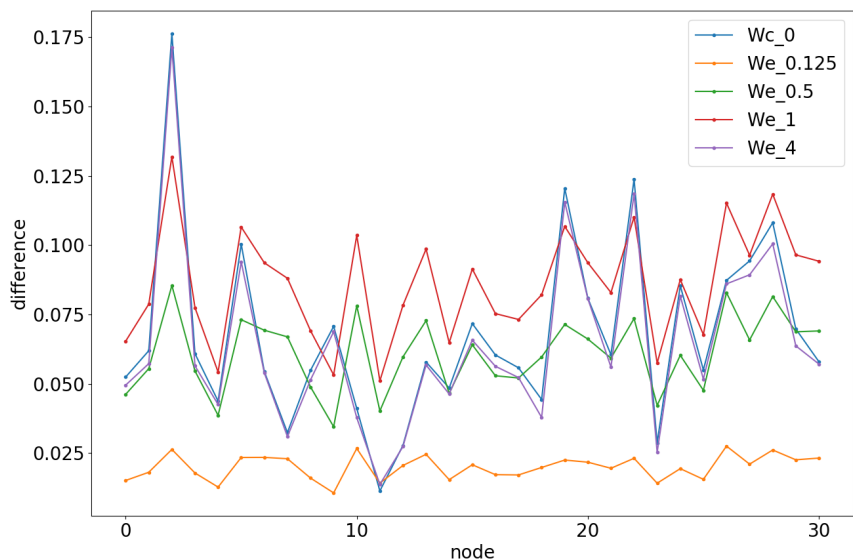


図 4.23: ノード毎の時間変化の比較 (We):SFHH-conf-sensor($t=16$)

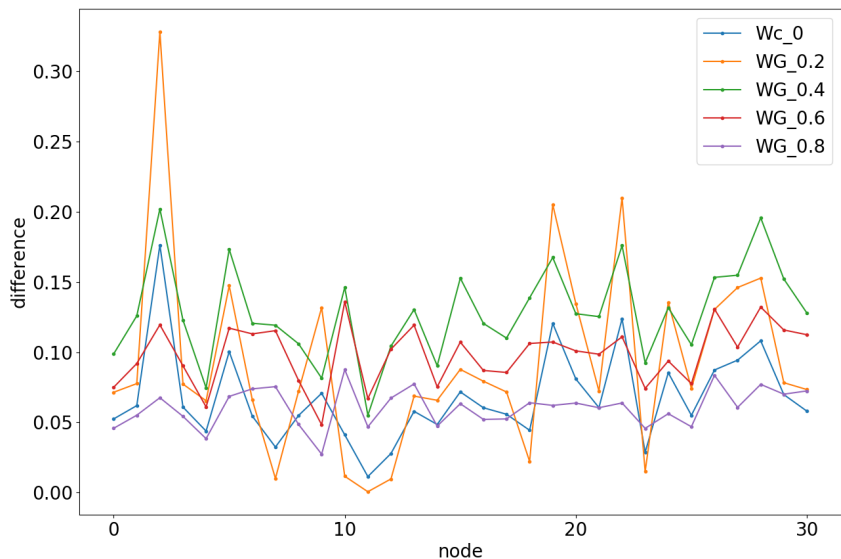


図 4.24: ノード毎の時間変化の比較 (WG):SFHH-conf-sensor($t=16$)

- エポック数に対する訓練データおよび検証データの精度と損失

図 4.25, 図 4.26 に結果を示す. 横軸はエポック数は縦軸はそれぞれ

精度と損失の値を表す．凡例の train が訓練データを表し， validation が検証データを表す．

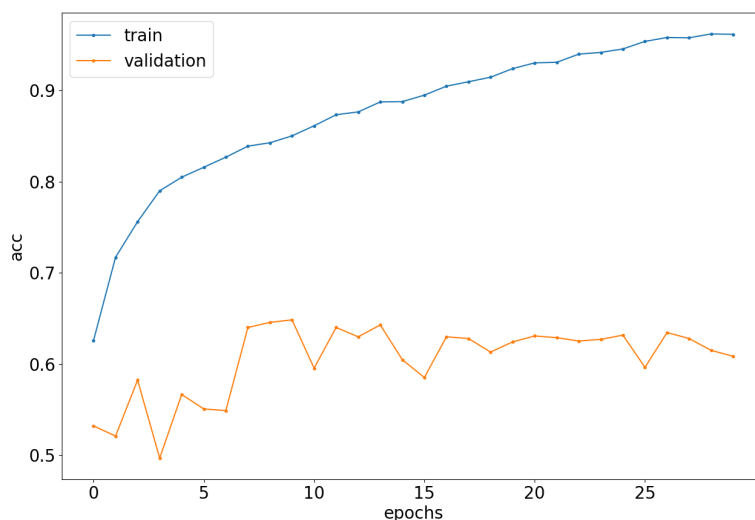


図 4.25: エポック数に対する精度の変化:SFHH-conf-sensor

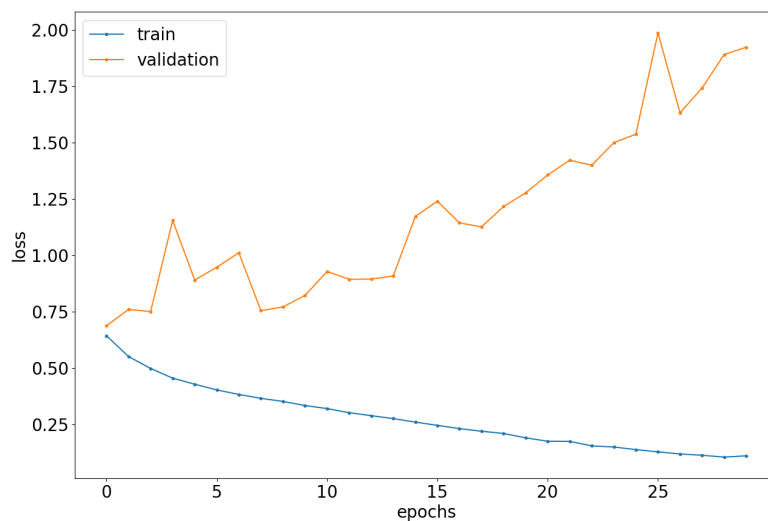


図 4.26: エポック数に対する損失の変化:SFHH-conf-sensor

考察

性能評価実験での結果から平均値に加え個別に見ても提案手法が高い値を記録している．よって提案手法のノードの選択が有効と言える．特にパラメータにより調節可能な重みの W_G , W_e は選択なしや重み W_e と比

べてより高い数値を記録しており，効果的に作用したと言える．図4.24, 図4.23から時間変化が0になるノードが少なく，図4.22, 図4.21から $(s_{\text{dif}})_v$ の値が他のデータセットと比べて大きいことからパラメータにより調節可能な重みの W_G , W_e が有効であることが予想でき，またそれが性能評価実験の結果に反映されていると考えられる．選択方法の閾値と割合の性能評価実験での差は W_e では顕著に確認できたもののそれ以外での差は比較的小さかった．エポック数に着目すると図4.17, 図4.18から検証データの損失は肥大化しているものの10から20に近いエポック数が必要であることが確認できる．

第5章 結論

本研究では動的グラフの分析のための Node Embedding であるプロクラステス回転を用いた既存手法の tNodeEmbed に着目し，より効果的にプロクラステス回転を適応するために回転基準ノードの選択を行うことを提案した．また，基準ノード選択のためノード間の最短経路を利用しノードの時間変化の定義を提案した．提案したノードの時間変化の定義を用いたノードの選択を取り入れた動的グラフに対する Node Embedding の手法である SPRDNE を提案し提案手法 SPRDNE を動的リンク予測を用いて評価し，ノードの選択の効果を確認した．今回，定義した時間変化のノード同士で比較した場合の差が大きいデータセットでパラメータ調節できる重みを用いた場合特に効果的なことが分かった．一方で時間変化が小さい場合，新たに別の時間変化を定義することで基準ノードの選択が有効になる可能性があり本研究の今後の課題となる．また，本研究では回転のみを用いたが今後スケーリングやシフトを用いることでより効果的に時間変化をノードのベクトル表現に反映することができると予想され同様に今後の課題である．

謝辞

本研究を進めるにあたり，お忙しい中私の研究の面倒を見ていただき，ゼミなどで適切かつ熱心な指導して下さった劉志准教授ならびに森田啓義名誉教授に心より深く御礼申し上げます．また，学部時代から異動後も積極的なご指導を下さった笠井裕之教授（現早稲田大学）に深く感謝申し上げます．そして，日頃よりゼミ等で切磋琢磨して活動してきた研究室の学生の皆様に感謝申し上げます．

参考文献

- [1] Norman Biggs, E Keith Lloyd, and Robin J Wilson. *Graph Theory, 1736-1936*. Oxford University Press, 1986.
- [2] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710. ACM, 2014.
- [3] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.
- [4] Uriel Singer, Ido Guy, and Kira Radinsky. Node embedding over temporal graphs. *arXiv preprint arXiv:1903.08889*, 2019.
- [5] John R Hurley and Raymond B Cattell. The procrustes program: Producing direct rotation to test a hypothesized factor structure. *Behavioral science*, Vol. 7, No. 2, pp. 258–262, 1962.
- [6] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, pp. 1–25, 2018.
- [7] Xiao Huang, Jundong Li, and Xia Hu. Label informed attributed network embedding. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 731–739. ACM, 2017.
- [8] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 817–826. ACM, 2009.
- [9] Shaosheng Cao, Wei Lu, and Qionгкаi Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM*

- international on conference on information and knowledge management*, pp. 891–900. ACM, 2015.
- [10] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 459–467. ACM, 2018.
- [11] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pp. 1067–1077. International World Wide Web Conferences Steering Committee, 2015.
- [12] Shaosheng Cao, Wei Lu, and Qionghai Xu. Deep neural networks for learning graph representations. In *Thirtieth AAAI Conference on Artificial Intelligence*, pp. 1145–1152, 2016.
- [13] Rui Feng, Yang Yang, Wenjie Hu, Fei Wu, and Yueting Zhang. Representation learning for scale-free networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 282–289, 2018.
- [14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.
- [17] Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. Dynamic network embedding by modeling triadic closure process. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [18] Piotr Bielak, Kamil Tagowski, Maciej Falkiewicz, Tomasz Kajdanowicz, and Nitesh V Chawla. Fildne: A framework for incremental learning of dynamic networks embeddings. *Knowledge-Based Systems*, p. 107453, 2021.

- [19] Medium of tnodeembedding. <https://medium.com/@urielsinger/tnodeembedding-over-temporal-graphs-b7bcbf59938f>.
- [20] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, Vol. 31, No. 1, pp. 1–10, 1966.
- [21] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, Vol. 20, No. 2, pp. 215–232, 1958.
- [22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.
- [23] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, Vol. 12, No. 10, pp. 2451–2471, 2000.
- [24] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, Vol. 28, No. 10, pp. 2222–2232, 2016.
- [25] Tore Opsahl Pietro Panzarasa and Kathleen M. Carley. Patterns and dynamics of users’ behavior and interaction: Network analysis of an online community. 2009.
- [26] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection, 2014.
- [27] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, Vol. 10, No. 4, pp. 255–268, 2006.
- [28] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.
- [29] Alain G’enois, Mathieuand Barrat. Can co-location be used as a proxy for face-to-face contacts? *EPJ Data Science*, Vol. 7, No. 1, p. 11, May 2018.

研究業績

国際会議

- Riku Hashimoto, Hiroyuki Kasai: “Sequential semi-orthogonal multi-level NMF with negative residual reduction for network embedding” 45th International Conference on Acoustics, Speech, and Signal Processing (IEEE), 2020.5.4-8 (Barcelona, Spain)

国内発表

- 橋本陸・笠井裕之: 「Network Embeddingのための負値残差低減および半直交制約付き非負行列分解」第107回オーディオビジュアル複合情報処理研究発表会 (情報処理学会), 2019.12.6 (盛岡)