

ノードがテキスト情報を持つ動的ネットワークにおける ノードと単語の分散表現学習

伊藤 寛祥[†] 駒水 孝裕^{††} 天笠 俊之^{††} 北川 博之^{††}

[†] 筑波大学大学院システム情報工学研究科 〒305-8577 茨城県つくば市天王台 1-1-1

^{††} 筑波大学計算科学研究センター 〒305-8577 茨城県つくば市天王台 1-1-1

E-mail: [†]hiro.3188@kde.cs.tsukuba.ac.jp, ^{††}taka-coma@acm.org ^{††}{amagasa,kitagawa}@cs.tsukuba.ac.jp

あらまし ネットワークにおけるノードの分散表現学習手法として network embedding が存在する。ノードの分散表現を得ることで、ノードに対してベクトル空間ベースの機械学習手法を適用できるようになり、ノードの分類、クラスタリング、可視化などの多様なアプリケーションに応用できる。本論文ではノードがテキスト情報を持つ動的ネットワークにおけるノードと単語の分散表現学習手法を提案する。本手法を入力となるネットワークデータに適用すると、ノードおよび単語の分散表現が得られ、さらに、分散表現の時系列的な変化が得られる。学術論文データベースにおける共著ネットワークを用いた実験より以下を示した。(1) 本手法で得られたノードの特徴に対して機械学習手法を適用すると、既存の手法と比較して高精度に分類、クラスタリングできる。(2) ノードと単語の分散表現学習において時系列性を考慮することで、考慮しない場合と比較して高い精度を実現する。(3) 本手法により学習されるノードと単語の分散表現の類似度は、入力となるネットワークにおけるノードの隣接性と単語の共起性の強さをそれぞれ反映しており、さらに、それらの時系列的な変化を捉えられる。

キーワード 分散表現学習, グラフデータマイニング, 時系列データ

1. はじめに

ノードがテキスト情報を持つネットワークは実世界に多く存在する。このようなネットワークの例として、SNS におけるソーシャルネットワークや、学術研究における共著ネットワークなどが挙げられる。共著ネットワークを例にすると、研究者がノード、共著関係がエッジとなり、各ノード（研究者）は各々がこれまでに発表した論文のタイトルまたはアブストラクトをテキスト情報として保持する。このようなネットワークにおける、グラフ構造とテキスト情報を利用したデータマイニング手法は数多く研究されている [4], [22], [24]。また、このような研究のなかで、エッジ情報とテキスト情報には相関関係があることが明らかにされてきた (homophily effect [6], [11])。この性質を捉えて、エッジ情報とテキスト情報を同時に用いることで、コミュニティ検出 [24]、感情分析 [4]、ノード分類 [22] といった問題に対して、エッジ情報やテキスト情報のみを考慮する場合と比較してより高精度な分析が行えることが報告されている。

ネットワークデータに対するデータマイニング手法のひとつに network embedding [3], [14], [18] という手法が存在する。network embedding とは、ネットワークにおけるノード間の隣接関係に基づいてノードをベクトル空間に埋め込み、ノードの分散表現を得る手法である。この手法をネットワークデータに対して適用することで、ノード集合に対してベクトル空間ベースの様々な機械学習手法が適用できるようになる。例として、ノードの分類 [25]、クラスタリング [24]、リンク予測 [2]、ネットワークの可視化 [16] に対して有用であることが知られている。

Network embedding に関するこれまでの多くの研究は、ネッ

トワークの構造のみに着目した手法がほとんどだった。近年、ノードが持つ属性を利用し、network embedding の性能を向上させる取り組みが行われてきた [5], [10], [17]。これらの手法は、ネットワーク構造と属性値の間の相関関係を利用し、それらを補い合わせた上でノードの分散表現を学習する手法で、ネットワーク構造のみを用いる手法よりも、機械学習のタスクにおいてより有用な分散表現が得られることが示されてきた。しかしながら、これまでの研究は、ノードをベクトル空間に埋め込むことに注目しており、単語の分散表現は得ることができない。すなわち、既存の手法では、単語とノードの相互関係を分散表現の類似度として得ることができず、ノードの分散表現の意味付けが困難であるという問題がある。

さらに、ノードがテキスト情報をもつネットワークは、時間の経過とともに変化する場合がある。すなわち、時間の経過とともに、ネットワークにおけるノードやエッジが増減したり、ノードが保持するテキスト情報が変化するという状況が考えられる。研究者ネットワークを例にすると、時間の経過により新しい研究者が登場し、また既存の研究者も新しい共著関係を構成する可能性がある。また、研究者の研究分野も変化する場合があるため、関連するテキスト情報も時間とともに変化する場合がある。これらのことから、ノードおよび単語の分散表現は時間の経過とともに変化すると考えられる。このような時系列変化するネットワークにおける分散表現学習のための最も単純な方策としては、各時刻ごとに既存の分散表現学習手法を適用することが考えられる。しかしながら、このアプローチでは、前の時刻における分散表現の学習結果を考慮することができないため、現時刻におけるデータにオーバーフィットする可能性

があるという問題が存在する。

本稿では、ノードがテキスト情報をもつ動的ネットワークにおけるノードと単語の分散表現学習手法を提案する。本手法では、ノードと単語の分散学習は相互に補い合いながら同時に学習され、さらにこれらの分散表現の時間的な変化が検出される。本研究における基本的な仮定は、以下の3つである。(i) ネットワーク中で隣接性が高いノード同士の分散表現は類似度が高い。(ii) あるノードとそれに関連する単語の分散表現の類似度は高い。(iii) 隣接する時間区分におけるノードと単語の分散表現は大きく変化しない。本研究では仮定 (i),(ii),(iii) を非負値行列分解 [8], [9] として定式化し、これらを同時に最適化するアルゴリズムを提案する。

実験では、学術論文データベースにおける共著ネットワークを用い、本手法の有用性に関して検証を行った。実験では以下の3つの項目に関して検討を行った。(Q1) 本手法により得られるノードの分散表現は、機械学習手法に対してどれだけ有用であるか?(Q2) ノードがテキスト情報をもつ動的ネットワークにおける分散表現学習において、前の時刻の分散表現を考慮することがどれだけ有益であるか?(Q3) ノードと単語の分散表現の類似度はノードの隣接性、単語の共起性の強さを反映しているか? またそれらの時間的な変化が検出できるか? 実験の結果、以下の3つの結論が得られた。(A1) 本手法により得られたノードの分散表現に対して、ベクトル空間ベースの分類手法 (one-vs-rest SVM)、クラスタリング手法 (k-means) を適用すると、既存の分散表現学習により得られた分散表現に対して適用するよりも高い精度を実現する。(A2) 提案手法において前の時刻の分散表現を考慮することで、分類とクラスタリングにおいてより有用な分散表現が得られる。(A3) 提案手法により得られた分散表現の類似度は、ノードの隣接関係と、単語と単語の共起性、ノードと単語の共起性の強さを反映しており、さらに、それらの時間的な変化を検出できる。

本研究の貢献は以下にまとめられる。

- ノードがテキスト情報をもつ動的ネットワークにおけるノードと単語の分散表現学習手法を提案した。本手法により以下が得られる。(a) ノードの分散表現, (b) 単語の分散表現, (c) 分散表現の時間的な変化。
- 本手法により得られたノードの分散表現は、既存の分散表現学習手法により得られた分散表現と比較して、ベクトル空間ベースの分類、クラスタリング手法に対して有用である。
- 本手法により得られる分散表現の類似度は、ノードの隣接性、単語の共起性を表現しており、それらの時系列的な変化が得られる。

2. 関連研究

Network embedding 手法は、ネットワーク分析の研究分野において、近年多くの注目を集めている。ネットワークのエッジ情報を保存する network embedding 手法として、LLE [15], Isomap [19], Laplacian eigenmap [1] が存在する。これらの手法は隣接行列の固有ベクトルを検出することにより達成される。より最近では、ノードのより高次の隣接性を保存する network

embedding 手法が提案されている。DeepWalk [14] では、ネットワークのランダムウォークにより得られるノードのシーケンスに対して skip-gram [12] を適用することでノードの分散表現を得る。LINE [18] では、1次の隣接性、2次の隣接性を保存する手法を提案した。node2vec [3] では、DeepWalk におけるノードのシーケンス作成の際、深さ優先探索と幅優先探索の双方を考慮することで、network embedding の性能向上に成功した。

単語の分散表現学習手法は、自然言語処理、情報検索、データマイニングにおいて近年多くの注目を集めている。Skip-gram と CBOW [12], [13] は word2vec としても知られ、最もよく知られる単語の分散表現学習手法である。この手法は、単語の意味的な類似度を表現するベクトル表現を効率的に学習する手法である。また、この手法は多くの拡張手法が存在しており、その例として、文書と単語の分散表現学習手法の ParagraphVector [7] が存在する。この手法では、PV-DM と PV-DBOW 2つのモードが存在する。

ネットワークのリンク構造に加え、ノードの属性情報を考慮した network embedding 手法が提案されている。ノードのテキスト情報を考慮した network embedding 手法として TADW [23] が存在する。また、この研究では、DeepWalk は行列分解モデルとして表現できることを示した。ノードの属性情報を考慮した network embedding 手法として、LANE [5] が存在する。この手法は、ネットワーク中でのノードの隣接性が高く、属性の類似度が高いノードは分散表現の類似度が高いという直感に基づいている。

本研究では、ノードの分散表現と単語の分散表現を同時に獲得するという新しいアプローチをとり、ネットワークの時系列的变化に対応した新しい分散表現学習手法を提案する。

3. 問題設定

本節では、本手法の入力となる、ノードがテキスト情報を持つ動的ネットワークの定義、および、本研究の目的である、ノードと単語の分散表現学習の定義に関して記述する。

時刻 t における、 n 個のノードがそれぞれ d 次元の単語特徴量を持つ、ノードがテキスト情報を持つ動的ネットワークを $\mathcal{G}^{(t)} = \{\mathbf{A}^{(t)}, \mathbf{X}^{(t)}\}$ とする。ここで、 $\mathbf{A}^{(t)} \in \mathbb{R}^{n \times n}$ はネットワークの重み付き隣接行列を表し、行列 $\mathbf{A}^{(t)}$ の各要素 $A_{u,v}^{(t)}$ はノード u と v のエッジの重みを表す。 $\mathbf{X}^{(t)} \in \mathbb{R}^{n \times d}$ は単語特徴量行列を表し、各要素 $X_{u,x}^{(t)}$ はノード u の単語 x に対する重みを表現する。ノードがテキスト情報を持つネットワークの時刻 $t = 0 \sim T$ におけるすべてのスナップショットを $\mathcal{G}^{(t)}$ の集合とし、 $\mathcal{G} = \{\mathcal{G}^{(t)}\}_{t=0}^T$ と表現する。本研究では [20] と同様の設定を用い、ノードの数は時間の経過で変化させず、エッジ数のみを変化させる (ある時刻 t で存在しないノードはエッジと単語特徴を持たない)。これにより、行列 $\mathbf{A}^{(t)}$ および $\mathbf{X}^{(t)}$ のサイズは時間の経過で変化しなくなる。

本研究の目的は、ノードと単語の k 次元の分散表現を獲得し、それらの時間的な変化を検出することである。ここで、 k は n と d と比較して十分に小さいとする ($k \ll \{n, d\}$)。本研

究の目的は以下に定義される。

$$\mathcal{L}_{word} = \left\| \mathbf{X}^{(t)} - \mathbf{U}^{(t)}(\mathbf{M}^{(t)})^T \right\|_F^2 \quad (2)$$

定義 3.1 (研究目的) ノードがテキスト情報を持つ動的ネットワーク G が与えられた時、本研究の目的は時刻 $t = 0 \sim T$ におけるすべてのノードの分散表現を表す非負行列 $\mathbf{U}^{(t)} \in \mathbb{R}^{n \times k}$ と、すべての単語の分散表現を表す非負行列 $\mathbf{M}^{(t)} \in \mathbb{R}^{d \times k}$ を得ることである。ここで、 $\mathbf{U}_i^{(t)} \in \mathbb{R}^k$ (行列 \mathbf{U} の i 行目) はノード i の分散表現を表し、 $\mathbf{M}_i^{(t)} \in \mathbb{R}^k$ (行列 \mathbf{M} の i 行目) は単語 i の分散表現を表す。□

4. 提案手法

本節では、本研究の提案手法である、ノードがテキスト情報をもつ動的ネットワークにおける分散表現学習手法に関して説明する。本手法では、ノードと単語の分散表現を k 次元の非負ベクトルで表現する。本手法では、ノードと単語の分散表現に関する以下の3つの損失関数を設定する。(1) ノードの隣接性の保存, (2) ノードと単語の関連性の保存, (3) 前時刻における分散表現の保存。これらを統合した損失関数を、複数の制約付き Non-negative Matrix Factorization (NMF) として定式化し、それらを同時に最適化することで、ノードと単語の分散表現を得る。

4.1 損失関数

4.1.1 ノードの隣接性の保存

本研究では、[21] に倣い、ノード間の隣接性を、1次の隣接性と2次の隣接性の線形和で表現する。ノード間の1次の隣接性を表す行列 $\mathbf{S}_1^{(t)} \in \mathbb{R}^{n \times n}$ を $\mathbf{S}_1^{(t)} = \mathbf{A}^{(t)}$ とする。ノード間の2次の隣接性を表す行列 $\mathbf{S}_2^{(t)} \in \mathbb{R}^{n \times n}$ を $\mathbf{S}_2^{(t)} = (\mathbf{A}^{(t)})^T \mathbf{A}^{(t)}$ とする。最後に、ノード間の隣接性行列を $\mathbf{S}^{(t)} = \mathbf{S}_1^{(t)} + \eta \mathbf{S}_2^{(t)}$ とする。ここで、 $\eta > 0$ は1次と2次の重みパラメータで、本研究では $\eta = 3$ で固定する。

Network embedding の基本的な方策は、ノード同士の分散表現の類似度とノード同士の隣接性との差異を最小化することである [14]。本研究では非負の重み行列 $\mathbf{V}^{(t)} \in \mathbb{R}^{n \times k}$ を設定し、分散表現行列 $\mathbf{U}^{(t)} \in \mathbb{R}^{n \times k}$ との行列積で隣接性行列 $\mathbf{S}^{(t)}$ を復元させることで、ノードの分散表現を得る。これは以下の損失関数で記述される。

$$\mathcal{L}_{node} = \left\| \mathbf{S}^{(t)} - \mathbf{U}^{(t)}(\mathbf{V}^{(t)})^T \right\|_F^2 \quad (1)$$

ここで、 $\|\cdot\|_F^2$ はフロベニウスノルムである。

4.1.2 ノードと単語の関連性の保存

直感的には、ノード i が単語 w と関連するとき、 i と w の分散表現の類似度は高くなると考えられる。ノードと単語の関連性は、ノードがもつテキスト情報における単語の出現頻度として表現される。本研究では分散表現の類似度を内積として定義すると、上記の直感に関する損失は $(X_{i,w}^{(t)} - \mathbf{U}_i^{(t)} \cdot \mathbf{M}_w^{(t)})^2$ と表現される。この損失をすべてのノードと単語の組み合わせに関して考慮すると、ノードと単語の関連性の保存に関する損失関数は以下で記述される。

4.1.3 前時刻からの分散表現の保存

時刻 t における分散表現を得る際、前時刻 $t-1$ における分散表現を考慮させることは、ある時刻 t におけるデータへのオーバーフィットを避けるために有効である。さらに、直感的には、時刻 t における分散表現は、前時刻 $t-1$ の分散表現から大きく変化しないと考えられる。この直感は時刻 t と $t-1$ における分散表現行列のフロベニウスノルムの大きさで測ることができる。

$$\mathcal{L}_{shift} = \left\| \mathbf{U}^{(t)} - \mathbf{U}^{(t-1)} \right\|_F^2 + \left\| \mathbf{V}^{(t)} - \mathbf{V}^{(t-1)} \right\|_F^2 + \left\| \mathbf{M}^{(t)} - \mathbf{M}^{(t-1)} \right\|_F^2 \quad (3)$$

時刻 t と $t-1$ における分散表現が類似している時、 \mathcal{L}_{shift} の値は小さくなる。このことから、 \mathcal{L}_{shift} の項を考慮することで、時刻 $t-1$ における分散表現を時刻 t における分散表現に対して影響を与えられるということを示している。

4.1.4 統合した損失関数

ノードがテキスト情報をもつ動的ネットワークにおけるノードと単語の分散表現を得るためには、上記の3つの損失関数が最小化されなければならない。以下は3つの損失関数を統合した損失関数である。

$$\mathcal{L} = \delta \mathcal{L}_{node} + (1 - \delta) \mathcal{L}_{word} + \alpha \mathcal{L}_{shift} \quad (4)$$

ここで、 $\delta \in [0, 1]$ は \mathcal{L}_{node} と \mathcal{L}_{word} のバランスを調整するパラメータ、 $\alpha \in [0, \infty)$ は前時刻における影響の大きさを調整するパラメータである。本論文では、 $\delta = 0.5$ で固定する。

4.2 最適化

提案手法における最適化問題は以下で記述される。

$$\left\{ \mathbf{U}^{(t)}, \mathbf{V}^{(t)}, \mathbf{M}^{(t)} \right\} = \arg \min_{\mathbf{U}^{(t)}, \mathbf{V}^{(t)}, \mathbf{M}^{(t)} \geq \mathbf{0}} \mathcal{L} \quad (5)$$

式4はすべての変数行列に関して同時に凸にならない。ここで、[8]の最適化方式にならない、式4の最適化アルゴリズムを導出する。

最適化問題5のKKT条件は以下である。

$$\mathbf{U}^{(t)} \geq \mathbf{0}, \mathbf{V}^{(t)} \geq \mathbf{0}, \mathbf{M}^{(t)} \geq \mathbf{0} \quad (6)$$

$$\nabla_{\mathbf{U}^{(t)}} \mathcal{L} \geq \mathbf{0}, \nabla_{\mathbf{V}^{(t)}} \mathcal{L} \geq \mathbf{0}, \nabla_{\mathbf{M}^{(t)}} \mathcal{L} \geq \mathbf{0} \quad (7)$$

$$\mathbf{U}^{(t)} \odot \nabla_{\mathbf{U}^{(t)}} \mathcal{L} = \mathbf{0}, \mathbf{V}^{(t)} \odot \nabla_{\mathbf{V}^{(t)}} \mathcal{L} = \mathbf{0}, \mathbf{M}^{(t)} \odot \nabla_{\mathbf{M}^{(t)}} \mathcal{L} = \mathbf{0} \quad (8)$$

ここで、 \odot はアダマール積である。つづいて各変数に関する勾配は以下で表される。

$$\begin{aligned} \nabla_{\mathbf{U}^{(t)}} \mathcal{L} = & -\delta \mathbf{S}^{(t)} \mathbf{M}^{(t)} - (1-\delta) \mathbf{X}^{(t)} \mathbf{M}^{(t)} - \alpha \mathbf{U}^{(t-1)} \quad (9) \\ & + \delta \mathbf{U}^{(t)} (\mathbf{V}^{(t)})^T \mathbf{V}^{(t)} + (1-\delta) \mathbf{U}^{(t)} (\mathbf{M}^{(t)})^T \mathbf{M}^{(t)} \\ & + \alpha \mathbf{U}^{(t)} \end{aligned}$$

$$\begin{aligned} \nabla_{\mathbf{V}^{(t)}} \mathcal{L} = & -\delta (\mathbf{S}^{(t)})^T \mathbf{U}^{(t)} - \alpha \mathbf{V}^{(t-1)} \quad (10) \\ & + \delta \mathbf{V}^{(t)} (\mathbf{U}^{(t)})^T \mathbf{U}^{(t)} + \alpha \mathbf{V}^{(t)} \end{aligned}$$

$$\begin{aligned} \nabla_{\mathbf{M}^{(t)}} \mathcal{L} = & -(1-\delta) (\mathbf{X}^{(t)})^T \mathbf{U}^{(t)} - \alpha \mathbf{M}^{(t-1)} \quad (11) \\ & + (1-\delta) \mathbf{M}^{(t)} (\mathbf{U}^{(t)})^T \mathbf{U}^{(t)} + \alpha \mathbf{M}^{(t)} \end{aligned}$$

式 8 に対して勾配を代入することで、各変数に関して更新式を導出した。

$$\mathbf{U}^{(t)} \leftarrow \mathbf{U}^{(t)} \odot \quad (12)$$

$$\frac{\delta \mathbf{S}^{(t)} \mathbf{M}^{(t)} + (1-\delta) \mathbf{X}^{(t)} \mathbf{M}^{(t)} + \alpha \mathbf{U}^{(t-1)}}{\mathbf{U}^{(t)} (\delta (\mathbf{V}^{(t)})^T \mathbf{V}^{(t)} + (1-\delta) (\mathbf{M}^{(t)})^T \mathbf{M}^{(t)}) + \alpha \mathbf{U}^{(t)}}$$

$$\mathbf{V}^{(t)} \leftarrow \mathbf{V}^{(t)} \odot \frac{\delta (\mathbf{S}^{(t)})^T \mathbf{U}^{(t)} + \alpha \mathbf{V}^{(t-1)}}{\delta \mathbf{V}^{(t)} (\mathbf{U}^{(t)})^T \mathbf{U}^{(t)} + \alpha \mathbf{V}^{(t)}} \quad (13)$$

$$\mathbf{M}^{(t)} \leftarrow \mathbf{M}^{(t)} \odot \frac{(1-\delta) (\mathbf{X}^{(t)})^T \mathbf{U}^{(t)} + \alpha \mathbf{M}^{(t-1)}}{(1-\delta) \mathbf{M}^{(t)} (\mathbf{U}^{(t)})^T \mathbf{U}^{(t)} + \alpha \mathbf{M}^{(t)}} \quad (14)$$

上記更新式を繰り返し実行することで、損失関数 4 は単調減少し、ある程度収束したところで行列分解が完了する。

Algorithm 1 は最適化問題 5 を最適化するためのアルゴリズムである。行列の正規化は行列の更新の後に行われる。 $\mathbf{Q}_U^{(t)}, \mathbf{Q}_V^{(t)}, \mathbf{Q}_M^{(t)}$ は正規化行列で、 $\mathbf{Q}_U^{(t)} = \text{Diagonalize}(\mathbf{U}^{(t)})$, $\mathbf{Q}_V^{(t)} = \text{Diagonalize}(\mathbf{V}^{(t)})$, $\mathbf{Q}_M^{(t)} = \text{Diagonalize}(\mathbf{M}^{(t)})$ である。ここで、

$$\text{Diagonalize}(Z \in \mathbb{R}^{a \times b}) = \text{Diag} \left(\sum_{i=1}^b Z_{1,i}, \dots, \sum_{i=1}^b Z_{a,i} \right) \quad (15)$$

である。 $\text{Diag}(\cdot)$ は対角行列の成分のリストを入力とし、対角行列を返す関数である。

5. 実験

本節では、実データを用いた実験を通して提案手法の有用性に関して議論する。本実験では以下の 3 つの項目に関して議論を行う。(Q1) 本手法により得られるノードの分散表現は、機械学習手法に対してどれだけ有用であるか？(Q2) ノードがテキスト情報をもつ動的ネットワークにおける分散表現学習において、前の時刻の分散表現を考慮することがどれだけ有益であるか？(Q3) ノードと単語の分散表現の類似度はノードの隣接性、単語の共起性の強さを表現しているか？またそれらの時間的な変化が検出できるか？

5.1 データセット

本研究では、コンピュータサイエンス分野の学術論文データベースの DBLP (Digital Bibliography Project)^(注1) を用いて本手法の有用性の検証を行う。DBLP は各著作のタイトルや

Algorithm 1 Optimization Algorithm

Input: $\{\mathbf{A}^{(t)}, \mathbf{X}^{(t)}\}_{t=0}^T, \delta, \alpha$

Output: $\{\mathbf{U}^{(t)}, \mathbf{V}^{(t)}, \mathbf{M}^{(t)}\}_{t=0}^T$

```

1:  $\epsilon' \leftarrow \max Int, \epsilon \leftarrow \frac{\epsilon'}{2}$ 
2: for  $t \in 0 : T$  do
3:    $\mathbf{U}^{(t)}, \mathbf{V}^{(t)}, \mathbf{M}^{(t)} \leftarrow$  random non-negative init
4:   while  $abs(\epsilon' - \epsilon) \geq \delta$  do
5:      $\mathbf{U}^{(t)} \leftarrow \mathbf{U}^{(t)} \odot$ 
6:        $\frac{\delta \mathbf{S}^{(t)} \mathbf{M}^{(t)} + (1-\delta) \mathbf{X}^{(t)} \mathbf{M}^{(t)} + \alpha \mathbf{U}^{(t-1)}}{\mathbf{U}^{(t)} (\delta (\mathbf{V}^{(t)})^T \mathbf{V}^{(t)} + (1-\delta) (\mathbf{M}^{(t)})^T \mathbf{M}^{(t)}) + \alpha \mathbf{U}^{(t)}}$ 
7:      $\mathbf{V}^{(t)} \leftarrow \mathbf{V}^{(t)} \odot$ 
8:        $\frac{\delta (\mathbf{S}^{(t)})^T \mathbf{U}^{(t)} + \alpha \mathbf{V}^{(t-1)}}{\delta \mathbf{V}^{(t)} (\mathbf{U}^{(t)})^T \mathbf{U}^{(t)} + \alpha \mathbf{V}^{(t)}}$ 
9:      $\mathbf{M}^{(t)} \leftarrow \mathbf{M}^{(t)} \odot$ 
10:       $\frac{(1-\delta) (\mathbf{X}^{(t)})^T \mathbf{U}^{(t)} + \alpha \mathbf{M}^{(t-1)}}{(1-\delta) \mathbf{M}^{(t)} (\mathbf{U}^{(t)})^T \mathbf{U}^{(t)} + \alpha \mathbf{M}^{(t)}}$ 
11:      $\mathbf{U}^{(t)} \leftarrow \mathbf{U}^{(t)} \mathbf{Q}_U^{(t)}$ 
12:      $\mathbf{V}^{(t)} \leftarrow \mathbf{V}^{(t)} \mathbf{Q}_V^{(t)}$ 
13:      $\mathbf{M}^{(t)} \leftarrow \mathbf{M}^{(t)} \mathbf{Q}_M^{(t)}$ 
14:      $\epsilon' \leftarrow \epsilon$ 
15:      $\epsilon \leftarrow \mathcal{L}(\mathbf{U}^{(t)}, \mathbf{V}^{(t)}, \mathbf{M}^{(t)})$ 
16:   end while
17: end for

```

著者、会議といった情報を保持するデータベースである。本研究では使用する論文データを、データベース (DB)、データマイニング (DM)、機械学習 (ML)、情報検索 (IR) の 4 つの分野に絞り、それぞれの分野に関して 5 つずつ会議を選択して、2016 までにそれらの会議で発表された論文を対象にして実験を行った。実験で使用した会議の内訳を表 1 に示す。ノイズの削減のため、5 本以上論文を出していない著者はデータから除いた。テキスト情報に関しては、前処理としてストップワードの除去とステミングを行った。各著者は各年において最も多く著作を投稿した分野にラベル付けされる。

本実験では、ノードがテキスト情報を持つ動的ネットワークは以下の方式で構成される。ノードは著者 (研究者) である。各著者は共著関係に基いてエッジが張られており、エッジの重みは共著論文の数である。ネットワークのスナップショットは各年ごとに構成される。ノードのテキスト情報は、各著者が各年に発表した論文のタイトルをすべて連結したものを bag-of-words ベクトルに変換したものとす。著者 (ノード) の数は 6625 で、特徴となる単語数は 2845 である。本実験では簡単のためエッジの消去は考慮しない。図 1 はグラフにおける各年でのエッジ数を示しており、時間の経過とともにエッジ数が増えていることがわかる。

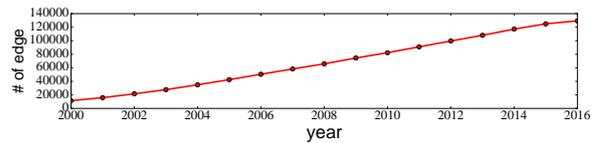


図 1: ネットワークの各年におけるエッジ数

5.2 定量的評価

本節では、(Q1) 本手法で得られるノードの分散表現は、機械学習手法に関してどれだけ有用であるか？ に関して検証を

(注1): <http://dblp.uni-trier.de/>

表 1: 各研究分野において選出した国際学会のリスト

DB	DM	ML	IR
SIGMOD, VLDB	KDD, ICDM	NIPS, ICML	SIGIR, ECIR
PODS, EDBT	PKDD, SDM	ECML, UAI	JCDL, ECDL
ICDT	PAKDD	COLT	TREC

行う。ここでは、本手法と比較手法それぞれによって得られたノードの分散表現に対して、線形代数ベースの機械学習手法を適用することで、分類、クラスタリングを行い、本手法によって得られるノードの分散表現の有用性に関して議論する。分類に関しては one-vs-rest SVM を用いてノードのラベルの推定を行う。分類精度は 10 分割交差検定で検証を行い、学習データに 90%、テストデータに 10% の割合でデータを分割する。クラスタリングには k-means を使い、精度は 10 回実験を行ったときの精度の平均で測る。

比較手法には以下の 2 つの手法を用いた。

- **DeepWalk** [14]: ノードの分散表現学習手法のベースラインで、ネットワークのエッジ構造のみから分散表現を学習する手法である。この手法ではノードのシーケンスを、複数のランダムウォークから生成し、そのシーケンスに対して skip-gram を適用することで、ノードの分散表現を得る。

- **PV-DBOW** [7]: 単語の分散表現学習手法のベースラインで、文書の分散表現と単語の分散表現を同時に得る手法である。

これらの手法は、ネットワークの動的な変化に対応できないため、各時刻のスナップショットに対してこれらの手法を適用することで、それぞれの分散表現を得る。本実験では、すべての手法における埋め込み空間の次元を 100 に固定して実験を行う。

図 2(a) は各時刻における、既存手法と提案手法の分類精度を示したものである。この図から、本手法は既存の手法と比較してより高い精度で分類を行えることが分かる。図 2(b) は各時刻における、既存手法と提案手法のクラスタリングの精度を示したものである。この図から、本手法は既存の手法と比較してより高い精度でクラスタリングを行えることが分かる。

5.3 パラメータ α の影響

本節では、(Q2) ノードがテキスト情報をもつ動的ネットワークにおける分散表現学習において、前の時刻の分散表現を考慮することがどれだけ有益であるか? に関して議論する。本実験では、提案手法におけるパラメータ α を変化させることにより、分類、クラスタリング精度がどのように変化するかを検証する。パラメータ α は分散表現の学習において、前時間における分散表現をどれだけ考慮するかを調節するパラメータである。 $\alpha = 0$ のときは前の時刻における分散表現を全く考慮しなくなり、 $\alpha = \infty$ のときは学習される分散表現は前の時刻における分散表現と全く同じになる。このため、本実験では最も適切な α を探索することで、前の時刻における分散表現をどれだけ考慮することが最も有効なのかに関して調査する。

図 3(a) は各年におけるノードの分類精度を示したものであ

る。この図を見ると、 $\alpha = 0.01, 0.1$ のときに、全く考慮しない場合と比較してより高い精度で分類、クラスタリングできていることがわかる。図 3(b) は各年におけるノードのクラスタリング精度を示したものである。これを見ると $\alpha = 0.1, 1$ のときに特に高い精度を示していることが分かる。これらの結果から、前の時刻における分散表現をある程度考慮することで、分類とクラスタリングにおいてより有用な分散表現が得られることが示された。

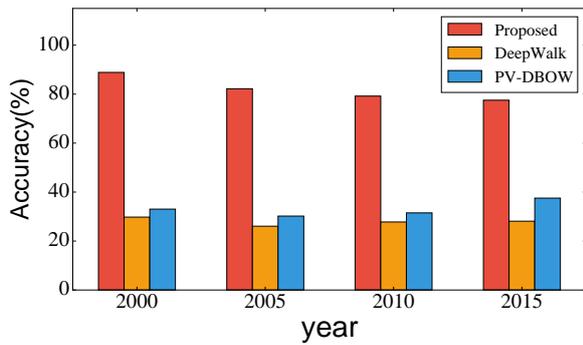
5.4 定性的評価

本節では、(Q3) ノードと単語の分散表現の類似度はノードの隣接性、単語の共起性の強さを表現しているか? またそれらの時間的な変化が検出できるか? に関して議論する。本実験では、ノードとノード、単語と単語、ノードと単語の分散表現の類似度を測り、それが大きいものがどのようなノードまたは単語であるかを観察することで、学習された分散表現の特性を調査する。本実験では、分散表現の類似度は内積の大きさとする。

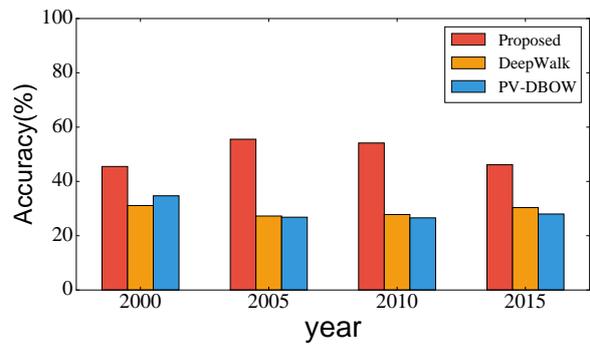
図 4 は 2000 年、2005 年、2010 年、2015 年において、ノード *jiawei han* の分散表現と最も類似する単語の分散表現の top-10 を並べたものである。この図を見ると、2000 年において最も類似度が高い単語は “knowledge discovery” や “database”，2005 年においては “efficient itemset” や “graph mining”，2010 年では “network mining”，2015 年では “heterogeneous network mining” と時間の経過とともに類似度が高い単語が変化していることが分かる。これは、時間の経過による “jiawei han” の研究テーマの変化を検出できたということを示している。

図 5 は 2000 年、2005 年、2010 年、2015 年において、ノード *jiawei han* の分散表現と最も類似するノードの分散表現の top-10 を並べたものである。この図をみると、*jiawei han* はすべての時刻において *raymond t. ng, philip s. yu, jian pei* と最も類似度が高く、また、時間の経過とともに、類似度が高いノードの内訳が変化していることがわかる。これは、研究活動における定常的な共著関係（教授職同士の共著）と一時的な共著関係（教授と学生の共著）のそれぞれが検出できていることを示している。

図 6 は 2000 年、2005 年、2010 年、2015 年において単語 *neural* の分散表現と最も類似度が高い単語の分散表現の top-10 を示したものである。この図を見ると、2000 年には “recognition” と類似度が高く、2005 年には “kernel” や “bayesian”，2010 年には “convolution”，2015 年には “deep neural networks” や “recurrent neural network” といった単語との類似度が高いということが分かる。これらのことは、ある単語と関連する単語が時間の経過とともに変化していることを示しており、研究トピックに関連する単語の時間経過による変化を検出できていることを示している。しかしながら、2010 年、2015 年においては、類似度が高い単語に “social” が含まれている。この結果は “network” という単語が、“neural network” と “social network” の二通りの使われ方が存在することに起因していると考えられる。このような単語の多義性に対応していくことは今後の課題として挙げられる。

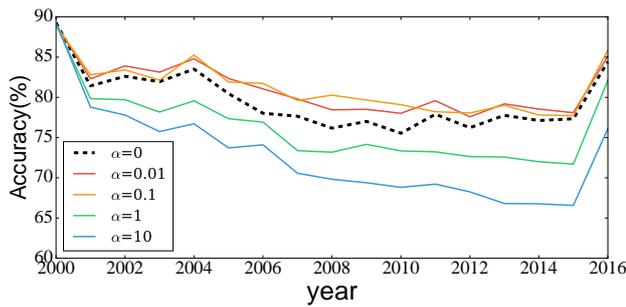


(a) 分類精度

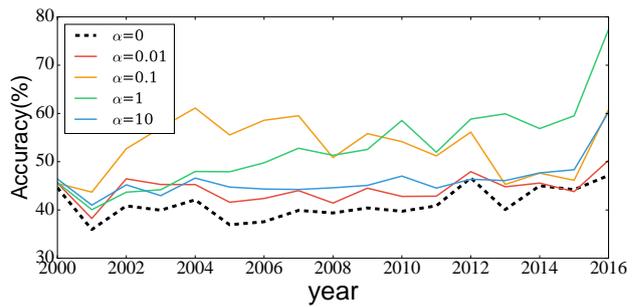


(b) クラスタリング精度

図 2: 既存手法と提案手法の各年における分類精度とクラスタリング精度



(a) 分類精度



(b) クラスタリング精度

図 3: 提案手法のパラメータ α を変化させた際の各年における分類精度とクラスタリング精度

6. まとめ

本稿では、ノードがテキスト情報を持つ動的なネットワークにおけるノードと単語の分散表現学習手法を提案した。実験の結果より、以下の3つを示した。(1) 本手法より学習されたノードの分散表現は、既存手法による分散表現と比較して分類とクラスタリングに関して有用である。(2) 時間的な変化を考慮することで、分類とクラスタリングに関してより有効な分散表現が得られる。(3) 本手法により学習される分散表現同士の類似度は、ノードの隣接性と、単語の共起性、ノードと単語の関連性をそれぞれ反映しており、それらの時間的な変化を捉えられている。

今後の展望としては以下の二つが挙げられる。(i) 単語の多義性の問題の解消。実験において議論したとおり、データ工学の分野では“network”という単語は“social network”と“neural network”の2通りの使用方法がある。このような単語の多義性は、ノードおよび単語の埋め込みにおいて、“social”と“neural”の類似度が不当に高くなるといった、誤った学習をもたらす可能性がある。今後はこのような単語の多義性をもたらす問題を解決する方策に関して検討していく予定である。(ii) 分散表現の予測。まだ観測されていないネットワークの状態を予測し、分散表現を得ることは様々なアプリケーションにおいて極めて有用である。このような問題設定に対応し、現時刻までに得られた分散表現から次の時刻の分散表現を予測する手法を検討していく予定である。

謝 辞

本研究はAMEDの課題番号JP161k1010017h0001の支援を受けたものである。

文 献

- [1] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pp. 585–591, 2002.
- [2] Sheng Gao, Ludovic Denoyer, and Patrick Gallinari. Temporal link prediction by integrating content and structure information. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 1169–1174, 2011.
- [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] Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 537–546, 2013.
- [5] 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, 2017.
- [6] Denise B Kandel. Homophily, selection, and socialization in adolescent friendships. *American journal of Sociology*, Vol. 84, No. 2, pp. 427–436, 1978.
- [7] Quoc Le and Tomas Mikolov. Distributed representations

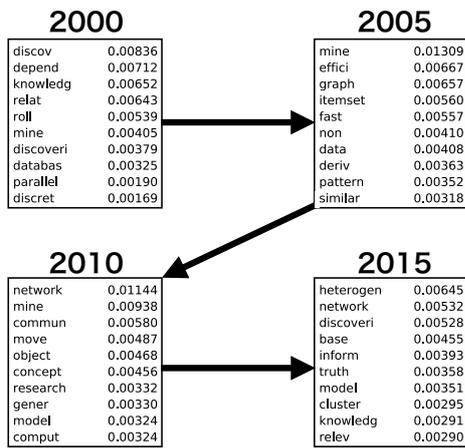


図 4: “jiawei han” の分散表現と最も類似度が高い単語の分散表現

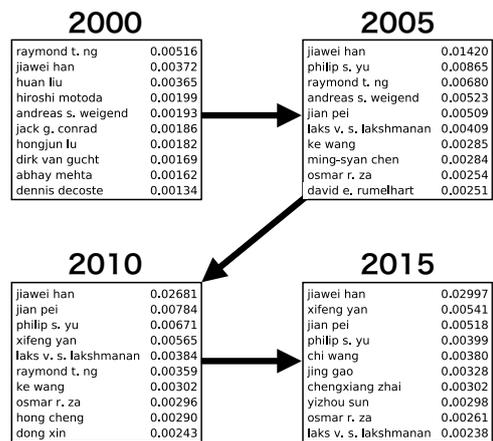


図 5: “jiawei han” の分散表現と最も類似度が高いノードの分散表現

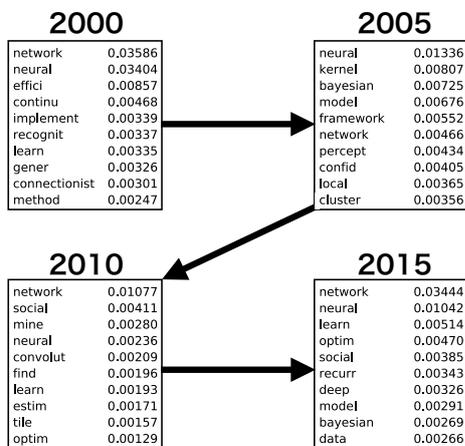


図 6: “neural” の分散表現と最も類似度が高い単語の分散表現

of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1188–1196, 2014.

- [8] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, Vol. 401, No. 6755, pp. 788–791, 1999.
- [9] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pp. 556–562, 2001.
- [10] Jundong Li, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, and Huan Liu. Attributed network embedding for learning in a dynamic environment. *arXiv:1706.01860*, 2017.
- [11] Peter V Marsden. Homogeneity in confiding relations. *Social networks*, Vol. 10, No. 1, pp. 57–76, 1988.
- [12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013.
- [13] 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.
- [14] 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, 2014.

- [15] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, Vol. 290, No. 5500, pp. 2323–2326, 2000.
- [16] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, pp. 287–297, 2016.
- [17] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1165–1174, 2015.
- [18] 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, 2015.
- [19] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, Vol. 290, No. 5500, pp. 2319–2323, 2000.
- [20] Hanghang Tong, Spiros Papadimitriou, Jimeng Sun, Philip S Yu, and Christos Faloutsos. Colibri: fast mining of large static and dynamic graphs. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 686–694, 2008.
- [21] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *AAAI*, pp. 203–209, 2017.
- [22] Yuto Yamaguchi, Christos Faloutsos, and Hiroyuki Kitagawa. Omni-prop: Seamless node classification on arbitrary label correlation. In *AAAI*, pp. 3122–3128, 2015.
- [23] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *IJCAI*, pp. 2111–2117, 2015.
- [24] Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining*, pp. 1151–1156, 2013.
- [25] Shenghuo Zhu, Kai Yu, Yun Chi, and Yihong Gong. Combining content and link for classification using matrix factorization. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 487–494, 2007.