

# 属性付きグラフのための非線形関数を用いた 接合加重非負値行列分解

前川 政司<sup>†</sup> 竹内 孝<sup>††</sup> 佐々木勇和<sup>†</sup> 鬼塚 真<sup>†</sup>

<sup>†</sup> 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

<sup>††</sup> NTT コミュニケーション科学基礎研究所 〒619-0237 京都府相楽郡精華町光台 2-4

E-mail: †{maekawa.seiji,sasaki,onizuka}@ist.osaka-u.ac.jp, ††takeuchi.koh@lab.ntt.co.jp

あらまし ソーシャルグラフの発展に伴いノード間のリンク構造とノードの属性情報の両方を含むグラフデータが出現しており、構造と属性の2つの情報を同時に利用する属性付きグラフクラスタリングに注目が集まっている。本研究では、構造と属性の非線形な関係性を利用したグラフクラスタリング手法である NJNMF を提案する。提案手法は、構造と属性を2つの行列として扱い、これらを非線形関数を用いてモデル化し同時分解を行うことで、従来の線形関数のみを用いたモデルよりも複雑な関係性を利用可能とする。クラスタリング結果の評価として3つの実データについて実験を行い、ノードと属性の間に複雑な関係があるとき NJNMF が既存の属性付きグラフクラスタリング手法を上回ることを示した。

キーワード グラフクラスタリング, 属性, 行列分解

## 1. はじめに

近年、ソーシャルネットワークや Web2.0 の発展に伴って、構造とノードの属性の両方を持ったグラフデータが多く出現している。例えば、ウェブページをノードと考えると、ハイパーリンクによってノード間の構造関係を与えることができ、ウェブページのテキストはノードに付与された属性情報として扱うことができる。これらの情報からクラスタの抽出が実現されれば、ウェブページのコミュニティの自動発見が可能になる。現在、広く用いられている手法であるグラフクラスタリング [5-7] では、グラフ構造で密に接続するノードを同一のクラスタに、疎に結合するノードを別々のクラスタに分類することによってグラフからコミュニティを発見する。グラフクラスタリングはウェブデータの解析のみならず、タンパク質の解析 [6] やソーシャルネットワークのコミュニティ推定 [7] などのデータ解析に応用されており、データマイニングと機械学習の分野で注目を集めている。しかし、既存のグラフクラスタリングでは構造の情報のみを用いており、属性の情報を考慮することができない。構造に加え属性情報を考慮することが出来れば、構造のみでは見つけられない質の高い、すなわちノード同士が密に接続する上に似た属性を持つクラスタが発見できる可能性があるため、構造と属性の両方を利用する属性付きクラスタリングは重要な技術となってくると言える。

属性付きグラフクラスタリングでは構造に加え属性を同時に考慮するために主に以下の2つの問題について取り組まれている。1つ目は構造と属性の関係性の効果的なモデル化である。多くの既存手法はクラスタを発見するために線形関数しか用いていないため、複雑な関係を考慮することができない。2つ目の問題は、膨大な属性情報のすべてがグラフクラスタリングに効果的とは限らないことである。そのため与えられた属

性情報からクラスタリングに有効な属性を自動で決定し利用する技術が重要である。Joint Weighted Nonnegative Matrix Factorization (JWNMF) [4] は、これらの問題を解決するために、構造と属性から得られる2つの行列に、非負値行列分解法 (Non-negative Matrix Factorization: NMF) を用いる手法である。JWNMF は構造の評価指標であるモジュラリティ [16] と属性の評価指標である平均エントロピー<sup>(注1)</sup> において従来のグラフクラスタリング手法や属性付きグラフクラスタリング手法よりも、良いクラスタリング結果を得た。しかし、JWNMF では構造と属性の間に線形の関係しか仮定しておらず、より複雑な非線形な関係が存在している場合にそれをクラスタリングに利用することができない。そこで本稿では、JWNMF の非線形拡張である、Non-Linear Joint Weighted Nonnegative Matrix Factorization (NJNMF) を提案する。提案手法は、属性情報について非線形問題を解くために中間層を導入するとともに活性化関数を利用することで、JWNMF では発見不可能な複雑でより詳細な分析を可能にする。モデルの複雑化によって生じうる過学習の問題に対応するために深層学習の最適化技術であるドロップアウトを採用する。このモデルは反復更新によって自動で最適化を行うことができる。実験は3つの実データについて行った。最先端の既存手法に比べ、高いクラスタリング結果を得られることを示した。

本稿の構成は以下の通りである。2章で属性付きグラフクラスタリングの既存手法について述べ、3章では提案手法である NJNMF について述べる。4章で実験結果を示し、それについて考察し、5章で関連する既存技術について述べる。6章で結論を述べる。

(注1) : [https://en.wikipedia.org/wiki/Entropy\\_\(information\\_theory\)](https://en.wikipedia.org/wiki/Entropy_(information_theory))

## 2. 事前準備

属性付きグラフは  $G = (V, E, A)$  として定義され、ここでは  $V = \{v_1, v_2, \dots, v_n\}$  はノード集合、 $E = \{(v_i, v_j), 1 \leq i, j \leq n\}$  はエッジ集合、 $A = [A_1, A_2, \dots, A_m]$  はノードの属性集合を表す。ここでは、集合はレギュラーの大文字、行列はボールド体の大文字、スカラーは小文字、行列  $X$  の  $i, j$  要素は  $X_{ij}$  とする。属性付きグラフ  $G$  では  $V$  中の各ノード  $v_i$  が属性ベクトル  $(a_1^i, a_2^i, \dots, a_m^i)$  と関連付けられている。ここでのベクトルの各要素は各属性での  $v_i$  の属性の値である。

属性付きグラフクラスタリングと従来のグラフクラスタリングとの違いはノードの属性を考慮に入れる必要があるかどうかである。理想的なクラスタリング結果として以下の2つのような特性が得られるべきである。(1) 同じクラスに属するノードは密に接続し、異なるクラスに属するノードは疎な接続を持つ。(2) 同じクラスに属するノードは似た属性の値を持ち、異なるクラスに属するノードは多様な属性の値を持つ。

### 2.1 SymNMF

まず Kuang らによって提案されたグラフクラスタリング法である SymNMF [2,3] について述べる。SymNMF はグラフクラスタリングのための NMF に特殊な制約を科したものである。通常の NMF はある非負の観測行列  $\mathbf{X}$  を近似する低ランクの因子行列  $\mathbf{C}$  と  $\mathbf{G}$  を推定する技術であり、次の最適化問題として定式化される。

$$\min_{\mathbf{C}, \mathbf{G} \geq 0} \|\mathbf{X} - \mathbf{C}\mathbf{G}^T\|_F^2 \quad (1)$$

ここでの  $\mathbf{X} \in \mathbb{R}_+^{m \times n}$ ,  $\mathbf{C} \in \mathbb{R}^{m \times k}$ ,  $\mathbf{G} \in \mathbb{R}_+^{n \times k}$ ,  $\mathbb{R}_+$  は非負の実数集合、 $\|\cdot\|$  はフロベニウスノルムを表す。次に SymNMF の目的関数を式 2 に示す。

$$\min_{\mathbf{H} \geq 0} \|\mathbf{S} - \mathbf{H}\mathbf{H}^T\|_F^2 \quad (2)$$

ここでの  $\mathbf{S} \in \mathbb{R}_+^{n \times n}$  はノード同士の類似度の情報を持った行列、 $\mathbf{H} \in \mathbb{R}_+^{n \times k}$  であり、分解後の行列が 1 つであることから  $\mathbf{H}$  を用いてクラスタリング結果を得るので解釈が自然であり、容易である。

SymNMF は  $\mathbf{S}$  を低ランク行列  $\mathbf{H}$  に圧縮する手法として考えることができ、これは主成分分析の特殊例と言える。また主成分分析は深層学習のオートエンコーダの 1 層で線形演算のみで構築されたものとみなすことができる。

### 2.2 JWNMF

本節で、Huang らに提案された最先端の属性付きグラフクラスタリング手法である JWNMF [4] について述べる。ここでは構造の情報を持つ隣接行列を  $\mathbf{S} \in \mathbb{R}_+^{n \times n}$ 、属性の情報を持つ行列を  $\mathbf{A} \in \mathbb{R}_+^{n \times m}$  で表すものとする。 $\mathbf{A}$  の行はノードを表し、列は属性を表している。加えて、各属性のための重みを割り当てる対角行列  $\mathbf{\Lambda} \in \mathbb{R}_+^{m \times m}$  を提案している。グラフクラスタリングで高い性能を示すことで知られている SymNMF [2] を参考に、 $\mathbf{S} \approx \mathbf{V}\mathbf{V}^T$  と  $\mathbf{A}\mathbf{\Lambda} \approx \mathbf{V}\mathbf{U}^T$  のように 2 つの行列分解を行う。ここでの  $\mathbf{V} \in \mathbb{R}_+^{n \times k}$ ,  $\mathbf{U} \in \mathbb{R}_+^{m \times k}$  であり、 $k$  は行列分解の

際を用いる潜在変数の数を表す。 $\mathbf{V}$  はノードについての構造と属性の情報を考慮した表現である。式 3 に JWNMF の目的関数を示す。

$$\min_{\mathbf{V}, \mathbf{U}, \mathbf{\Lambda} \geq 0} \frac{1}{2} (\|\mathbf{S} - \mathbf{V}\mathbf{V}^T\|_F^2 + \lambda \|\mathbf{A}\mathbf{\Lambda} - \mathbf{V}\mathbf{U}^T\|_F^2) \quad (3)$$

$\lambda > 0$  は構造と属性を統合する重みである。 $\mathbf{V}, \mathbf{U}, \mathbf{\Lambda}$  それぞれについて目的関数を偏微分し、その勾配に基づいて更新して最適化を行う。求めた  $\mathbf{V}$  に k-means を適用し、最終的なクラスタリング結果を得る。JWNMF では、各属性の重みを自動で最適化する行列を導入することによって、属性に含まれるノイズへの頑健性を得た。しかし、この手法では行列分解の線形性から線形分離可能な属性の寄与しか考慮することができない。

JWNMF は構造については SymNMF と同様に 1 つの行列を用いてオートエンコーダで表現し、属性については 2 つの行列を用いてエンコードしていると考えることができる。

### 2.3 ドロップアウト

本節では、深層学習で用いられる技術であるドロップアウト [12] について述べる。ドロップアウトでは、ニューラルネットワークの学習を行う際に、層の中のノードを事前に決定した割合だけ無効にする。次の更新では無効にするノードを選び直し、学習を行う。これによって学習時のネットワークの自由度を抑え、汎化性能を向上させることが知られている。隠れ層において、一般的に 50% 程度のノードを無効にすると良いと言われている。[12] によると全結合層だけでなく、畳み込み層などに適用した場合においても同様に性能を向上させることが確かめられている。ドロップアウトが高性能を実現する理由として、アンサンブル学習の近似であることが挙げられる。すなわち、ドロップアウトはモデルサイズを増大させることなくアンサンブル学習の近似を行い、モデルの汎化性能を向上している。学習ごとに無効にしているノードがあることから学習速度が遅く、同じサイズの通常モデルの学習に比べて多くの反復が必要になる。

## 3. 提案手法

本稿では、属性付きグラフを行列分解を用いてクラスタリングする際に、構造と属性の両方の情報を利用するとともに属性に関して非線形関数を導入することによってより詳細な分析を可能にした NJNMF を提案する。前章と同様に、 $\mathbf{S}$  を隣接行列、 $\mathbf{A}$  を属性行列、 $\mathbf{\Lambda}$  を各属性の重み決める対角行列とする。構造については SymNMF に影響を受け、 $\mathbf{S} \approx \mathbf{V}\mathbf{V}^T$  のように分解する。属性については従来の手法より詳細な分析を行うために中間層  $\mathbf{\Theta}$  および活性化関数  $f(\cdot)$  を導入し、 $\mathbf{A}\mathbf{\Lambda} \approx f(\mathbf{V}\mathbf{\Theta})\mathbf{U}^T$  のように分解する。このモデルを 2 つの連結した 3 部グラフとして考えたときの構造を図 1 に示す。

またこれら 2 つの行列分解を統合した目的関数を式 4 に示す。式 4 の第一項は構造に関する誤差を表し、第二項は属性に関する誤差を表している。

$$\min_{\mathbf{V}, \mathbf{U}, \mathbf{\Lambda}, \mathbf{\Theta} \geq 0} \frac{1}{2} (\|\mathbf{S} - \mathbf{V}\mathbf{V}^T\|_F^2 + \lambda \|\mathbf{A}\mathbf{\Lambda} - f(\mathbf{V}\mathbf{\Theta})\mathbf{U}^T\|_F^2) \quad (4)$$

ここでは  $\mathbf{S} \in \mathbb{R}_+^{n \times n}$ ,  $\mathbf{A} \in \mathbb{R}_+^{n \times m}$ ,  $\mathbf{V} \in \mathbb{R}_+^{n \times k}$ ,  $\mathbf{U} \in \mathbb{R}_+^{m \times h}$ ,  $\mathbf{\Theta} \in$

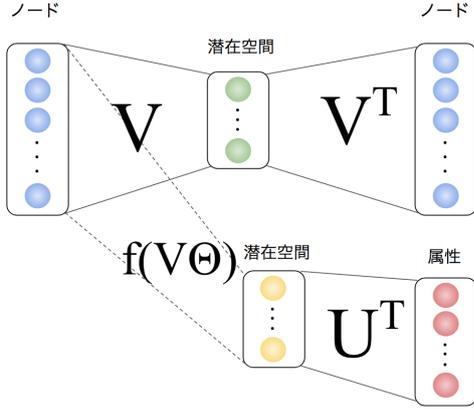


図1 NJNMF のモデル図

$\mathbb{R}_+^{k \times h}$ ,  $f(\cdot) : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ ,  $\mathbb{R}_+$  は非負の実数集合を表し,  $n$  はノード数,  $m$  は属性の種類を表す. 対角行列  $\Lambda$  は  $\sum_{i=1}^m \Lambda_{i,i} = 1$  を満たし,  $\lambda > 0$  は構造と属性を統合する重みである.  $k$  は抽出するクラスタ数を表し,  $h$  は中間層のユニット数を表す. 活性化関数  $f(\cdot)$  はステップ, シグモイド, ReLU, ソフトマックス, 恒等関数など多くの種類が考えられる. 線形関数を用いた場合, 同様の振る舞いを行う 2 層のモデルが存在するので本手法では非線形関数であり微分が容易であるシグモイド関数を採用する. 式 5 にシグモイド関数を示す.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

構造と属性の情報のうちどちらか一方だけの影響が強くなり過ぎないようにするため式 4 の最適化を行う前に,  $\mathbf{S}$  と  $\mathbf{A}$  について式 6 に示す正規化を行う.

$$\mathbf{S} = \frac{\mathbf{S}}{\sum_{i=1}^n \sum_{j=1}^n S_{i,j}}, \mathbf{A} = \frac{\mathbf{A}}{\sum_{i=1}^n \sum_{j=1}^m A_{i,j}} \quad (6)$$

### 3.1 更新規則

次に  $\mathbf{V}, \mathbf{U}, \Lambda$  および  $\Theta$  の更新規則について述べる.  $\alpha, \beta, \gamma$  および  $\delta$  をそれぞれ  $\mathbf{V} \geq 0, \mathbf{U} \geq 0, \Lambda \geq 0$  および  $\Theta \geq 0$  の制約のためのラグランジェの未定乗数行列とする. ラグランジェの方程式を用いて, 式 4 で得た目的関数を損失関数として書き換えると式 7 が得られる.

$$L = \frac{1}{2} (\|\mathbf{S} - \mathbf{V}\mathbf{V}^T\|_F^2 + \lambda \|\Lambda\mathbf{A} - f(\mathbf{V}\Theta)\mathbf{U}^T\|_F^2) + \text{Tr}(\alpha^T \mathbf{V}) + \text{Tr}(\beta^T \mathbf{U}) + \text{Tr}(\gamma^T \Lambda) + \text{Tr}(\delta^T \Theta) \quad (7)$$

この損失関数を  $\mathbf{V}, \mathbf{U}, \Lambda, \Theta$  それぞれについて偏微分を行い, 得た勾配を式 8, 式 9, 式 10 および式 11 に示す.

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{V}} = & -(\mathbf{S}\mathbf{V} + \mathbf{S}^T \mathbf{V} + \lambda((\Lambda\mathbf{A}\mathbf{U}) \circ (f'(\mathbf{V}\Theta))\Theta^T)) \\ & + (2\mathbf{V}\mathbf{V}^T \mathbf{V} + \lambda((f(\mathbf{V}\Theta)\mathbf{U}^T \mathbf{U}) \circ (f'(\mathbf{V}\Theta))\Theta^T) \\ & + \alpha) \end{aligned} \quad (8)$$

$$\frac{\partial L}{\partial \mathbf{U}} = -\lambda \Lambda \mathbf{A}^T f(\mathbf{V}\Theta) + \lambda \mathbf{U} f(\mathbf{V}\Theta)^T f(\mathbf{V}\Theta) + \beta \quad (9)$$

$$\frac{\partial L}{\partial \Lambda} = -\lambda \mathbf{A}^T f(\mathbf{V}\Theta) \mathbf{U}^T + \lambda \mathbf{A}^T \Lambda \mathbf{A} + \gamma \quad (10)$$

$$\begin{aligned} \frac{\partial L}{\partial \Theta} = & -\lambda \mathbf{V}^T ((\Lambda\mathbf{A}\mathbf{U}) \circ (f'(\mathbf{V}\Theta))) \\ & + \lambda \mathbf{V}^T ((f'(\mathbf{V}\Theta) \circ f(\mathbf{V}\Theta)) \mathbf{U}^T \mathbf{U}) + \delta \end{aligned} \quad (11)$$

ここでは,  $\circ$  はアダマール積を表している. Karush-Kuhn-Tucker (KKT) 条件から  $\alpha_{p,r} V_{p,r} = 0, \beta_{p,r} U_{p,r} = 0, \gamma_{q,q} \Lambda_{q,q} = 0$  および  $\delta_{p,r} \Theta_{p,r} = 0$  とすると,  $\frac{\partial L}{\partial \mathbf{V}} = 0, \frac{\partial L}{\partial \mathbf{U}} = 0, \frac{\partial L}{\partial \Lambda} = 0, \frac{\partial L}{\partial \Theta} = 0$  である. これらの条件に基づいて導出した更新式を式 12, 式 13, 式 14 および式 15 に示す.

$$\begin{aligned} \mathbf{V} \leftarrow & \mathbf{V} \circ (\mathbf{S}\mathbf{V} + \mathbf{S}^T \mathbf{V} + \lambda((\Lambda\mathbf{A}\mathbf{U}) \circ (f'(\mathbf{V}\Theta))\Theta^T)) \\ & ./ (2\mathbf{V}\mathbf{V}^T \mathbf{V} + \lambda((f(\mathbf{V}\Theta)\mathbf{U}^T \mathbf{U}) \circ (f'(\mathbf{V}\Theta))\Theta^T)) \end{aligned} \quad (12)$$

$$\mathbf{U} \leftarrow \mathbf{U} \circ (\Lambda \mathbf{A}^T f(\mathbf{V}\Theta)) ./ (f(\mathbf{V}\Theta)^T f(\mathbf{V}\Theta)) \quad (13)$$

$$\Lambda \leftarrow \Lambda \circ (\mathbf{A}^T f(\mathbf{V}\Theta) \mathbf{U}^T) ./ (\mathbf{A}^T \Lambda \mathbf{A}) \quad (14)$$

$$\begin{aligned} \Theta \leftarrow & \Theta \circ (\mathbf{V}^T ((\Lambda\mathbf{A}\mathbf{U}) \circ (f'(\mathbf{V}\Theta)))) \\ & ./ (\mathbf{V}^T ((f'(\mathbf{V}\Theta) \circ f(\mathbf{V}\Theta)) \mathbf{U}^T \mathbf{U})) \end{aligned} \quad (15)$$

ここでは,  $./$  は要素ごとの除算を表している.  $\Lambda$  の重みを正規化空間へ割り当てるために, 式 16 に示すように  $\Lambda$  を正規化する.

$$\Lambda = \frac{\Lambda}{\sum_{i=1}^m \Lambda_{i,i}} \quad (16)$$

NJNMF は構造について 1 つの行列を用いてオートエンコーダで表現し, ノードと属性の間を多層で表現するため深層学習の一種として考えることができる. このモデルの複雑さによって起こりうる過学習に対応するために以下の 2 つの方法をそれぞれ適用する. 1 つ目は, 深層学習で用いられるドロップアウト [12] と同様に中間層にあたる  $\Theta$  と  $\mathbf{U}$  の間の層のユニットを学習ごとに  $0 < p \leq 1$  の割合だけ残して実行する方法である. 2 つ目は,  $\Theta$  はノードと属性に直接接続しておらず, 学習の際の自由度が高くなることが考えられることから, 深層学習の正規化のために提案されたドロップコネクト [13] のように  $\Theta$  の要素のうち  $0 < q \leq 1$  の割合だけ更新を行う方法である. 次に上記の更新規則の収束性と計算時間について分析する. 収束性については [14] で示されている補助関数を用いて証明することができる. 加えて目的関数の定常点を求める KKT 条件もまた更新規則の収束性を示している. 学習の反復回数を  $t$  とすると, NJNMF の全体の計算量は  $O(((n^2 + nm + nh)k + (m^2 + nm)h)t)$  である.

### 3.2 Non-Linear Joint Weighted NMF アルゴリズム

NJNMF アルゴリズムを Algorithm1 で説明する. 1 行目で隣接行列  $\mathbf{S}$  と属性行列  $\mathbf{A}$  に前処理を行い, 2 行目で行列  $\mathbf{V}, \mathbf{U}$  および  $\Theta$  をランダム値, 対角行列  $\Lambda$  の値を  $1/m$  で初期化する. 3-8 行目までで行列  $\mathbf{V}, \mathbf{U}, \Lambda$  および  $\Theta$  を収束するまで繰り返し更新する. 最後に 9 行目で分解結果である  $\mathbf{V}$  からクラスタリング結果を得る.  $\mathbf{V}$  からクラスタリング結果を導出する手続きとして, ノードごとに  $\mathbf{V}$  の対応する行で最大の値を持つクラスタに割り当てる最大エントリ法もしくは  $k$  平均法を用いる.

---

**Algorithm 1** NJNMF アルゴリズム

---

**Input:**  $S, A$ **Output:** clustering result  $C$ 

```
1: Preprocess:  $S, A$ 
2: Initialize:  $V, U, \Lambda, \Theta$ 
3: while not(converge) do
4:    $V^{(t+1)} \leftarrow \text{update}(V^{(t)})$ 
5:    $U^{(t+1)} \leftarrow \text{update}(U^{(t)})$ 
6:    $\Lambda^{(t+1)} \leftarrow \text{update}(\Lambda^{(t)})$ 
7:    $\Theta^{(t+1)} \leftarrow \text{update}(\Theta^{(t)})$ 
8: end while
9:  $C \leftarrow \text{max\_entry}(V)$  or  $\text{kmeans}(V)$ 
```

---

## 4. 評価実験

この章では、まずクラスタリング結果の導出のために最大エントリを用いる場合と  $k$  平均法で割り当てる場合を比較する。その後、ドロップアウトを適用したものを含む 3 つの提案手法の性能実験を行い、属性付きグラフクラスタリング手法である JWNMF と構造の情報のみに着目したモジュラリティクラスタリング [15] および属性のみに着目した  $k$  平均法と比較する。

### 4.1 データセット

本実験では 3 つの実データを用いる。Disney<sup>(注2)</sup> および Amazon Fail<sup>(注3)</sup> はアマゾンの商品データのネットワークであり、Enron(注 3 を参照) はメールアドレスのネットワークである。上記 3 つのグラフの属性はいずれも離散値と連続値の両方を含む。用いたデータセットの詳細な情報を表 1 に示す。

表 1 データセット

データセット	ノード数	エッジ数	属性数
Disney	124	335	25
Amazon Fail	418	3695	21
Enron	13533	176987	18

### 4.2 評価指標

属性付きグラフクラスタリングの目標は構造と属性の両方の情報から類似するデータを抽出することなので、2 つの観点で結果を評価する。構造の観点からクラスタリング結果を評価するためにモジュラリティ [16] を用いて、属性の観点から評価するために平均エントロピー (注 1 を参照) を用いる。モジュラリティはグラフ分割の基準として広く用いられている。平均エントロピーは属性を用いたクラスタリング結果の評価に用いられる指標である。モジュラリティが高いほどクラスタ内のノード間での接続が密であり、異なるクラスタに含まれるノード間での接続が疎であることを示している。平均エントロピーが低いほどクラスタ内のノードが似た属性値を持ち、異なるクラスタに含まれるノードは異なる属性値を持っている。

$C = (C_1, C_2, \dots, C_k)$  が属性付きグラフの  $k$  個の分割を表すとすると、モジュラリティ  $Q$  と平均エントロピー  $Avg\_entropy$

をそれぞれ式 17, および式 18 に示す。

$$Q = \sum_{i=1}^k (e_{i,i} - c_i^2) \quad (17)$$

$$Avg\_entropy = \sum_{t=1}^m \sum_{j=1}^k \frac{|C_j|}{nm} \text{entropy}(a_t, C_j) \quad (18)$$

ここでの  $e_{ij}$  はクラスタ  $i$  に含まれるノードを始点として、クラスタ  $j$  に含まれるノードを終点とするエッジの割合である。  $c_i$  はクラスタ  $i$  に含まれるノードに接続するエッジの終点の割合を示している。  $\text{entropy}(a_t, C_j)$  はクラスタ  $C_j$  の中で属性  $a_t$  の情報エントロピーである。本実験で用いるデータセットは連続地である属性を含むため、直接情報エントロピーを適用することができない。そのため、以下の 2 つの指標を用いた。1 つ目は微分エントロピー [17] であり、同じクラスタに含まれたノードが持つ属性値の分散を用いて算出する。2 つ目は値域を  $r$  等分し、離散化してから情報エントロピーを用いる。最も簡単な離散化である  $\{0, 1\}$  に置き換えた場合より連続値の情報を残す一例として  $r = 10$  とした。

### 4.3 実験設定

ドロップアウトを適用したものを NJNMF-out と呼び、ドロップコネクトを適用したものを NJNMF-con と呼ぶこととする。本実験では、ドロップアウトのパラメータである  $p$  は [12] で中間層に適用する場合に経験的に良いとされる  $p = 0.5$  とし、ドロップコネクトのパラメータである  $q$  についても [13] で経験的に良いとされる  $q = 0.5$  とした。学習の反復回数は実験的に十分収束した 100 回に固定し、モデルに初期値依存性があるため評価は異なる初期値で行なった 5 回の実験の平均値とした。構造と属性を統合するための重み  $\lambda$  を  $\{10^{-10}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{0.5}\}$  の集合の中からデータごとに最適なものを実験で決定した。また NJNMF で設定が必要な中間層のユニット数  $h$  を  $\{5, 8, 10, 15, 20\}$  の集合の中からデータごとに最適なものを実験で決定した。  $k$  平均法は属性行列を利用した属性 kmeans と属性ごとにその最大値が 1 になるように正規化したものを利用した kmeans-reg をそれぞれ実験した。  $k$  平均法は本実験で用いた他の手法に比べモジュラリティが非常に低かったためモジュラリティを比較したグラフには表示しない。

### 4.4 クラスタ割当方法の比較

本節では、反復更新によって学習した  $V$  からクラスタリング結果を得るために用いる最大エントリと  $k$  平均法を比較する。図 2, 3 では、JWNMF および NJNMF は最大エントリ法を用いたものであり、JWNMF-kmeans および NJNMF-kmeans は  $k$  平均法によってクラスタリング結果を得たものである。これらの図から最大エントリを用いた場合、  $k$  平均法を用いた場合に比べほとんどの地点で、高いモジュラリティと低い平均エントロピーを実現している。  $k$  平均法を用いた場合、次数の小さいノードが集まってクラスタを構築する可能性がある。一方で、最大エントリ法を用いた場合ではノードごとに最も関連の強い  $V$  の列に属するので、次数の小さなノードも最も関連が強いクラスタに属することになるため高い性能が得られたと考えられる。

---

(注2) : <http://www.perozzi.net/projects/focused-clustering/>.

(注3) : <http://www.ipd.kit.edu/~muellere/consul/>.

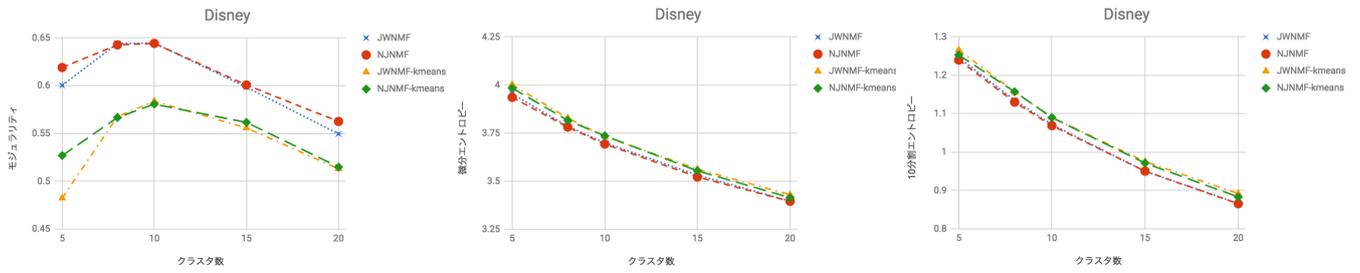


図 2 Disney データでの最大エントリと  $k$  平均法の比較

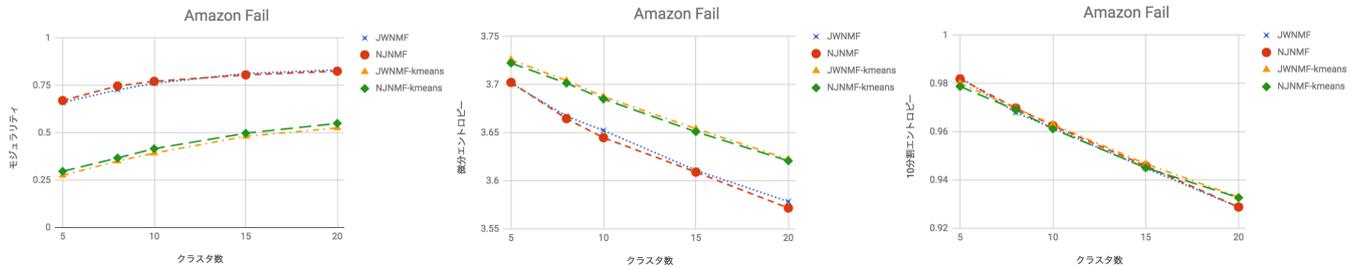


図 3 Amazon データでの最大エントリと  $k$  平均法の比較

#### 4.5 他のクラスタリング手法との比較

前節の実験結果より、以下の実験では最大エントリからクラスタリング結果を得る方法を用いて、他の手法と比較実験を行う。Disney データに対してクラスタ数を変えながらモジュラリティ、微分エントロピーおよび 10 分割エントロピーの変化をプロットしたグラフを図 4 に示す。Disney データでは、NJNMF が既存手法の JWNMF をモジュラリティの観点で上回っていることがわかる。エントロピーについては、NJNMF は JWNMF と同等の性能を示し、構造のみを考慮してクラスタリングを行う modularity-based より良いエントロピーを実現している。属性 kmeans と kmeans-reg の間でエントロピーに大きな開きがある原因として、属性間で値域に差があることが挙げられる。属性 kmeans は大きい値をとる属性から大きな影響を受けるため、全ての属性を平等に扱う平均エントロピーが大きくなったと考えられる。

Amazon データに対してクラスタ数を変えながらモジュラリティ、微分エントロピーおよび 10 分割エントロピーの変化をプロットしたグラフを図 5 に示す。modularity-based の手法がモジュラリティの面で高い性能を示し、エントロピーに関しては属性を考慮した NJNMF や JWNMF と同等の性能を示していることから、Amazon データではモジュラリティを高めるようにクラスタリングを行えば、エントロピーも良い結果が得られることがわかる。また属性 kmeans と kmeans-reg はともにエントロピーの観点で良い結果が得られていることから正規化を行う前から、属性値に大きな偏りがないことがわかる。

Enron データでは NJNMF が JWNMF をモジュラリティおよびエントロピーを実験した全てのクラスタ数で上回る性能を示した。属性 kmeans に着目すると、微分エントロピーでは他の手法に比べて高い性能を示している一方で、10 分割エントロピーでは他の手法に比べて低い性能を示している。この結果から Enron のデータに対して  $k$  平均法を実行したとき、1 つのク

ラスタの中で属性に関して 2 つ以上の密集したグループが存在していることが考えられる。このような複雑な条件下では、NJNMF は属性について既存手法より詳細な分析が可能になることがわかる。特に NJNMF-out が 10 分割エントロピーの観点では、実験した全てのクラスタ数で良い性能を示した。これは NJNMF-out が提案手法の中で最も良い汎化性能を持ち、ノードと属性の間の非線形な関係をうまく捉えることができたためであると考えられる。Enron データに対し  $k$  平均法を適用し、属性の正規化を行ったもので行っていないものとで比較した結果を図 7 に示す。属性 kmeans と kmeans-reg の性能に差があることから、disney データと同様に属性間で値域に差があることが考えられる。

## 5. 関連研究

近年、属性付きグラフマイニングのためにいくつかの手法が提案されてきている。それらは主に距離ベース [8, 9] とモデルベース [10, 11] の 2 つに分類することができる。距離ベースの手法は構造と属性の両者から定義される距離基準を考慮することで、 $k$  平均法やスペクトラルクラスタリングといった既存手法を属性付きグラフクラスタリングへ拡張している。モデルベースの手法ではクラスタリングのために結合モデルを構築するためにエッジとノードの属性の相互関係を活用する。

### 5.1 距離ベースの手法

Zhou らが距離ベースの手法である **SA-Cluster** [8] を提案し、それを効率化した **Inc-Cluster** [9] を提案した。これら 2 つの手法は属性を新しいノードとして元のグラフに追加して新しいグラフを構築する。そのとき、属性の値が 0 でないノードにリンクを設ける。ランダムウォークを用いて拡張されたグラフの距離を設計し、 $k$ -medoids によってグラフを分割する。グラフを拡張するステップにおいてグラフのサイズを大きくすることから、大規模グラフで実行することは困難である。

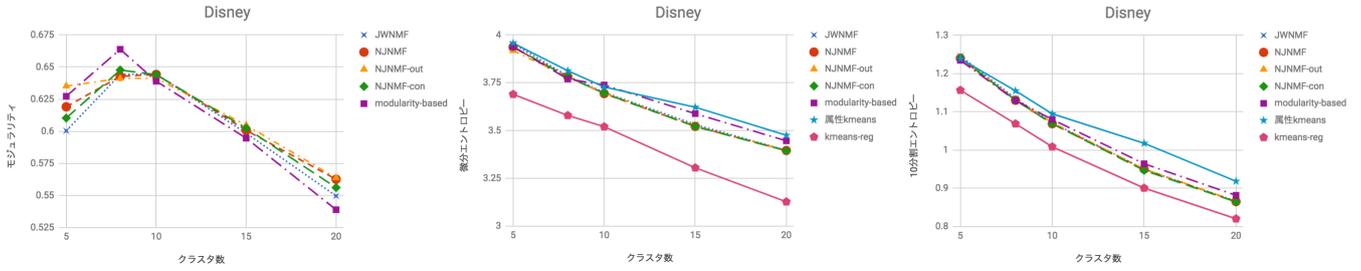


図 4 Disney データでのクラスタリング性能

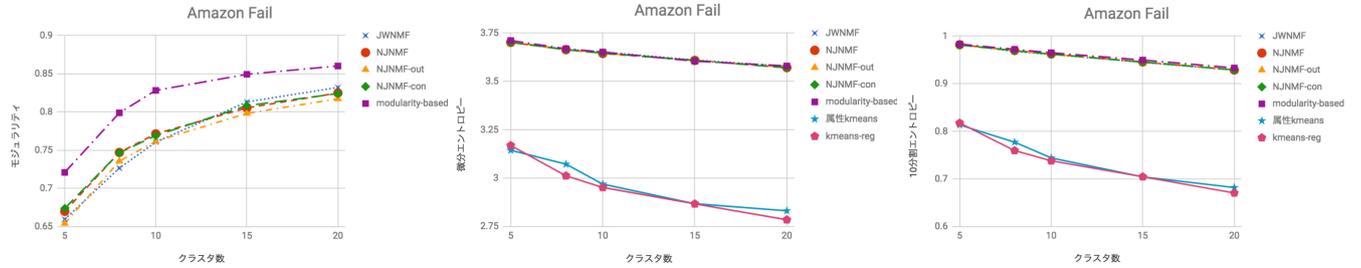


図 5 Amazon データでのクラスタリング性能

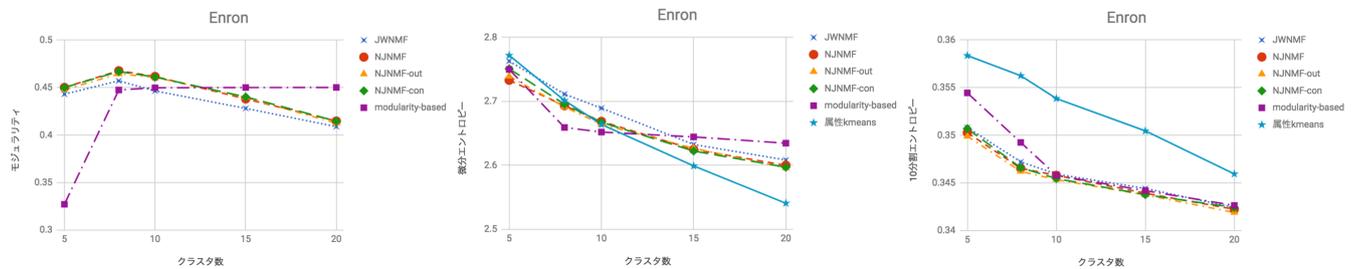


図 6 Enron データでのクラスタリング性能



図 7 Enron データでの  $k$  平均法の比較

## 5.2 モデルベースの手法

Xu らがモデルベースのアプローチである **BAGC** [10] を提案した。この手法はクラスタが各属性の多項分布とノードの結合のベルヌーイ分布を持っていると仮定することによってベイジアン確率モデルを構築する。結果として属性付きグラフクラスタリング問題は標準的な確率推定の問題へと変形させることができる。この手法の問題点としては重み付きの属性付きグラフを扱うことができないことが挙げられる。この問題を解決するために Xu らは後に BAGC を拡張し、**GBAGC** [11] を提案した。これらの手法では多重グラフを扱うことができないという欠点が挙げられる。

## 6. おわりに

本稿では、非線形関数を組み込んだモデルベースの属性付きグラフクラスタリングである NJMNF を提案し、その概要について示した。提案手法では、グラフの構造的な情報とノードの持つ属性を統合して行列分解を行う。属性についての行列を 3 つに分解し、中間層に活性化関数を導入することによって既存のモデルベースの手法に比べて詳細な分析を可能にした。またモデルの複雑さから生じる過学習に対応するために提案モデルへのドロップアウトの適用も考案した。実データを用いて実験を行い、ノードと属性の関係が複雑な場合に NJMNF が高いクラスタリング性能であることを示した。過学習を抑えるためにドロップアウトとドロップコネクトを適用して実験を行い、深

層学習で広く用いられているドロップアウトを用いたとき, 良い結果が得られることを示した.

## 文 献

- [1] Lee D.D.: Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* (2001)
- [2] Kuang D., Ding C., Park H.: Symmetric nonnegative matrix factorization for graph clustering. *SIAM* (2012)
- [3] Kuang D., Yun S., Park H.: SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering. *J. Glob. Optim.* (2015)
- [4] Z. Huang, Y. Ye, X. Li, F. Liu, and H. Chen: Joint Weighted Nonnegative Matrix Factorization for Mining Attributed Graphs. *PAKDD* (2017)
- [5] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger: Scan: a structural clustering algorithm for networks. *KDD* (2007)
- [6] S. Brohee, J. V. Helden: Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinf.* (2006)
- [7] S. Fortunato: Community detection in graphs. *Phys. Rep.* 486(3), 75-174 (2010)
- [8] Zhou Y., Cheng H., Yu J.X.: Graph clustering based on structural/attribute similarities. *Proc. VLDB* (2009)
- [9] Zhou Y., Cheng H., Yu J.X.: Clustering large attributed graphs: an efficient incremental approach. *ICDM* (2010)
- [10] Xu Z., Ke Y., Wang Y., Cheng H., Cheng J.: A model-based approach to attributed graph clustering. *SIGMOD* (2012)
- [11] Xu Z., Ke Y., Wang Y., Cheng H., Cheng J.: GBAGC: a general bayesian framework for attributed graph clustering. *TKDD* (2014)
- [12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *JMLR* (2014)
- [13] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, R. Fergus: Regularization of Neural Networks using DropConnect. *ICML* (2013)
- [14] Lee, D.D.: Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* (2001)
- [15] 藤森 俊匡, 塩川 浩昭, 鬼塚 真: 効率的な分散グラフ処理のためのグラフ分割. *DEIM* (2017)
- [16] M. E. J. Newman: Modularity and community structure in networks. *Proc. Natl Acad. Sci.* (2006)
- [17] C. M. Bishop: Pattern recognition and machine learning. New York, NY: Springer Verlag (2006)