

加重核型ノルム正則化に基づくグラフ単純化 Weighted Nuclear Norm Regularization based Graph Simplification

佐々木 崇元*
Takayuki Sasaki

坂東 幸浩*
Yukihiro Bandoh

北原 正樹*
Masaki Kitahara

1. はじめに

地理情報学や画像工学では、複数の頂点を辺で繋いだグラフに頂点座標が埋め込まれたデータ（以降、埋め込みグラフあるいは単にグラフ）を取り扱い、様々な形状データを表現する。グラフの代表例は、地理情報学であれば海岸線や境界線の情報、画像工学であれば関心領域やオブジェクト形状の情報である。これらのグラフはその形状により対象情報を表現し、頂点や辺が密であるほど精細に情報を伝達できる。

一方で過密な頂点や辺を持つグラフは、実用において様々な問題を引き起こすため、グラフの単純化が必要である。例えば海岸線情報であれば、表示装置の解像度や処理粒度以上に密な頂点や辺は、描画や処理に不要で、保存領域の圧迫や処理遅延の原因となる。グラフを単純化することで不要な頂点や辺を取り除き、選別した少ない頂点や辺で形状を近似すれば、上記問題は解決される。

グラフ単純化の最も基本的な手法は、多角形や連続線分の単純化を行う Douglas-Peucker (DP) 法 [1, 2] であり、地理情報学、画像工学で広く利用されている。DP 法は近似誤差が減少する様に順次頂点と辺を選択して加えていく貪欲法である。DP 法は実装が容易かつ高速に動作するため、両分野のソフトウェアで標準的に実装されている。

しかし DP 法は雑音の影響を受けやすく、入力グラフが観測誤差等を含む場合に適切な出力を得られないという問題がある。DP 法は入力グラフの頂点から選択して出力頂点を決定する手法なので、選択される頂点によって形状がよく近似できることが求められる。従って、全ての頂点位置が雑音により劣化している場合は形状をよく近似する頂点を選択できず、所望の出力結果を得られない。

そこで筆者らは局所線形近似法 [3] を提案し、雑音に頑健なグラフ単純化手法を実現している。局所線形近似法は図 1 に示す様に、局所線形整列と不要頂点除去の 2 つの処理から構成される手法である。最初に局所線形整列では、グラフの連結構造を保ったまま各頂点の座標を微小に動かすことで、グラフ形状を近似する少数の線分上に頂点を整列させる。続いて不要頂点除去では、局所線形整列させたグラフから分岐や角などの線分の端点を構成する頂点以外を除去する。以上の 2 つの処理により、グラフ単純化を実現できる。

この中で局所線形整列は数理最適化に基づく処理で、その目的関数は雑音の特性に対応した忠実化項とグラフの折れ曲がりを抑制する正則化項から成る。特に [3]

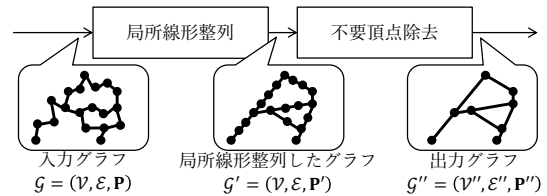


図 1: 局所線形近似法の概略図

では正則化項として、核型ノルム正則化に基づく正則化関数を提案している。これに雑音の種類に応じた忠実化関数を組み合わせることで、雑音に頑健な局所線形整列を達成できる。

しかしながら局所線形近似法によるグラフ単純化は、グラフの急峻な折れ曲がり部において頂点数を削減できず、かつ形状の近似精度が下がる問題があった。結果として急峻な折れ曲がり部は多くの線分で近似され、丸まった形状となってしまった。この現象は、局所線形整列において正則化項に用いた核型ノルムがグラフ全体の辺の長さを正則化する効果を持っているため、発生したと推察される。このため、核型ノルムを適切な他のノルムに置き換えることで、改善される見込みがある。

また局所線形近似法では並列計算が難しく、頂点数が多い大型のグラフにおいて、多くの計算時間を要する問題があった。局所線形整列問題の解法としては交互方向乗数法 (ADMM) [4] などの近接分離法が用いられる。折れ曲がりの正則化関数が核型ノルムにより構成されるため、その近接写像である特異値閾値処理 [5] が必要になる。局所線形整列問題では小型多数の行列に特異値閾値を繰り返し施すが、この計算を並列処理することが難しく、多くの処理時間を要した。

そこで筆者らは小型多数行列に対する高速並列特異値閾値処理法 [6, 7] を提案している。特異値閾値処理の算出過程から特異値分解を排除して、計算のほとんどを計算並列性の高い線形処理で構成した。これにより、局所線形近似法の処理時間を大幅に短縮することに成功した。しかし高速並列特異値閾値処理法は、核型ノルムを折れ曲がりの正則化に用いた場合にのみ適用できるものであり、上述の様に他のノルムに置き換えるならば、そのノルムの近接写像に適合する高速並列計算法を新たに考案する必要がある。

本研究では加重核型ノルム正則化を用いた局所線形近似に基づくグラフ単純化を提案する。局所線形整列の正則化関数を加重核型ノルムに置き換えることで、辺の長さの正則化を抑制し、従来法と比較してより頂点数が少なく近似精度の高いグラフ単純化を達成する。また計算速度改善のために、2 つの高速化法についても

*日本電信電話株式会社
NTT コンピュータ&データサイエンス研究所
Nippon Telegraph and Telephone Corporation
NTT Computer and Data Science Laboratories

提案する. 1つ目は加重核型ノルムにおける高速並列特異値閾値処理法を新たに提案し, 計算速度の第 1 のボトルネックを解消する. 2つ目は ADMM における反復的な線形方程式の求解を計算量削減し, 計算速度の第 2 のボトルネックを解消する.

2. 従来の局所線形近似法によるグラフ単純化

本節では本研究に最も関連する従来法である局所線形近似法 [3] について説明する. まず本節以降に共通する数学的な記法や定義について述べ, 続いて局所線形近似法の概要について説明する. 続いて局所線形近似法を高速化させるための特異値閾値処理の計算法について述べ, 最後に局所線形近似法の問題点について解析的に示す.

2.1. 記法と定義

実数集合を \mathbb{R} と表し, 太字小文字, 大文字のアルファベットはそれぞれベクトルと行列を意味する (例: \mathbf{a}, \mathbf{A}). ベクトルや行列の転置は右上に \mathbf{T} を添える. ベクトルのユークリッドノルムや行列のフロベニウスノルムを $\|\cdot\|$ と表記する.

ベクトルの各要素を非負にクリッピングする関数をランプ関数と呼び, $(\cdot)_+$ と表記する. 例えば $\mathbf{y} = \mathbf{x}_+$ であれば, 各要素は $y_i = \max(x_i, 0)$ である.

ベクトルあるいは行列の集合を \mathcal{X} としたとき, 関数 $f: \mathcal{X} \rightarrow \mathbb{R}$ による近接写像 [8] とは, 下記式で定義される最適化問題の解である.

$$\text{prox}_f(\mathbf{x}) = \arg \min_{\mathbf{y} \in \mathcal{X}} f(\mathbf{y}) + \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|^2 \quad (1)$$

上記最適化問題の解を効率的に求められる関数 f を, 近接写像可能な関数という.

2.2. 局所線形近似法の概要

埋め込みグラフを $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{P})$ とする. ここで $\mathcal{V} = \{v_i\}_{i=1}^N$ は頂点の集合であり N は頂点数である. また各頂点 v_i は D 次元空間に埋め込まれているとし, 座標を $\mathbf{p}_i \in \mathbb{R}^D$ とする. 全ての頂点座標を並べた行列を $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_N]^T \in \mathbb{R}^{N \times D}$ とする. $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ は辺の集合であり, 辺 $e = \{v_i, v_j\}$ が \mathcal{E} の要素であるとき頂点 v_i, v_j は隣接している.

局所線形近似法は図 1 の様に, 局所線形整列, 不要頂点除去の 2 ステップから構成される. 局所線形整列は座標 \mathbf{P} のみを変化させた埋め込みグラフ $\mathcal{G}' = (\mathcal{V}, \mathcal{E}, \mathbf{P}')$ を獲得する. 新たな座標 \mathbf{P}' を得るための最適化問題は,

$$\mathbf{P}' = \arg \min_{\mathbf{X} \in \mathbb{R}^{N \times D}} \frac{1}{2} \|\mathbf{P} - \mathbf{X}\|^2 + \lambda \sum_{v \in \bar{\mathcal{V}}} \|\mathbf{L}_v \mathbf{X}\|_* \quad (2)$$

である. 目的関数の第 1 項は忠実化項で, 第 2 項はグラフの折れ曲がりを正則化する項である. ここで $\bar{\mathcal{V}} \subseteq \mathcal{V}$ は次数が 2 の頂点の集合である. 行列 $\mathbf{L}_v \in \mathbb{R}^{2 \times N}$ は, 頂点 $v \in \bar{\mathcal{V}}$ についての隣接差分信号 $\mathbf{L}_v \mathbf{X}$ を生成する疎行列である. 頂点 v に関するインデックスを j_v とし, 頂点 v に隣接する頂点に関するインデックスを i_v, k_v とする. このとき \mathbf{L}_v の非零要素は,

$$(\mathbf{L}_v)_{1, j_v} = (\mathbf{L}_v)_{2, j_v} = -1 \quad (3)$$

$$(\mathbf{L}_v)_{1, i_v} = (\mathbf{L}_v)_{2, k_v} = 1 \quad (4)$$

である. また $\|\cdot\|_*$ は核型ノルムで入力行列の特異値の和である. すなわち核型ノルム $\|\mathbf{Y}\|_*$ は, 入力行列 $\mathbf{Y} \in \mathbb{R}^{2 \times D}$ の特異値を $\sigma_1, \sigma_2 (\sigma_1 \geq \sigma_2 \geq 0)$ として,

$$\|\mathbf{Y}\|_* = \sigma_1 + \sigma_2 \quad (5)$$

である [9]. 核型ノルムはランク関数の凸包であり, 行列の低ランク性を推進する正則化関数である. 最適化問題 (2) においては, 隣接差分信号 $\mathbf{L}_v \mathbf{X}$ を低ランクにすることで, グラフの折れ曲がりを正則化している.

最適化問題 (2) の目的関数は凸であり, 主双対分離法 (PDS) 法 [10] や ADMM [4] などの近接分離法で解くことができる. 近接分離法では核型ノルムの近接写像である特異値閾値処理を反復的に算出する. 特異値閾値処理とは, 特異値分解とソフト閾値処理と行列積の 3 つの計算から構成される処理で,

$$\text{prox}_{\mu \|\cdot\|_*}(\mathbf{Y}) = \mathbf{U} \text{diag}((\sigma - \mu)_+) \mathbf{V}^T \quad (6)$$

で定義される. ここで $\sigma = [\sigma_1, \sigma_2]^T$, $\mathbf{U} \in \mathbb{R}^{2 \times 2}$, $\mathbf{V} \in \mathbb{R}^{D \times 2}$ は特異値分解で, $\mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{V}^T \mathbf{V} = \mathbf{I}$ と

$$\mathbf{Y} = \mathbf{U} \text{diag}(\sigma) \mathbf{V}^T \quad (7)$$

を満たす.

特異値閾値処理は特異値分解を含むため計算量が多く, また複数の行列が入力されたときに並列に算出できない. このため多くの実行時間を要し, 局所線形整列の計算時間の大部分を占める.

不要頂点除去は埋め込みグラフ $\mathcal{G}' = (\mathcal{V}, \mathcal{E}, \mathbf{P}')$ を入力として, 各頂点に接続する 2 辺の成す角度を基準に頂点を除去し, 単純化された埋め込みグラフ $\mathcal{G}'' = (\mathcal{V}'', \mathcal{E}'', \mathbf{P}'')$ を求める.

2.3. 特異値閾値処理の高速計算

本小節では特異値閾値処理 (6), (7) の高速計算法 [6, 7] について述べる. 記述を簡単にするため, 入力 \mathbf{Y} は D 行 2 列の行列とする. $\text{prox}_{\mu \|\cdot\|_*}(\mathbf{Y}^T)^T = \text{prox}_{\mu \|\cdot\|_*}(\mathbf{Y})$ が成立するため, 2 行 D 列の行列が入力の時は D 行 2 列入力の特異値閾値処理をする前後に転置操作を施すだけでよい.

行列 \mathbf{Y} の列ベクトルを $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^D$ とする (すなわち $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2] \in \mathbb{R}^{D \times 2}$). このとき特異値閾値処理は以下の通り算出することができる.

$$\text{prox}_{\mu \|\cdot\|_*}(\mathbf{Y}) = \gamma(1 - \delta) \mathbf{Y} + \gamma \delta \mathbf{R}[\mathbf{y}_2, -\mathbf{y}_1] \quad (8)$$

ここで γ, δ は

$$\gamma = \begin{cases} \left(1 - \frac{(\mu - \sigma_2)_+}{\sigma_1 - \sigma_2}\right)_+ & \text{if } \sigma_1 - \sigma_2 \neq 0 \\ \left(1 - \frac{\sqrt{2}\mu}{\|\mathbf{Y}\|}\right) & \text{else if } \sigma_1 + \sigma_2 \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$$\delta = \begin{cases} \frac{\min(\mu, \sigma_2)}{\sigma_1 + \sigma_2} & \text{if } \sigma_1 + \sigma_2 \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

である. また特異値の和及び差は

$$\sigma_1 \pm \sigma_2 = \sqrt{\text{tr} \mathbf{Y}^T \mathbf{Y} \pm 2 \sqrt{\det \mathbf{Y}^T \mathbf{Y}}} \quad (11)$$

により算出できる. また行列 \mathbf{R} は $\pi/2$ [rad] 回転行列で,

$$\mathbf{R} = \begin{cases} \frac{1}{\sqrt{\det \mathbf{Y}^T \mathbf{Y}}} (\mathbf{y}_2 \mathbf{y}_1^T - \mathbf{y}_1 \mathbf{y}_2^T) & \text{if } D \geq 3 \\ \text{sgn}(\det \mathbf{Y}) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} & \text{else if } D = 2 \end{cases} \quad (12)$$

である.

以上の算出法は (6),(7) の方法と比較して特異値分解が不要であるので, 高速に特異値閾値処理を実行できる. 計算量は浮動小数点演算で 60 – 84% 削減されている. また複数の行列が入力されたとき, (8) がベクトル $\mathbf{y}_1, \mathbf{y}_2$ についての線形変換であり, 変換係数はベクトルの要素についての四則演算, 冪乗及び根号計算のみで構成されていることから, 並列に特異値閾値処理を算出できる. これは近代の CPU や GPU における SIMD アーキテクチャにより効率的に計算される.

2.4. 局所線形近似法の問題点

局所線形近似法の特性を解析的な観点から考察し, その問題点を明らかにする. 局所線形整列において折れ曲がりの回数を正則化する関数として核型ノルムを用いている. 核型ノルムの入力行列を $\mathbf{Y} \in \mathbb{R}^{D \times 2}$ とし特異値を σ_1, σ_2 とすると (11) は, $\text{tr} \mathbf{Y}^T \mathbf{Y} \gg \det \mathbf{Y}^T \mathbf{Y}$ であることより, $\sqrt{1+x} \simeq 1 + \frac{1}{2}x$ の近似を用いて,

$$\sigma_1 \pm \sigma_2 \simeq \sqrt{\text{tr} \mathbf{Y}^T \mathbf{Y}} \pm \sqrt{\frac{\det \mathbf{Y}^T \mathbf{Y}}{\text{tr} \mathbf{Y}^T \mathbf{Y}}} \quad (13)$$

である. これより,

$$\sigma_1 \simeq \sqrt{\text{tr} \mathbf{Y}^T \mathbf{Y}}, \sigma_2 \simeq \sqrt{\frac{\det \mathbf{Y}^T \mathbf{Y}}{\text{tr} \mathbf{Y}^T \mathbf{Y}}} \quad (14)$$

が導かれる. ここで $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2]^T$ とし, $\mathbf{y}_1, \mathbf{y}_2$ の成す角度を θ とすると,

$$\sqrt{\text{tr} \mathbf{Y}^T \mathbf{Y}} = \sqrt{\|\mathbf{y}_1\|^2 + \|\mathbf{y}_2\|^2} \quad (15)$$

$$\sqrt{\det \mathbf{Y}^T \mathbf{Y}} = \|\mathbf{y}_1\| \|\mathbf{y}_2\| |\sin \theta| \quad (16)$$

であるので,

$$\sigma_1 \simeq \sqrt{\|\mathbf{y}_1\|^2 + \|\mathbf{y}_2\|^2}, \sigma_2 \simeq \frac{\|\mathbf{y}_1\| \|\mathbf{y}_2\| |\sin \theta|}{\sqrt{\|\mathbf{y}_1\|^2 + \|\mathbf{y}_2\|^2}} \quad (17)$$

である. よって σ_1 を正則化することで \mathbf{Y} の列ベクトルの長さの 2 乗和を短くし, σ_2 を正則化することで \mathbf{Y} の列ベクトルが作る角度の正弦の絶対値を小さくする効果がある. これより, 核型ノルム正則化では 2 辺の成す角度を正則化するだけでなく, 辺の長さを短くする効果が表れてしまい, 局所線形整列により角が丸まってしまう.

3. 加重核型ノルム正則化に基づくグラフ単純化

前節で述べた問題を解決するために, 加重核型ノルム正則化に基づくグラフ形状単純化法を提案する. 提案法は従来の局所線形近似法と同じく局所線形整列と不要頂点除去の 2 ステップで構成される. 局所線形整

列において用いる正則化関数を核型ノルムから加重核型ノルム [11] に変更する. すなわち,

$$\|\mathbf{Y}\|_{*,\mathbf{w}} = w_1 \sigma_1 + w_2 \sigma_2 \quad (18)$$

を用いる. ここで $\mathbf{w} = [w_1, w_2]^T$ は重みで, $0 < w_1 < w_2$ と設定する. 加重核型ノルムは行列の低ランク性を推進する非凸な正則化関数である. 加重核型ノルムでは, (17) より, 小さい特異値 σ_2 により大きく加重されるため, 通常の核型ノルムと比較して, 行列のプロベニウスノルムを損なうことなく, 低ランク性をより強く推進することができる. このノルムを局所線形整列問題に用いれば, 辺の長さを短くする正則化が抑制され, 辺の成す角度の正則化をより強調できる.

加重核型ノルムを用いて提案法の局所線形整列問題を,

$$\mathbf{P}' = \arg \min_{\mathbf{X} \in \mathbb{R}^{N \times D}} \frac{1}{2} \|\mathbf{P} - \mathbf{X}\|^2 + \sum_{v \in \mathcal{V}} \|\mathbf{L}_v \mathbf{X}\|_{*,\mathbf{w}} \quad (19)$$

と定義する. この問題は非凸最適化問題であるが, 加重核型ノルムの近接写像である特異値閾値処理が

$$\text{prox}_{\|\cdot\|_{*,\mathbf{w}}}(\mathbf{Y}) = \mathbf{U} \text{diag}((\boldsymbol{\sigma} - \mathbf{w})_+) \mathbf{V}^T \quad (20)$$

と計算できるため, ADMM を使って準最適解を求められる. ここで $\mathbf{Y} = \mathbf{U} \text{diag}(\boldsymbol{\sigma}) \mathbf{V}^T$ は \mathbf{Y} の特異値分解である.

問題 (19) の ADMM による解法について述べる. まず $\mathbf{Z}_v = \mathbf{L}_v \mathbf{X} (v \in \mathcal{V})$ とし,

$$\begin{bmatrix} \mathbf{Z}_{v_1} \\ \vdots \\ \mathbf{Z}_{v_{|\mathcal{V}|}} \end{bmatrix} = \mathbf{Z}, \begin{bmatrix} \mathbf{L}_{v_1} \\ \vdots \\ \mathbf{L}_{v_{|\mathcal{V}|}} \end{bmatrix} = \mathbf{L} \quad (21)$$

と置くと, $\mathbf{Z} = \mathbf{L} \mathbf{X}$ が成り立つ. また $g(\mathbf{Z}) = \sum_{v \in \mathcal{V}} \|\mathbf{Z}_v\|_{*,\mathbf{w}}$ と置く. これより問題 (19) は以下の

$$\min_{\mathbf{X} \in \mathbb{R}^{N \times D}} \frac{1}{2} \|\mathbf{P} - \mathbf{X}\|^2 + g(\mathbf{Z}) \text{ s.t. } \mathbf{Z} = \mathbf{L} \mathbf{X} \quad (22)$$

と同値である. この問題は ADMM を適用できる標準形である. ADMM の反復更新式は更新パラメータを $\gamma > 0$ とし,

$$\begin{cases} \mathbf{X}^{(k+1)} = \arg \min_{\mathbf{X} \in \mathbb{R}^{N \times D}} \frac{1}{2} \|\mathbf{P} - \mathbf{X}\|^2 + \frac{1}{2\gamma} \|\mathbf{Y}^{(k)} - \mathbf{L} \mathbf{X} - \mathbf{D}^{(k)}\|^2 \\ \mathbf{Y}^{(k+1)} = \arg \min_{\mathbf{Y} \in \mathbb{R}^{2|\mathcal{V}| \times D}} g(\mathbf{Y}) + \frac{1}{2\gamma} \|\mathbf{Y} - \mathbf{L} \mathbf{X}^{(k+1)} - \mathbf{D}^{(k)}\|^2 \\ \mathbf{D}^{(k+1)} = \mathbf{D}^{(k)} + \mathbf{L} \mathbf{X}^{(k+1)} - \mathbf{Y}^{(k+1)} \end{cases} \quad (23)$$

である. これらの更新式の計算方法について述べる. まず第 1 式については, 以下の線形方程式の解である.

$$(\mathbf{I} + \gamma^{-1} \mathbf{L}^T \mathbf{L}) \mathbf{X} = \mathbf{P} + \gamma^{-1} \mathbf{L}^T (\mathbf{Y}^{(k)} - \mathbf{D}^{(k)}) \quad (24)$$

$$\Leftrightarrow \left(\mathbf{I} + \gamma^{-1} \sum_{v \in \mathcal{V}} \mathbf{L}_v^T \mathbf{L}_v \right) \mathbf{X} = \mathbf{P} + \gamma^{-1} \sum_{v \in \mathcal{V}} \mathbf{L}_v^T (\mathbf{Y}_v^{(n)} - \mathbf{D}_v^{(n)}) \quad (25)$$

アルゴリズム 1 最適化問題 (19) を解く方法

Input: $\mathbf{P}, \mathbf{w}, \gamma_k, N_{\text{iter}}$
Output: $\mathbf{X} = \arg \min_{\mathbf{X}} \frac{1}{2} \|\mathbf{P} - \mathbf{X}\|^2 + \sum_{v \in \bar{\mathcal{V}}} \|\mathbf{L}_v \mathbf{X}\|_{*, \mathbf{w}}$

- 1: $\mathbf{Y}_v^{(0)} = \mathbf{O}, \mathbf{D}_v^{(0)} = \mathbf{O}$ for $v \in \bar{\mathcal{V}}$
- 2: **for** $k = 0, 1, \dots, N_{\text{iter}} - 1$ **do**
- 3: % 1st updating formula
- 4: $(\mathbf{L}, \mathbf{U}) = \text{matrixDecomposition}(\mathbf{I} + \gamma_k^{-1} \sum_{v \in \bar{\mathcal{V}}} \mathbf{L}_v^T \mathbf{L}_v)$
- 5: $\mathbf{X}^{(k+1)} = \mathbf{U}^{-1} \mathbf{L}^{-1} (\mathbf{P} + \gamma_k^{-1} \sum_{v \in \bar{\mathcal{V}}} \mathbf{L}_v^T (\mathbf{Y}_v^{(k)} - \mathbf{D}_v^{(k)}))$
- 6: % 2nd updating formula
- 7: **for** $v \in \bar{\mathcal{V}}$ **do**
- 8: $\mathbf{Y}_v^{(k+1)} = \text{prox}_{\|\cdot\|_{*, \gamma_k \mathbf{w}}}(\mathbf{L}_v \mathbf{X}^{(k+1)} + \mathbf{D}_v^{(k)})$
- 9: **end for**
- 10: % 3rd updating formula
- 11: **for** $v \in \bar{\mathcal{V}}$ **do**
- 12: $\mathbf{D}_v^{(k+1)} = \mathbf{D}_v^{(k)} + \mathbf{L}_v \mathbf{X}^{(k+1)} - \mathbf{Y}_v^{(k+1)}$
- 13: **end for**
- 14: **end for**
- 15: $\mathbf{X} = \mathbf{X}^{(N_{\text{iter}})}$

これは左辺の係数行列を Cholesky 分解 [12] や LU 分解 [13] により三角行列の積に分解することで、効率的に解くことができる。

続いて第 2 式については、加重核型ノルムの和の近接写像計算である。

$$\mathbf{Y}^{(k+1)} = \text{prox}_{\gamma g}(\mathbf{LX}^{(k+1)} + \mathbf{D}^{(k)}) \quad (26)$$

$$\Leftrightarrow \mathbf{Y}_v^{(k+1)} = \text{prox}_{\|\cdot\|_{*, \gamma \mathbf{w}}}(\mathbf{L}_v \mathbf{X}^{(k+1)} + \mathbf{D}_v^{(k)}) \text{ for } v \in \bar{\mathcal{V}} \quad (27)$$

よって式 (20) の特異値閾値処理により計算できる。

最後に第 3 式については、

$$\mathbf{D}_v^{(k+1)} = \mathbf{D}_v^{(k)} + \mathbf{L}_v \mathbf{X}^{(k+1)} - \mathbf{Y}_v^{(k+1)} \text{ for } v \in \bar{\mathcal{V}} \quad (28)$$

で計算できる。

以上をまとめて、問題 (19) の解を求める方法をアルゴリズム 1 に示す。アルゴリズム 1 において関数 `matrixDecomposition` は、Cholesky 分解や LU 分解による三角行列の積への分解を表している。また ADMM の更新パラメータ γ_k は、通常の ADMM では k に対して一定値であるが、本アルゴリズムにおいては k に応じて変化する数列とする。これにより非凸な本問題における収束の安定性が増し、また収束速度が改善する効果が得られる。具体的に γ_k にどのような数列を選ぶかは、アルゴリズムの 4.5 行目の第 1 更新式の高速化と関わるため、後の小節 4.2 において明らかにする。

4. グラフ単純化の高速計算法

前節で述べたグラフ形状単純化を高速に計算する方法について述べる。手法は 2 つあり、1 つ目は高速特異値閾値処理法を加重核型ノルムに拡張し、第 1 のボトルネックである近接写像計算を高速化する。2 つ目は ADMM における第 2 のボトルネックである反復的な線形方程式の求解について計算量を削減し、高速化を実現する。

4.1. 加重核型ノルムのための高速特異値閾値処理

アルゴリズム 1 の計算時間において最大のボトルネックとなっているのが、8 行目の特異値閾値処理である。

そこで小節 2.3 で述べた特異値閾値処理の高速計算法を、加重核型ノルムのための方法に拡張する。ここでは結果のみ示し、詳細な証明は付録に記載する。

まず $D \geq 3$ の $D \times 2$ 行列 $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2]$ の特異値閾値処理は以下で求められる。

$$\text{prox}_{\|\cdot\|_{*, \mathbf{w}}}(\mathbf{Y}) = \mathbf{Y} \left(\frac{\alpha + \beta}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{\beta - \alpha}{2\sqrt{\det \mathbf{Y}^T \mathbf{Y}}} \begin{bmatrix} -\mathbf{y}_2^T \mathbf{y}_2 & \mathbf{y}_1^T \mathbf{y}_2 \\ \mathbf{y}_1^T \mathbf{y}_2 & -\mathbf{y}_1^T \mathbf{y}_1 \end{bmatrix} \right) \quad (29)$$

続いて 2×2 行列 $\mathbf{Y} = \begin{bmatrix} y_{1,1} & y_{1,2} \\ y_{2,1} & y_{2,2} \end{bmatrix}$ の特異値閾値処理は以下で求められる。

$$\text{prox}_{\|\cdot\|_{*, \mathbf{w}}}(\mathbf{Y}) = \frac{\alpha + \beta}{2} \mathbf{Y} + \text{sgn}(\det \mathbf{Y}) \frac{\beta - \alpha}{2} \begin{bmatrix} -y_{2,2} & y_{2,1} \\ y_{1,2} & -y_{1,1} \end{bmatrix} \quad (30)$$

ただしここで係数 α, β は以下の式で求める。

$$\alpha = \begin{cases} \frac{(\sigma_1 - w_1 + (\sigma_2 - w_2)_+)_+}{\sigma_1 + \sigma_2} & \text{if } \sigma_1 + \sigma_2 \neq 0 \\ 0 & \text{if } \sigma_1 + \sigma_2 = 0 \end{cases} \quad (31)$$

$$\beta = \begin{cases} \frac{(\sigma_1 - w_1 - (\sigma_2 - w_2)_+)_+}{\sigma_1 - \sigma_2} & \text{if } \sigma_1 - \sigma_2 \neq 0 \\ 0 & \text{if } \sigma_1 - \sigma_2 = 0 \end{cases} \quad (32)$$

ここで σ_1, σ_2 は \mathbf{Y} の第 1 特異値、第 2 特異値で、式 (11) により求められる。以上をまとめて、特異値閾値処理の算出法をアルゴリズム 2.3 に記述する。

ここで提案した手法の計算量について考える。浮動小数点演算の回数においてアルゴリズム 2 で $12D + 26$ 回、アルゴリズム 3 で 33 回である。これは式 (20) で算出する $30D + 166$ 回よりも $60 - 85\%$ 浮動小数点演算回数を削減できている。

続いて提案手法の計算並列性について考える。多数の入力行列があって特異値閾値処理を並列に実行する場合、アルゴリズム 2.3 の計算のほとんどが積和演算、四則演算、根号演算で構成されるため、データ並列に計算を実行可能である。唯一の条件分岐部である 5, 6 行目についても、マスク処理を用いれば容易に並列実行可能である。よって、CPU や GPU に備わる SIMD 機能を活用して高効率に特異値閾値処理を並列計算可能である。

4.2. ADMM における線形方程式の高速解法

アルゴリズム 1 の計算時間の次なるボトルネックとなるのが、4 行目の行列分解である。この行列分解は Cholesky 分解や LU 分解を用いるが、これらの計算量は $O(N^3)$ (N はグラフの頂点数) のオーダーである。これはその他の処理が高々 $O(N^2)$ のオーダーであるので、非常に高い計算コストであると言える。

そこで本稿では、行列分解の回数を減らすことで、計算量を削減することを考える。このために、以下のス

アルゴリズム 2 特異値閾値処理算出法 ($D \geq 3$)**Input:** $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2] \in \mathbb{R}^{D \times 2}$, $\mathbf{w} = [w_1, w_2]^T$ **Output:** $\mathbf{Z} = \text{prox}_{\|\cdot\|_{*,\mathbf{w}}}(\mathbf{Y})$

- 1: $a = \mathbf{y}_1^T \mathbf{y}_1, b = \mathbf{y}_1^T \mathbf{y}_2, c = \mathbf{y}_2^T \mathbf{y}_2$
- 2: $d = ac - b^2, e = \sqrt{d}, f = a + c$
- 3: $g = \sqrt{f + 2e}, h = \sqrt{f - 2e}$
- 4: $\sigma_1 = \frac{g+h}{2}, \sigma_2 = \frac{g-h}{2}$
- 5: $\alpha = \begin{cases} \frac{(\sigma_1 - w_1 + (\sigma_2 - w_2)_+)_+}{g} & \text{if } g \neq 0 \\ 0 & \text{if } g = 0 \end{cases}$
- 6: $\beta = \begin{cases} \frac{(\sigma_1 - w_1 - (\sigma_2 - w_2)_+)_+}{h} & \text{if } h \neq 0 \\ 0 & \text{if } h = 0 \end{cases}$
- 7: $\mathbf{Z} = \mathbf{Y} \left(\frac{\alpha + \beta}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{\beta - \alpha}{2e} \begin{bmatrix} -c & b \\ b & -a \end{bmatrix} \right)$

アルゴリズム 3 特異値閾値処理算出法 ($D = 2$)**Input:** $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2] = \begin{bmatrix} y_{1,1} & y_{1,2} \\ y_{2,1} & y_{2,2} \end{bmatrix}$, $\mathbf{w} = [w_1, w_2]^T$ **Output:** $\mathbf{Z} = \text{prox}_{\|\cdot\|_{*,\mathbf{w}}}(\mathbf{Y})$

- 1: $a = \mathbf{y}_1^T \mathbf{y}_1, c = \mathbf{y}_2^T \mathbf{y}_2$
- 2: $d = y_{1,1}y_{2,2} - y_{1,2}y_{2,1}, e = |d|, f = a + c$
- 3: $g = \sqrt{f + 2e}, h = \sqrt{f - 2e}$
- 4: $\sigma_1 = \frac{g+h}{2}, \sigma_2 = \frac{g-h}{2}$
- 5: $\alpha = \begin{cases} \frac{(\sigma_1 - w_1 + (\sigma_2 - w_2)_+)_+}{g} & \text{if } g \neq 0 \\ 0 & \text{if } g = 0 \end{cases}$
- 6: $\beta = \begin{cases} \frac{(\sigma_1 - w_1 - (\sigma_2 - w_2)_+)_+}{h} & \text{if } h \neq 0 \\ 0 & \text{if } h = 0 \end{cases}$
- 7: $\mathbf{Z} = \frac{\alpha + \beta}{2} \mathbf{Y} + \text{sgn}(d) \frac{\beta - \alpha}{2} \begin{bmatrix} -y_{2,2} & y_{2,1} \\ y_{1,2} & -y_{1,1} \end{bmatrix}$

ステップ型増加数列 γ_k を導入する.

$$\gamma_k = \gamma'_{\lfloor k/T \rfloor} \text{ for } k = 0, 1, \dots, \quad (33)$$

ここで γ'_k は任意の単調増加数列であり, T は増加周期である. また $\lfloor \cdot \rfloor$ は床関数である. 図 2 にステップ型増加数列 γ_k の例を示す. 図の通り, ステップ型増加数列 γ_k は周期 T で値が増加する数列であり, 増加時以外は一定値になっている.

ステップ型増加数列 γ_k を用いてアルゴリズム 1 を適用すれば, 4 行目の行列分解の回数を $1/T$ に減らせる. 何故ならば, 入力行列 $\mathbf{I} + \gamma_k^{-1} \sum_{v \in \mathcal{V}} \mathbf{L}_v^T \mathbf{L}_v$ が T 回に 1 回しか変化しないためである.

この改良は計算量オーダー $\mathcal{O}(N^3)$ を根本的に減らしはしていないが, 頂点数 N が非常に大きいときに速度改善効果もより大きくなることを期待できる.

5. 実験

提案法の有効性を確かめるためにグラフ単純化シミュレーションを行う. 実験は CPU : Core i9-10980XE@4.8GHz, RAM : 64.0GB のコンピュータを用い, ソフトウェアは MATLAB 2018b を用いた.

提案法は加重核型ノルム正則化に基づくグラフ単純

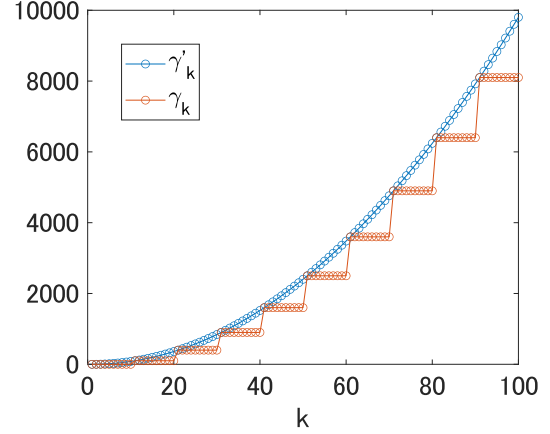


図 2: ステップ型増加数列の例

化法 (Weighted Nuclear Norm based Graph Simplification; WNNGS) で, 重みパラメータ $\mathbf{w} = [w_1, w_2]^T$ として, $w_1 = 0.01, 0.1, 0.5, 1.0, 1.5$, $w_2 = 50$ を用いた. また ADMM のパラメータはステップ型増加関数として $\gamma_k = \frac{1}{3} 10^{\lfloor \frac{k}{50} \rfloor} \lfloor \frac{k}{50} \rfloor^{50}$, 反復回数 $N_{\text{iter}} = 300$ を用いた. また従来法として DP [1, 2] 法と筆者らの核型ノルム正則化に基づくグラフ単純化法 (Nuclear Norm based Graph Simplification; NNGS) [3] を用いた.

図 3 にグラフ単純化結果を示す. また図 4 に頂点数と 2 乗平均誤差 (Mean Square Error; MSE) のプロットを示す. ここで 2 つの埋め込みグラフ $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{P})$ と $\mathcal{G}' = (\mathcal{V}', \mathcal{E}', \mathbf{P}')$ の間の MSE は

$$\text{MSE}(\mathcal{G}, \mathcal{G}') = \frac{\sum_{i=1}^N \|\mathbf{p}_i - \text{proj}_{\mathcal{G}'}(\mathbf{p}_i)\|^2}{N} \quad (34)$$

と定義する. ここで $\text{proj}_{\mathcal{G}'}(\mathbf{p}_i)$ は, 点 \mathbf{p}_i を \mathcal{G}' が作る辺上の最も近い点に射影する関数である. 2 つの図が示す通り, 提案法は DP 法や NNGS と比較して, より少ない頂点数, MSE でグラフを単純化できる.

続いて雑音の頑健性について述べる. 図 5 は雑音を重畳したグラフに対する単純化結果を示している. また図 6 に頂点数と MSE のプロットを示す. 2 つの図が示す通り, DP 法や NNGS と比較して提案法 WNNGS は雑音下においても頑健に働き, グラフを適切に単純化できる.

続いて, 第 4 節で述べた高速化法の効果について述べる. 図 7 は, 高速化法を適用する前後の実行時間をプロファイリングしたものである. 図が示すように高速化前は 18.20 秒計算に時間を要し, そのうち 17.57 秒が特異値閾値処理, 0.50 秒が線形方程式の求解に費やされていた. 一方高速化後においては計算時間は 0.54 秒で, そのうち 0.30 秒が特異値閾値処理, 0.15 秒が線形方程式の求解に要する時間となった. よって全体では 33.7 倍の高速化となった.

6. おわりに

本稿では加重核型ノルム正則化に基づくグラフ単純化法を提案した. グラフの折れ曲がり回数を正則化する関数として加重核型ノルムを用いることで, グラフ

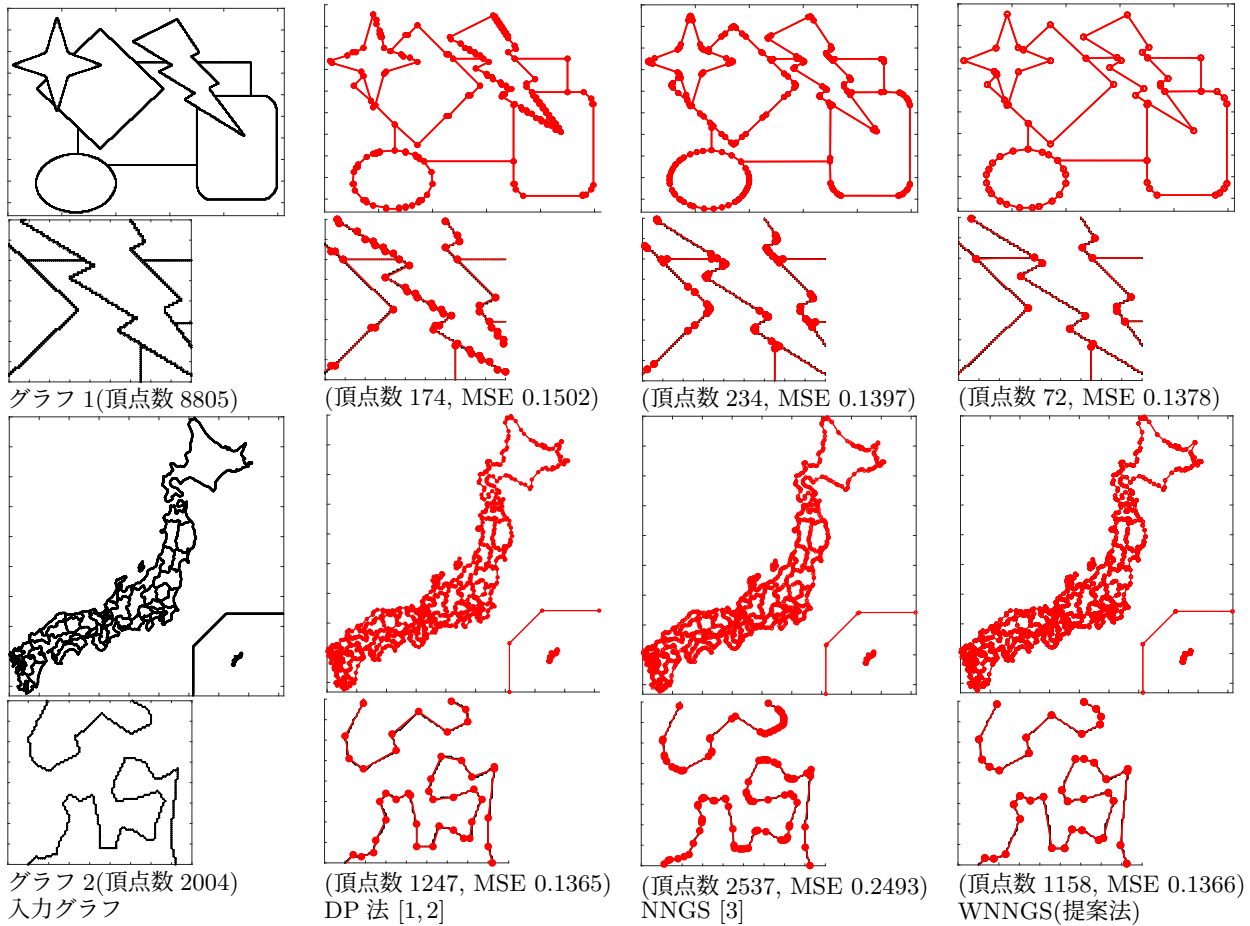


図 3: グラフ単純化結果

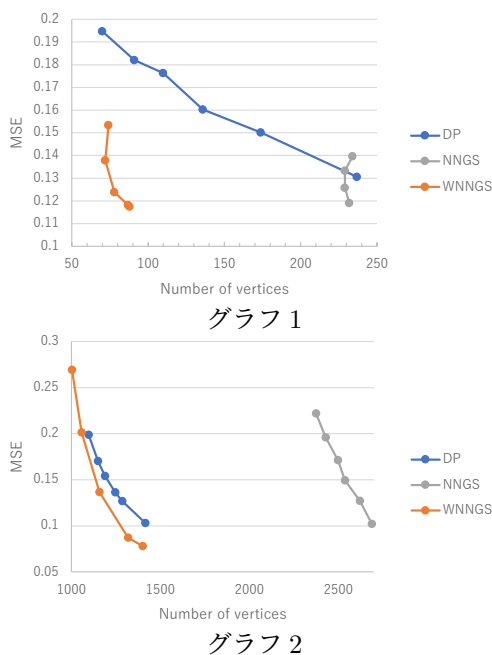


図 4: グラフ単純化の頂点数と MSE のプロット

形状の精度良い近似と頂点数の削減を同時に達成した。加重核型ノルムは非凸関数であるが近接写像が計算可能であるため、近接分離法である交互方向乗数法により効率的に準最適解を求められた。また高速特異値閾値処理を核型ノルムに拡張し、また線形方程式の計算量を削減し、本手法の高速計算法を提案した。実験では提案法によるグラフ形状単純化を行い、従来法よりも近似精度が高くより頂点数の少ない単純なグラフを高速に得られることを示した。

付録

高速特異値閾値処理の導出

加重核型ノルムのための高速特異値閾値処理 (29)-(32) を導出する。入力 $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2]$ の特異値を σ_1, σ_2 とする ($\sigma_1 \geq \sigma_2$)。特異値の和及び差について以下の式が成立する [7]。

$$\sigma_1 \pm \sigma_2 = \|\mathbf{y}_1 \mp \mathbf{R}\mathbf{y}_2\| \quad (35)$$

ここで \mathbf{R} は $\text{Im}\mathbf{Y}$ 上のベクトルを $\pi/2$ 回転させる回転行列で、

$$\mathbf{R} = \frac{1}{\sqrt{\det \mathbf{Y}^T \mathbf{Y}}} (\mathbf{y}_2 \mathbf{y}_1^T - \mathbf{y}_1 \mathbf{y}_2^T) \quad (36)$$

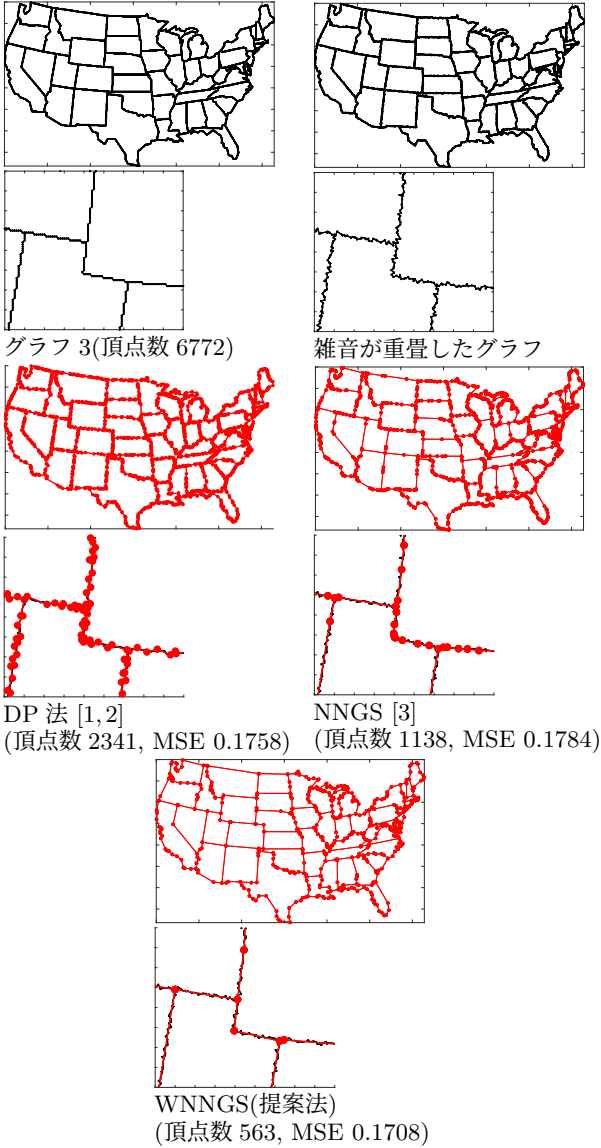


図 5: 雑音 that 重畳したグラフの単純化結果

である。これより加重核型ノルムは

$$\begin{aligned} \|\mathbf{Y}\|_{*,\mathbf{w}} &= \frac{w_1 + w_2}{2} \|\mathbf{y}_1 - \mathbf{R}\mathbf{y}_2\| + \frac{w_1 - w_2}{2} \|\mathbf{y}_1 + \mathbf{R}\mathbf{y}_2\| \\ &= \Gamma_{\omega}(\mathbf{A}\text{vec}(\mathbf{Y})) \end{aligned} \quad (37)$$

となる。ここで $\omega = [\frac{w_1+w_2}{2}, \frac{w_1-w_2}{2}]^T$ とおき、

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ \mathbf{I} & \mathbf{R} \end{bmatrix} \quad (38)$$

とおいた。また $\Gamma_{\omega}(\cdot)$ はグループ尖星準ノルム [14] で、

$$\Gamma_{\omega}(\mathbf{x}_1, \mathbf{x}_2) = \omega_1 \max(\|\mathbf{x}_1\|, \|\mathbf{x}_2\|) + \omega_2 \min(\|\mathbf{x}_1\|, \|\mathbf{x}_2\|) \quad (39)$$

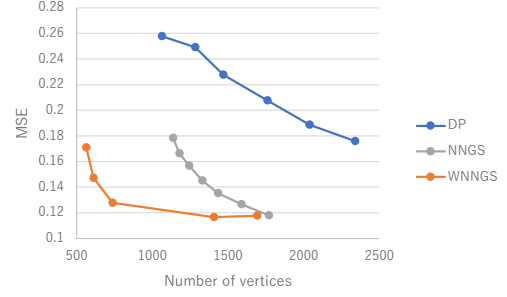


図 6: 雑音 that 重畳したグラフの単純化の頂点数と MSE のプロット

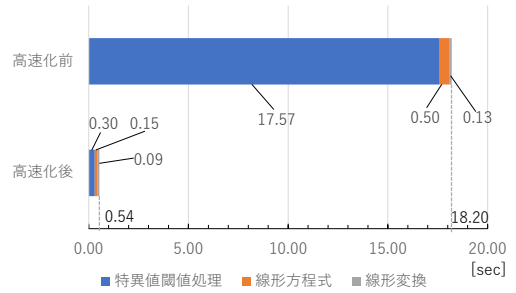


図 7: 実行時間プロファイリング結果

である。これより近接写像は、

$$\begin{aligned} \text{prox}_{\|\cdot\|_{*,\mathbf{w}}}(\mathbf{Y}) &= \text{prox}_{\Gamma_{\omega}}(\mathbf{A}\text{vec}(\cdot))(\mathbf{Y}) \\ &= \text{vec}^T \mathbf{A}^T \text{prox}_{\Gamma_{\omega}}\left(\frac{1}{2} \mathbf{A}\text{vec} \mathbf{Y}\right) \end{aligned} \quad (40)$$

$$= \text{vec}^T \mathbf{A}^T \text{prox}_{\Gamma_{\omega}}\left(\frac{1}{2}(\mathbf{y}_1 - \mathbf{R}\mathbf{y}_2), \frac{1}{2}(\mathbf{y}_1 + \mathbf{R}\mathbf{y}_2)\right) \quad (41)$$

となる。ここで (40) においては、 $\mathbf{A}^T \mathbf{A} = \mathbf{A} \mathbf{A}^T = 2\mathbf{I}$ が成立することと、その \mathbf{A} について $\text{prox}_{\psi(\mathbf{A}\cdot)} = \frac{1}{2} \mathbf{A}^T \text{prox}_{2\psi}(\mathbf{A}\mathbf{x})$ が成立すること [15] を用いた。また $\text{prox}_{\Gamma_{\omega}}(\cdot)$ は、

$$(\text{prox}_{\Gamma_{\omega}}(\mathbf{x}_1, \mathbf{x}_2))_i = \begin{cases} \frac{p_i}{\|\mathbf{x}_i\|} \mathbf{x}_i & \text{if } \mathbf{x}_i \neq \mathbf{0} \\ \mathbf{0} & \text{if } \mathbf{x}_i = \mathbf{0} \end{cases} \quad (42)$$

$$[p_1, p_2]^T = \text{prox}_{\Omega_{\omega}}(\|\mathbf{x}_1\|, \|\mathbf{x}_2\|) \quad (43)$$

である。ここで Ω_{ω} は尖星準ノルムである [14]。今、 $\|\mathbf{x}_1\| = \frac{1}{2} \|\mathbf{y}_1 - \mathbf{R}\mathbf{y}_2\| \geq \|\mathbf{x}_2\| = \frac{1}{2} \|\mathbf{y}_1 + \mathbf{R}\mathbf{y}_2\| \geq 0$ であるので (43) は簡単に計算できて、

$$p_1 = \frac{1}{2}(\sigma_1 - w_1 + (\sigma_2 - w_2)_+)_+ \quad (44)$$

$$p_2 = \frac{1}{2}(\sigma_1 - w_1 - (\sigma_2 - w_2)_+)_+ \quad (45)$$

である。以上より、 α, β を (31), (32) とおけば、

$$\text{prox}_{\|\cdot\|_{*,\mathbf{w}}}(\mathbf{Y}) = \frac{\alpha + \beta}{2} \mathbf{Y} + \frac{\alpha + \beta}{2} \mathbf{R}[\mathbf{y}_2, -\mathbf{y}_1] \quad (46)$$

が得られる。回転行列 \mathbf{R} を (12) により展開すれば、(29) 及び (30) を得る。

参考文献

- [1] D.H. Douglas and T.K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol.10, no.2, pp.112–122, 1973.
- [2] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer Graphics and Image Processing*, vol.1, no.3, pp.244–256, 1972.
- [3] 佐々木崇元, 谷田隆一, 清水淳, "グラフ信号の局所線形近似によるグラフ形状単純化," 第 15 回 情報科学技術フォーラム (FIT), 第 3 分冊, pp.1–4, 2016.
- [4] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & mathematics with applications*, vol.2, no.1, pp.17–40, 1976.
- [5] J.-F. Cai, E.J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on optimization*, vol.20, no.4, pp.1956–1982, 2010.
- [6] 佐々木崇元, 北原正樹, 清水淳, "領域情報符号化における核型ノルム最適化の高速計算法," 第 31 回 画像符号化シンポジウム (PCSJ), pp.140–141, 2016.
- [7] 佐々木崇元, 北原正樹, 清水淳, "低ランク最適化のための高速特異値閾値処理の数理," 第 16 回 情報科学技術フォーラム (FIT), 第 1 分冊, pp.5–12, 2017.
- [8] J.J Moreau, "Fonctions convexes duales et points proximaux dans un espace hilbertien," *Comptes Rendus de l'Académie des Sciences*, vol.225, pp.2897–2899, 1962.
- [9] M. Fazel, "Matrix rank minimization with applications," PhD thesis, Stanford University, 2002.
- [10] L. Condat, "A primal-dual splitting method for convex optimization involving Lipschitzian, proximal and linear composite terms," *Journal of Optimization Theory and Applications*, vol.158, no.2, pp.460–479, 2013.
- [11] K. Chen, H. Dong, and K.-S. Chan, "Reduced rank regression via adaptive nuclear norm penalization," *Biometrika*, vol.100, no.4, pp.901–920, 2013.
- [12] A.S. Householder, *The theory of matrices in numerical analysis*, Blaisdell Publishing Company, 1964.
- [13] A.M. Turing, "Rounding-off errors in matrix processes," *The Quarterly Journal of Mechanics and Applied Mathematics*, vol.1, no.1, pp.287–308, 1948.
- [14] 佐々木崇元, 坂東幸浩, 北原正樹, "尖星全変動正則化に基づくグラデーションとエッジ同時制御可能なエッジ保存平滑化," 第 36 回信号処理シンポジウム, pp.181–186, 2021.
- [15] P.L. Combettes and J.-C. Pesquet, "A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery," *IEEE Journal of Selected Topics in Signal Processing*, vol.1, no.4, pp.564–574, 2007.